DEVELOPING MUSCLE MEMORY USING HUMAN TO HUMAN INTERFACE

Harsha Niroshana Yapa

149161C

Department of Computational Mathematics
University of Moratuwa
Sri Lanka
February 2017

DEVELOPING MUSCLE MEMORY USING HUMAN TO HUMAN INTERFACE

Harsha Niroshana Yapa

149161C

Thesis submitted in partial fulfilment of the requirements of the
Degree of MSc in Artificial Intelligence.

Department of Computational Mathematics,
University of Moratuwa,
Sri Lanka.
February 2017

Declaration

I declare that this dissertation does not incorporate, without acknowledgment, any material previously submitted for a Degree or a Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organization.

UOM Verified Signature

Name of Student

Harsha Niroshana Yapa

Signature of Student

Date: 09/02/2018

Supervised by

Prof. Asoka S. Karunananda

Name of Supervisor(s)

Signature of Supervisor(s)

Date:

Acknowledgement

I would like to thank Professor Asoka S. Karunananda for all his input, advice, and encouragement throughout the course and always being on hand to provide helpful comments. In fact, when I met Prof. Asoka for the first time, he planted this research idea in my mind, which eventually came out in this project.

I am grateful to my wife, Dulanga Kodithuwakku & lovely baby girl Dulanya for kind support rendered to me when I need it the most. Thanks for being so patience with me during the stressful times.

My parents, they always wanted me study whatever and whenever i get an opportunity. Thanks for pushing me and encouraging me to successfully complete another milestone in my education.

Thanks must also go to Dr. Sisitha Ukwatta, for providing me with information on Human nerve system and related medical terms & procedures. Last, but certainly not least, I am grateful to my batch mates, who supported me in numerous ways.

Abstract

The Learning is something very crucial for the development of the human race. Humans are not a species that bring their intellect hard coded to their DNA, instead they learn things by observation and repeating the activity until it develop muscle memory. In this research, an attempt was made to minimize the time associated with the learning and memorizing of these repetitive tasks by using a Human to Human Interface, where it will capture neural impulses from one human and feed to second human, allowing the second human to develop same type of muscle memory.

As humans, we learn different things by observing how others perform it. If we take an example, children learn to write letters by observing the how their parents write them, or parents may actually hold their kids' hand and guide them on writing letters. When parent hold his/her child's hand and guide them, muscles on Child's hand send sensory signals to brain claiming the way to perform this particular action. However imagine action like learning to play piano, this is complex activity which needs insane amount of time of practicing. A research has been conducted to identify neural impulses generated when piano player is playing a song. Special EEG capturing device is constructed to capture neural impulses from Median, Ulnar, and Radial nerves of the Piano Player's Hands. The EEG signals were captured using the non-invasive methods. The captured signal is then processed through filter, where it will remove all the noises and unwanted EEG signals/muscle Contractions generated due to Blood Flow, etc. These processed signals then feed to TENS device, where it generate artificial stimulation as form of electric current. This new feed of current then applied to trainee's nerves using the passive techniques. Also this stage, there is a noise added to the signal, due to method used in administering the stimulation to trainee's hand. Once the signal applied to respective nerves on the second human (trainee), his hands are moved according to the trainers hand movements.

This solution has been tested using three test subjects (2 men and 1 women), with an average age of 28 years, and results show rapid increase in the effectiveness of the learning when this interface is used. The Error rate while playing piano using Human-to-Human interface was 1.4167 where conventional method showed rate of 2.1666. In conclusion, Human to Human interface assisted learning was 32.69% error free than the conventional method of learning.

Contents

Page

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Prolegomena

Neuro-prosthetics is designing a machine that interfaces with living neurons to control a device or for sensory substitution. But what about muscles? If people have damage to their spinal nerves, the muscles themselves can be stimulated, and this line of research is called Functional Electrical Stimulation. For Example, Functional electrical stimulation (FES)[1] is a technique that uses electrical currents to activate nerves innervating extremities affected by paralysis resulting from spinal cord injury (SCI), head injury, stroke and other neurological disorders, where can often be used to help someone stand up, or to improve walking by helping to swing a foot forward. These specific neural impulses are generated using special equipment which generally stored in human brain as a muscle memory.

Muscle memory is a form of Procedural memory that involves consolidating a specific motor task into memory by repetition. When a motor task is repeated over the time, a long-term muscle memory is created for that specific task, which allowing it to be performed without conscious effort. After developing a muscle memory, it decreases the need of continuous attention and where increases the chance of performing another task simultaneously.

It is true that practice makes human perfect, but practice takes long time as well human tend to learn things by mistakes. In this project, the application of training Piano Student by a Professional piano player is taken as an example. In case of Piano Teacher and Student, Student learn to play piano by observing how teacher plays and trying to imitate the same steps as performed by teacher.

Developing a muscle memory to play piano is a time consuming process. Usually it takes years of practice to learn exact steps to play piano. Playing piano is one of the most complicated process of all, compared to other music instruments. However it is hypothesize that using human to human interface[2], the time associated with the training can be reduced considerably. The Human to Human interface is used to capture neural impulses of professional piano player while playing the piano and feed the captured neural impulse stream to Student, this will increase the efficiency and also decrease the time of developing muscle memory.

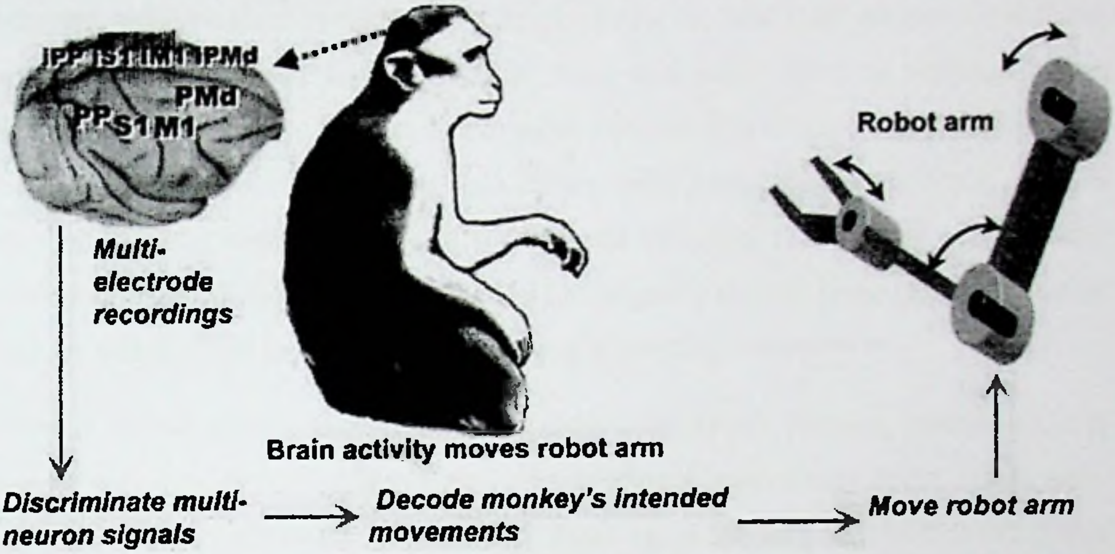


Figure 1.1 - Brain Activity of Monkey moves a Robot Arm

1.2 Aims & Objectives

The aim of this project is to develop human to human interface which can be used to develop muscle memory on another human.

In order to reach this aim following objectives are identified.

- To critically study the domain of EEG & BCI field.
- To critically analyze the current approaches on duplicating Motor actions on another human.
- Recognizing the suitable technology to develop Human to Human Interface.
- Design & Develop Human to Human Interface
- Evaluate the Solution
- Final Documentation

1.3 Background and Motivation

The Learning is something very crucial for the development of the human race. Humans are not a species that bring their intellect hard coded to their DNA, instead they learn things by observation and repeating the activity until it develop muscle memory.

As humans, we learn different things by observing how others perform it. If we take an example, children learn to write letters by observing the how their parents write them, or parents may actually hold their kids' hand and guide them on writing letters. However we know for a fact that it take some time to child to learn and grasp the way their parents write it. When parent hold his/her child's hand and guide them, muscles on Child's hand send sensory signals to brain claiming the way to perform this particular action. However imagine action like learning to play piano, this is complex activity which needs insane amount of time of practicing.

There is perfect method identified yet to improve the human learning experience, and surprisingly there is not much attention on the research community about this as well. Prof. Kevin Warwick from the University Reading, is the only person following this kind of research [6]. There are numerous Brain Computer Interface applications available in the market, but very little weight is given to Human to Human computer interface, due to poor signal capturing mechanisms as well as human nature of fear on the testing this solution on another human. Having said that, it is obvious if this sort of research succeed on reducing the time to develop muscle memory, it will definitely have huge commercial success as well.

1.4 Problem in Brief

The Learning is something very crucial for the development of the human race. Humans are not a species that bring their intellect hard coded to their DNA, instead they learn things by observation and repeating the activity until it develop muscle memory.

In this research, an attempt was made to minimize the time associated with the learning and memorizing of these repetitive tasks by using a Human to Human Interface, where it will capture neural impulses from one human and feed to second human, allowing the second human to develop same type of muscle memory.

1.5 Novel Approach to reduce time associated with the Learning

A research has been conducted to identify neural impulses generated when Professional piano player is playing a song. Special EEG capturing device is constructed to capture neural impulses from Median, Ulnar, and Radial nerves of the Piano Player's Hands. The EEG signals were captured using the non-invasive methods. The captured signal is then processed through filter, where it will remove all the noises and unwanted EEG signals/muscle Contractions generated due to Blood Flow, etc. These processed signals then feed to TENS device, where it generate artificial stimulation as form of electric current. This new feed of current then applied to trainee's Median, Ulnar, and Radial nerves using the passive techniques. Once the signal applied to respective nerves on the second human (trainee), his hands will be moved according to the trainers hand movements.

1.6 Structure of the Thesis

Rest of the thesis is structure as follows. Chapter 2 critically review the domain of Brain Computer Interface/Functional Electrical Stimulation & Transcutaneous electrical nerve stimulation(TENS), by highlighting current solution, practices, technologies, limitations defining the research problem. Chapter 3 described essentials of EEG and Artificial Muscle Stimulations. Chapter 4 present our Approach to Develop Muscle Memory using Human to Human Interface.

Chapter 5 is on the design of H2H Interface application to capture EEG signals and generate stimulations. Chapter 6 contains details of implementation of the system which captured the EEG signals and generating of Artificial Muscle Stimulation. Chapter 7 reports on evaluation of the new solution by explaining the evaluation strategy, participants, data collection, data representation and data analysis. Chapter 8 concludes the outcome of the research with the note on further work.

1.7 Summary

This chapter introduced to the problem, surrounding issues on this research area, objectives and listed the Aims & Objectives of this project, then the novel solution. Next chapter will be on literature review of Brain Computer interface, Functional Electrical Stimulation & Transcutaneous electrical nerve stimulation (TENS) with a view to define the research problem.

Chapter 2

Emergence of Human Computer Interfacing

2.1 Introduction

This chapter explains the background for this research, and identifies the main research questions and methods to bring clarity and define the projects focus, based on lessons learned from earlier research efforts and new anticipations on the related.

2.2 Brain Computer Interface (BCI)

A Brain Computer Interface [1] is a direct communication pathway between the brain and external device, which often use to assisting, repairing human cognitive or sensory-motor functions. Initial research has conducted in early 1970s at the University of California, since then BCI research and development has since focused primarily on the Neuro-prosthetics applications that aim at restoring damaged hearing, sight and movement. Due to the nature of Cortical Plasticity of the brain, signals from the implanted prostheses can be handled by the brain like natural sensor. After years of experiments, finally in mid-1990's first Neuro-prosthetic device implanted in human.

In 1924, Hans Berger's discovery of the electrical activity of the human brain and the development of electroencephalography (EEG). Berger was the first to record human brain activity by means of EEG. EEG signal discovery made a huge wave in the related subject area, where researcher were interested to use this new technology to diagnose brain diseases using EEG wave diagrams, by opening completely new possibilities.

The term Brain Computer Interface (BCI)[1] was introduced by Prof. Jacques Vidal, where he published term in his publications. Later he was recognized as the Inventor of BCI. His first BCI applications were to allow users to control cursor direction. However after his early contributions to BCI community, he went silent for many years, until he gave a lecture in 2011.

There are two types of Brain Computer interface technologies available, and Invasive BCIs are implanted directly into the grey matter of the brain during neurosurgery, whereas Non-Invasive BCIs are capture brain signals by the electrodes place on the surface of the human head.

The Non-Invasive BCIs are implants produce poor signal resolution because the skull dampens signals, dispersing and blurring the electromagnetic waves created by the neurons. Although the waves can still be detected it is more difficult to determine the area of the brain that created them or the actions of individual neurons.

The EEG is the most widely used brain computer interface technology, due to the fact it is a non-invasive, low set-up cost, portability & ease of use. Only drawback of this technology is, EEG signals are highly susceptible to noise.

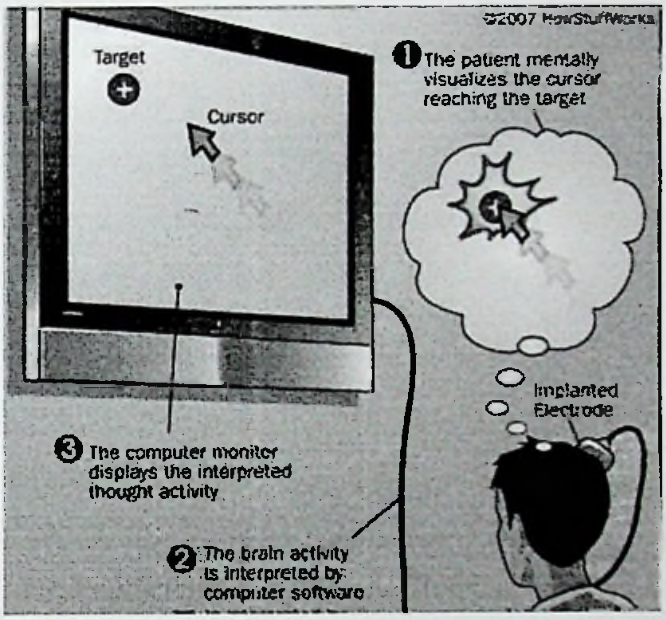


Figure 2.1 – Brain Computer Interface to Control Cursor

There are three types of Brain Computer Interfacing methods, namely Non – Invasive, Partially Invasive & Invasive.

2.2.1 Invasive Brain Computer Interface

In an Invasive Brain Computer Interface, Signals captured from Electronic sensor will be directly feed to relevant part of the Brain, and also Brain Signals are captured directly from the Brain. Usually Invasive Brain Computer Interface implantation requires Medical Surgery (Deep Brain Stimulation) which can implant devices to Brain itself.

Deep Brain Stimulation (DBS) is a special neuro-surgical procedure which introduced in 1987. During this procedure, special medical device called a Neuro-Stimulator implanted in the grey matter of the brain. The Neuro-Stimulator is capable of sending electrical impulses through implanted electrodes to specific areas in the brain. DBS is widely used in treatment of Parkinson's disease, essential tremor, dystonia, chronic pain, major depression and obsessive–compulsive disorder. However, Due to the complexity of brain, Invasive Brain Computer interfacing using DBS has no clear instructions on how to perform a DBS. Usually DBS is administered through trial and error basis and it is one of the few blinded experiments allowed.

Vision

Invasive BCI research has predominantly focused repairing damaged sight and providing new functionality for people with paralysis. Invasive BCIs are implanted directly into the grey matter of the brain during neurosurgery. Because they lie in the grey matter, invasive devices produce the highest quality signals of BCI devices but are prone to scar-tissue build-up, causing the signal to become weaker, or even non-existent, as the body reacts to a foreign object in the brain. Therefore once the BCIs are implanted in the brain, there is a need for continuous monitoring and cleaning the sensors, to prevent BCI itself from damaging the grey matter in the brain.

Motor Actions

BCIs focusing on motor Neuro-Prosthetics aim to either restore movement in individuals with paralysis or provide devices to assist them, such as interfaces with computers or robot arms. This is one of the most popular application of BCI, especially these type of researched were funded mainly by US Government during War Times, to treat their soldiers.

2.2.2 Partially - Invasive Brain Computer Interface

Partially - Invasive BCI devices are implanted inside the skull but rest outside the brain rather than within the grey matter. They produce better signals than non-invasive BCIs where the bone tissue of the cranium deflects and deforms signals and have a lower risk of forming scar-tissue in the brain than fully invasive BCIs. There has been preclinical demonstration of intercortical BCIs from the stroke perilesional cortex.

2.2.3 Non - Invasive Brain Computer Interface

There have also been experiments in humans using non-invasive neuroimaging technologies as interfaces. The majority of published BCI work involves non-invasive EEG-based BCIs. Non-invasive EEG-based technologies and interfaces have been used for a much broader variety of applications. Although EEG-based interfaces are easy to wear and do not require surgery, they have relatively poor spatial resolution and cannot effectively use higher-frequency signals because the skull dampens signals, dispersing and blurring the electromagnetic waves created by the neurons. EEG-based interfaces also require some time and effort prior to each usage session, whereas non-EEG-based ones, as well as invasive ones require no prior-usage training.

2.3 Functional Electrical Stimulation

Functional electrical stimulation (FES)[2] is a technique that uses electrical currents to activate nerves affected by paralysis resulting from spinal cord injury (SCI), head injury, stroke and other neurological disorders, where can often be used to help someone stand up, or to improve walking by helping to swing a foot forward. FES plays vital role to restore motor activities of human disabilities.

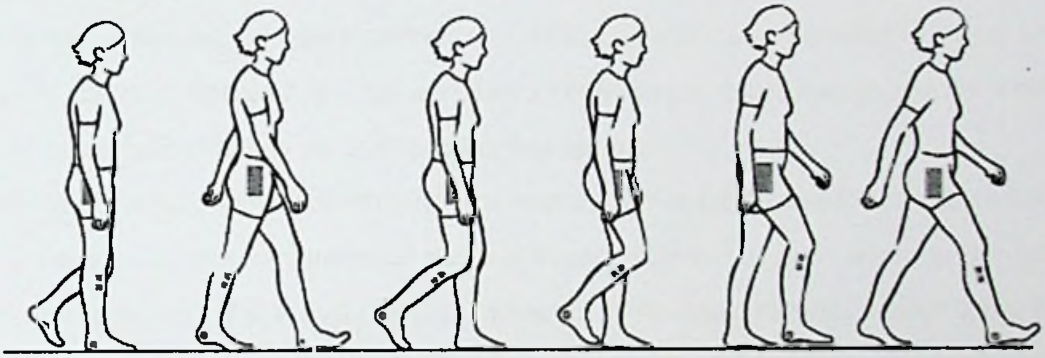
Neurons are like electrical storage. The electric potential in neuron's tissue lead to the polarization/depolarization activities which induce firing of action potentials. Above mentioned FES devices can be used to activate muscles or other nerves. It is worth mentioning here that FES devices are must be used with extreme care as passing electric current through nervous tissues can lead to adverse effects such as cell deaths.

The most common issue treated using FES is known as dropped foot, where in drop foot, person who is having this having inability to lift the foot and toes when walking, causing foot to drag. This illness is caused by weakness of the muscle that lift the foot and excessive tightness of the muscles. Using FES, stimulation is given to foot muscle when walking resulting the intended behaviour. Electrical stimulation can help people to walk faster, with less effort and with more confidence. Stimulators are continuing to be developed, computer technology is allowing them to be more finely controlled and more muscle groups can be stimulated to produce a more natural walking pattern.

A Functional Electrical Stimulation device consists of a stimulation device, which is either worn on patient's leg or around patient's waist. A Special sensor fixed on the patient's shoes which connected to device, which detects the pressure changes as the foot starts to move, then electrodes attached to the skin on the leg which apply the electrical signal to the leg. This electrical signal helps patient to lift the foot to the correct angle for taking the step and prevent the foot from dragging on the floor.

There are highly sophisticated FES sensors are available that uses a nerve stimulator that can be surgically implanted on the leg. However this surgically implanted versions are very new to the industry, therefore patients are reluctant to select them over the version available to wear. Therefore presently FES version that are worn are much more commonly used. As well as being a treatment for drop foot syndrome, FES can also be used in rehabilitation, physiotherapy techniques, in order to assist with movements in muscles.

As discussed above treating drop foot using FES is straight forward process, where treating for inabilities in Arms are quite difficult, due to complex neural system in the hand.



When the foot is standing, there's no stimulation.

But the heel switch detects that the foot is rising...

...and sends stimulation to the electrodes on the calf...

...this makes the foot lift correctly and not drag...

...as it swings to step forward.

When the heel hits the ground stimulation is switched off until the next step.

Figure 2.2 – FES on Foot Drop Syndrome

Who can use FES?

For Functional Electrical stimulation to be successful, both nerve and the muscle must be un-damaged. As a result of that, FES can be used on conditions such as;

- Stroke
- Multiple sclerosis (MS)
- Spinal cord injury, T12 and above (SCI)
- Parkinson's disease
- Cerebral palsy (CP)
- Head injury (HI)
- Familial or hereditary spastic paraparesis (FSP/HSP)

FES is not suitable for lower motor neurone conditions such as:

- Peripheral nerve lesions
- Polio
- Motor neurone disease
- Guillain-Barre Syndrome

Side effects of FES

The Electrical Stimulation causes a tingling 'pins and needles' or buzzing sensation on the skin. In order to reduce this, it's vital to keep the electrodes on the correct position and apply electrical signals only to relevant muscles.

However some people can't tolerate the effect of electrical stimulation, and some people are quite fine with it. This is in fact an experimental technology, where it could have cause further issues on ability to control nerves.

Very occasionally some patients find that stimulation or the electrodes causes irritation of their skin. In fact, unconfirmed medical reports claim that even seizures can occur due to frequent Electrical Stimulations. However it was again mentioned that using FES technology combined with the Brain Computer Interfaces, seizures can be prevented even before it started.

2.3.1 Electrical Activation of Human Muscle

An Electrical pulse applied to nerves can introduce action potentials. The stimulate electrode creates an electrical field that depolarizes the membranes of nearby neurons. If the depolarization reaches a critical threshold, an influx of sodium ions from the extracellular space to the intracellular space produces an action potential that propagates in both directions away from the site of stimulation. Action potentials that propagate proximally within the peripheral nerve axons can ultimately be destroyed at the cell body, and action potentials that propagate distally will be transmitted across the neuromuscular junction and cause muscle fibres to contract. In general, large-diameter axons (which innervate the larger motor units) are activated with less current than little axons as a result of the broader spacing between nodes of Ranvier in massive axons produces larger induced transmembrane voltage changes.

Electrical activation of neuro tissue needs a minimum of two electrodes to produce a current flow. There are two types of electrode arrangement known as Monopolar and Bipolar configuration. Regardless of the configuration, One Electrode is always referred to as Active Electrode, which place near the nerve to be stimulated. However in Monopolar Stimulation, the other electrode is known as in-different or return electrode, which place in the less excitable tissues. Usually reference electrode has a larger surface area than the active electrode. In Bipolar stimulation, the reference electrode is placed near the active electrode. In bipolar stimulation, the reference electrode is placed close to the active electrode. Multichannel monopolar systems cut back the amount of electrodes and leads needed by using only one remote return electrode with many active electrodes placed close to motor points or nerves targeted for excitation. In multichannel bipolar systems, every active electrode has its own return electrode, requiring additional leads; however, bipolar stimulation might permit greater property of activation as a result of each electrode combine creates an additional localized electric field.

2.4 Transcutaneous Electrical Nerve Stimulation

Transcutaneous Electrical Nerve Stimulation (TENS) is the method of using electric current to stimulate the nerves for therapeutic purposes. Mainly this technology has been used to treat pains. Usually this TENS devices comes with two or more electrodes which can be connected to skin. A typical battery-operated TENS unit is able to modulate pulse width, frequency and intensity. Generally TENS is applied at high frequency (>50 Hz) with an intensity below motor contraction (sensory intensity) or low frequency (<10 Hz) with an intensity that produces motor contraction.

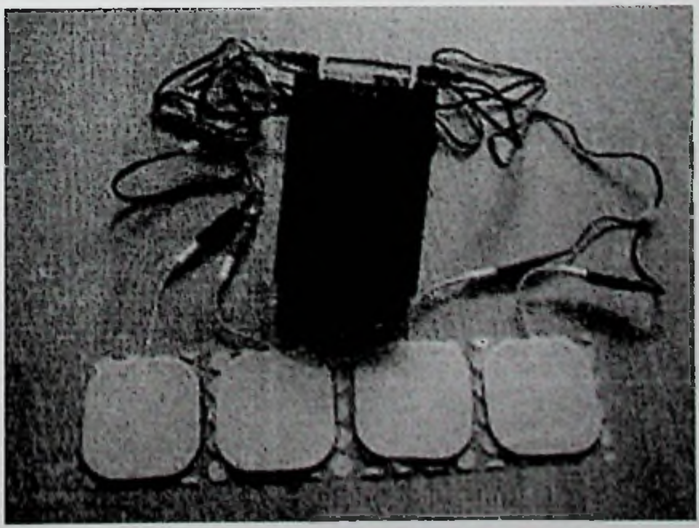


Figure 2.3 - A four-lead TENS unit.

Loss of control of muscles leads to a number of clinical problems, many of them related to long-term changes in the muscles and paralysis. Doctors often use transcutaneous electrical nerve stimulation (TENS) delivered via electrodes applied temporarily to the skin over various nerves and muscles [4]. The electrically induced muscle contractions are effective in maintaining muscle strength but the stimulation also tends to excite the sensory nerves of the skin, often producing unpleasant sensations.

This problem can be overcome by implanting the stimulation electrodes [5] near the nerve branches that innervate only the muscles but the surgery can be tedious and the wires are vulnerable to breakage from the motion of the muscles. Advances in microelectronics have made it possible to build single channel stimulators that are small enough to be injected into muscles, where they can be powered and individually controlled by radio frequency magnetic fields generated outside the patient. There have been several successful clinical trials of this technology [6].

However though this technology has been used widely to help people with physical inabilities, there were virtual no research has been done to amplify the memory using this simple technology. The major reason for not attempting this kind of research is that according to medical theories, brain/nerves are not something can be implanted, and also very few people, especially Professor Kevin Warwick[7] believe this kind of basic technology[8] can be used to manipulate memory or at-least guide/improve the efficiency of motor actions.

2.5 Summary

This chapter reviewed the history, current trends in the field of Brain Computer interface, Functional Electrical Stimulation & Transcutaneous Electrical Nerve Stimulation (TENS). It discusses little bit of past and current researches in the field and approaches. In the section of Brain Computer Interface, three methods of BCI were discussed and pros/cons of each method were identified.

Chapter 3

Electroencephalography and Artificial Muscle Stimulations

3.1 Introduction

This chapter presents the major technologies associated with this research. Electroencephalography (EEG) signals are the main input to the system. System captures EEG signals using four electrodes from two nerves on the hand. This capturing system uses specialised version of EEG technique called Surface-EEG since EEG signals are captured without penetrating arm with electrodes.

Then captured signal will be processed by specially designed Arduino shield, where it will generate regulated signal to TENS device. According to the signal received by TENS device, it generate electrical pulse, where pulse width, modulation, etc., will be controlled by TENS device. This chapter described those technologies in detail.

3.2 Electroencephalography

The brain has continuously fascinated humans, and notably a German scientist named Hans Berger, who discover electroencephalography (EEG)[9] about eighty years ago. The aim of this project is to research and explore the chances that lies inside the domain of Brain-Computer Interfaces, using user friendly instrumentation that have recently become on the market on the general public market.

The sphere of Brain-Computer Interfaces (BCI)[1] is a driving force for utilizing electroencephalography technology (EEG), which is the method of recording brain activity from the scalp using electrodes. In the past, the main focus have been on developing applications for medicinal purposes, serving to unfit or disabled patients to communicate with the other humans or machines, by mapping brain signals to human psychological feature and/or sensory-motor functions.

EEG waveforms are generally classified using their frequency, amplitude, and shape, as well as from where it has been recorded. However most commonly known classifications is done using EEG Signal frequency (alpha, beta, theta, and delta).

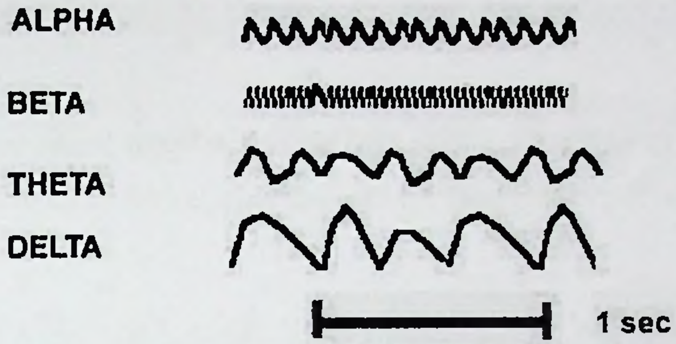


Figure 3.1 – EEG Signals

Frequency is a key criteria used to define normal or abnormal EEG signals. Most EEG signals of 8 Hz and higher frequencies are normal findings in the EEG of an adult who is not in sleep. When humans are fast to sleep the EEG signals are generated from the Brain, falls below the 7 Hz. Having said that, it was noticed that EEG signal frequencies [11] are varies by the age of the human.

There are four commonly identified EEG signal forms;

- Alpha waves - 8-13 Hz
- Beta waves - Greater than 13 Hz
- Theta waves - 3.5-7.5 Hz
- Delta waves - 3 Hz or less

Alpha waves

Alpha waves[11] are neural signal within the frequency range of 8-13 Hz originating from the neural activity of the thalamic pacemaker cells in human brain. These signals are also known as Berger's waves in memory of the founder of EEG, Hans Berger. This is the first type of brain wave observed either by electroencephalography (EEG) or magnetoencephalography (MEG), which mostly seen originating from the occipital lobe during the wakeful relaxation. Alpha waves commonly seen in all age groups and most commonly in adult EEG signals. They occur rhythmically on both sides of the head and frequency of the waves captured on the non-dominant side of the brain is far higher than the other side.

However it was noticed that alpha signals are prominent when the human closed eyes and relaxing. Alpha signals disappears normally with attention, arithmetic operations done in mind or stressed.

Beta waves

Beta waves[11] are neural signal within the frequency greater than 13 Hz originating from the neural activity of the cells in human brain. Beta waves can be further divided into three parts.

- Low Beta Waves (12.5–16 Hz, "Beta 1 power")
- Beta Waves (16.5–20 Hz, "Beta 2 power")
- High Beta Waves (20.5–28 Hz, "Beta 3 power")

Beta waves can be seen on all age groups and they tend to be small in amplitude but symmetric. Usually beta waves can be faded/distorted when person in drugs. However when a person is associated with active, busy or anxious thinking and active concentration, then the beta signals from all three categories can be seen.

Over the motor cortex beta waves are associated with the muscle contractions that happen in isotonic movements and are suppressed prior to and during movement changes

Theta waves

Theta waves[11] are neural signal within the frequency range between 3.5-7.5 Hz originating from the neural activity of the cells in human brain. Theta and delta waves are known as slow waves. Theta waves can be seen in sleep at any age. This type of waves can be seen in when humans are awake as well, but excess amount of waves can be a problem. Too much theta waves make people prone to depression and make them highly suggestible for external causes, based on they are deeply relaxed and semi-hypnotic state.

Delta waves

Delta waves[11] are normally very slow type of wave, with frequency of 3Hz or less, but the largest amplitude. They normally are seen in deep sleep in adults, infants and children. As the people grow old, less delta waves produced even during the deep sleep. This Delta waves are responsible of unconscious bodily functions such as regulating heart beat and digestion. However if the delta waves are observed in awake, it can be a symptom of serious issue on the brain.

3.3 EEG Tests

In order to identify the type/frequency of the above mentioned EEG signal waves, there are four types of EEG tests, which carried out.

Routine EEG

A routine electroencephalogram recording lasts for about twenty to forty minutes. Throughout the check, patient be asked to rest quietly and open or close eyes from time to time. In most cases, patient even be asked to breathe in and out deeply (known as hyperventilation) for a couple of minutes.

At the end of the procedure a flashing light may be placed nearby to check if this affects his/her brain activity.

Sleep EEG or sleep-deprived EEG

A sleep electroencephalogram is administered while patient asleep. It may be used if a routine electroencephalogram does not provide enough information, or to check for sleep disorders.

In some cases, Patient is also asked to remain awake the night before the test to help ensure patient can sleep whereas it's administered. This is referred to as a sleep-deprived electroencephalogram.

Ambulatory EEG

An ambulatory electroencephalogram is where brain activity is recorded throughout the day and night over a duration of one or more days. The electrodes will be hooked up to a small transportable electroencephalogram recorder which will be clipped on to patient's clothing.

Patient can continue with most of his/her normal daily activities while the recording is being taken, although he'll need to avoid getting the instrumentation wet.

Video telemetry

Video telemetry, also referred to as video electroencephalogram, is a special kind of electroencephalogram where you are filmed while a recording is taken. This can help provide a lot of data regarding patient's brain activity. The test is typically administered over many days while staying in a purpose-built hospital suite.

The electroencephalogram signals are transmitted wirelessly to a laptop. The video is also recorded by in the computer and kept under regular surveillance by trained staff.

3.4 Transcranial direct current stimulation (tDCS)

Transcranial Direct Current Stimulation (tDCS), is a non-invasive, painless brain stimulation treatment that uses direct electrical currents to stimulate specific elements of the brain [12][13]. A constant, low intensity current is applied to two electrodes placed over the head which modulates neuronal activity. There are two kinds of stimulation with tDCS: anodal and cathodal stimulation. Anodal stimulation acts to excite neuronal activity whereas cathodal stimulation inhibits or reduces neuronal activity.

Although tDCS continues to be an experimental sort of brain stimulation, it probably has many advantages over alternative brain stimulation techniques. It is low-cost, non-invasive, painless and safe. It is also straightforward to administer and the instrumentation is easily moveable. The most common side effect of tDCS is a slight itchiness or tingling on the scalp. Several studies suggest it's going to be a valuable tool for the treatment of neuropsychiatric conditions similar to depression, anxiety, Parkinson's disease, and chronic pain. Research has also demonstrated psychological feature improvement in some patients undergoing tDCS. Currently, tDCS is not an FDA-approved treatment.

Transcranial direct current stimulation works by causing constant, low direct current through the electrodes. Once these electrodes are placed inside the region of interest, this induces neural structure current flow. This current flow then either can increase or decreases the neuronic excitability within the specific space being stirred supported which kind of stimulation is being used. This transformation of neuronic excitability results in alteration of brain operate, which might be used in numerous therapies as well as to provide lots of information regarding the functioning of the human brain.

Transcranial direct current stimulation could be a comparatively simple technique requiring solely a couple of elements. These include two electrodes and electric battery powered device that delivers constant current. management code also can use in experiments that need multiple sessions with differing stimulation varieties so neither the person receiving the stimulation nor the experimenter is conscious of which kind is being administered. Every device has an anodal, charged conductor and a cathodal, negative conductor. Current is delineate as flowing from the positive anode, through the intervening conducting tissue, to the cathode, creating a circuit. Note that in ancient electrical circuits created from metal wires, current flow is made by the motion of charged electrons, that really be due cathode to anode. However, in biological systems, like the top, current is typically created by the flow of ions, will|which may|which might} be completely or negatively charged—positive ions will flow towards the cathode; negative ions can flow toward the anode. The device might management this additionally because the length of stimulation.

3.5 Arduino

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on a computer, used to write and upload computer code to the physical board [15].

The Arduino platform has become quite popular with people simply beginning out with electronics, and for good reason. In contrast to most previous programmable circuit boards, the Arduino doesn't need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

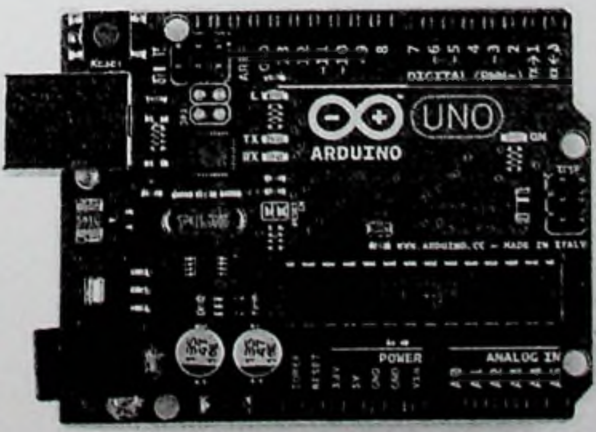


Figure 3.2 – Arduino UNO board

The Uno is one of the more popular boards in the Arduino family and an excellent choice for beginners. In addition, there are these things known as shields – essentially they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities – controlling motors, connecting to the internet, providing cellular or alternative wireless communication, controlling an LCD screen, etc.

3.6 Summary

Throughout this chapter we discussed technologies which are used to create Human to Human Interface. EEG technology, FES/TENS technology and Arduino was discussed in this chapter.

Chapter 4

Approach to Develop Muscle Memory using Human to Human Interface

4.1 Introduction

This idea is inspired by the Human to Human Interface which can be created with Microneurography & Functional Electrical Stimulation where it can transfer neural impulses captured from median nerve of one human to another allowing second human to perform motor actions exactly same as the first human. This allow second human to develop muscle memory on his/her brain to perform motor actions same as the first human. In this project, neural impulses of professional piano player is captured and feed to student, where proposed system guide the learning process by developing the effective muscle memory.

This chapter is about the approach taken to develop muscle memory on Student's brain using the Neural Impulses generate on Teacher's brain. System input, system output and the process are discussed with the non-functional requirements of the system.

4.2 Hypothesis

- Developing muscle memory is a time consuming process, which generally involves with repetition of the task.
- Neural Impulses of the Teacher is use to stimulate Student's Nerves which allowing Student's brain to develop muscle memory, which reduce the time of the learning process.

4.3 Inputs

As inputs to the system, Neural Impulses of Professional piano player captured during the piano session where player will be playing predefined notation. The Neural impulses will be captured from the Median nerve of the right hand and those signal stream will use as inputs for the system.

4.4 Output

System will modify the inputs according to the frequency/pulse width to suite to destination human, and this modified signal will use as output neural impulse which is ready to feed to Student Piano Player.

4.5 Process

Proposed solution consists of following technologies;

01. Transcutaneous electrical nerve stimulation (TENS)
02. Electrodes
03. EMG Cables
04. Hardware/Software based convertor

First set of electrodes will capture Neural Impulses of human and convert them to Electrical current pulses. Then Convertor will use to sync these electrical current pulses to second human neural impulse's frequencies. After these newly synced electrical pulses will send through transcutaneous electrical nerve stimulation device to generate neural impulses matching to second human.

Figure 4.1 – Proposed System

4.5.1 Neural Impulse Capturing System

Two Multielectrodes (tf-LIFE4) implanted on the skin directly above the median nerve allowing to capture muscle contractions of the nerve as form of EMG signals.

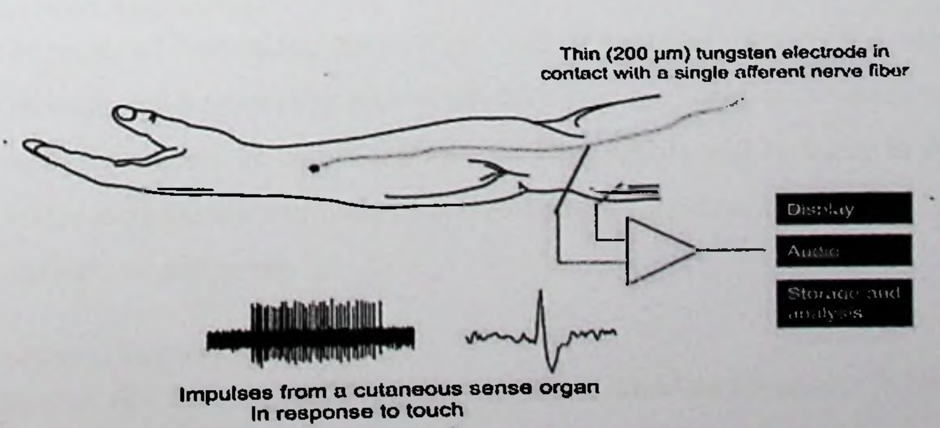


Figure 4.2 – Proposed Neural Impulse Capturing System

4.5.2 Signal Processing unit

The Signal Processing Unit is responsible for adjusting signal originating from the Electrodes to match the characteristics of the destination human neural system (Frequency, Pulse Width, etc.). This system also will be used as a filter to filter out the unwanted signals detecting using the Non-Invasive methods, such as blood flow, etc.

4.5.3 TENS device

Transcutaneous Electrical Nerve Stimulation (TENS) is the method of using electric current to stimulate the nerves for therapeutic purposes. Mainly this technology has been used to treat pains. Usually this TENS devices comes with two or more electrodes which can be connected to skin. A typical battery-operated TENS unit is able to modulate pulse width, frequency and intensity. Generally TENS is applied at high frequency (>50 Hz) with an intensity below motor contraction (sensory intensity) or low frequency (<10 Hz) with an intensity that produces motor contraction. TENS device takes processed neural impulse which coming from the Signal Processing Unit as Input to the system. Then TENS device amplify the signal by changing the characteristics of the signal (Frequency & Pulse Width) using its pre-programmed gauges.

4.5.4 Neural Impulse Feeding System

The Non-Invasive Brain Computer Interface methods will be used to feed the Neural Impulse back to Human Arm. The Two Multielectrodes (tf-LIFE4) implanted on the skin directly above the median nerve allowing to feed the neural impulses to make muscle contractions. The Non-Invasive method is selected due to the ease of usage as well as to reduce the complexity of the Neural Impulse Feeding System.

4.6 Suggested Application

Neural Impulses of Professional piano player will be captured from median nerve and will go through signal processing system which convert the signal to the characteristics matching to the student on end of the system. This signals will be taken as input to TENS device and generate artificial neural impulses. These neural impulses will be feed to the student's median nerve.

4.7 Functional Requirements

The suggested system must be able to convert neural impulses accurately in real time. However input signal must be converted to match the destination human's frequency. On top of the functionality, system must be easy to use, easy to caliber by user with limited knowledge in the relevant technical areas. To ease the calibration of the system, some sort of indicator/guidance method will be used, therefore users can rely on the indicator to caliber the system properly.

4.8 Users

There can be plenty of scenarios that this can be used. In fact, any human activity that rely on the muscle memory can be used to evaluate this. However in this project, experiment will be restricted to students who are interested in learning piano and to a professional piano player. Learning Piano is a time consuming process, where Student has to observe their teacher and imitate the steps. In this system, Student will be receiving neural impulse directly from the teacher where it will guide and help the student to develop related muscle memory more quickly.

4.9 Summary

This chapter discussed about the approach for design and implement the Development muscle memory using Human to Human Interface. Input, output are given along with the process which will be carried upon and also users of the system and non-functional requirements are also discussed. In here, the project application was restricted to Piano learning activity to have measurable the evaluation process.

Chapter 5

Analysis & Design of H2H Interface Program

5.1 Introduction

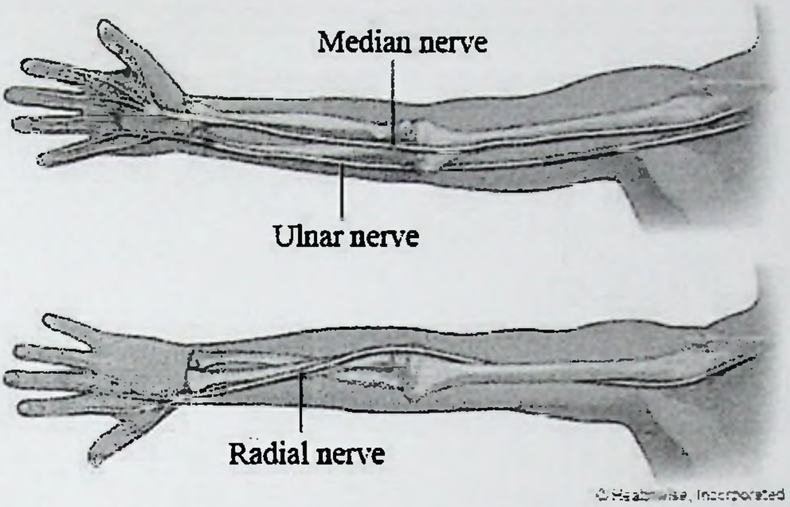
The method used to designing of this project will be discussed on this chapter which detail about designing of the Human to Human Interface. This chapter provides a detailed explanation of all methods used in the analyses and experiments, as well as the reasons behind using those methods.

The Research work was divided into Anatomical aspects of Human Arm, EEG Input/output and data processing. A several experiments were conducted to identify and detect EEG Signal using EEG electrodes. After that, Converting all the acquired data in the form of target human specification using TENS device.

5.2 Human Nerves

The nerves of the arm and hand perform a substantial two-fold role: commanding the intricate movements of the arms all the way down to the dexterous fingers, while also receiving the vast sensory information supplied by the sensory nerves of the hands and fingers. The movements of the arms must be fast, precise, and strong to complete the diverse activities the body engages in throughout the day. Even the tiny hand muscles, which perform very delicate and precise movements, are driven by about 200,000 neurons. Rapid conduction of sensory nerve signals from the hands provides critical information to the brain and feedback during precise activities. Starting in the trunk of the body, the nerves of the arm and hand arise from the cervical and thoracic regions of the spinal cord as spinal nerves. These nerves merge to form a network called the brachial plexus before continuing into the arm. Five major nerves extend from the brachial plexus into the arm: the axillary, musculocutaneous, median, radial, and ulnar nerves. Each of these nerves carries information in the form of nerve impulses to and from a particular region of the arm and hand. Some of these impulses are sent from various parts of the brain and spinal cord; some come from sense organs located in the joints, ligaments, and tendons; and some come from nervous tissue in the muscles themselves.

In this research, Surface EEG Signals will be captured from Median Nerve & Ulnar Nerve of the hand, since those nerves are covering all five fingers on the hand and those nerves are in a position where EEG signals can be captured very easily.



© Healthwise, Incorporated

Figure 5.1 – Nerves on the Human Arm

5.3 EEG Input/output

This section discuss how EEG Signals captured from the Human Arm. Following diagram shows, basic setup we used to capture the neural impulses from the human arm.

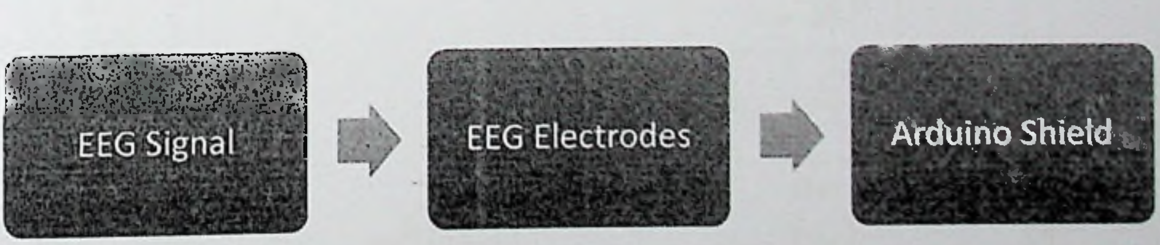


Figure 5.2 – EEG I/O System

To capture neural impulses on both Median and Ulnar nerves, four electrodes will need to used, where they were placed predefined locations on the human arm. These electrodes capture neural impulses and then transferred them to Arduino board for interpretation. Arduino board does not accommodate EEG connectors, therefore

Arduino shield will be created to handle this type of connection and interface them to Arduino. Arduino shield will be created using widely used EEG caputing samples diagrams available on research papers.

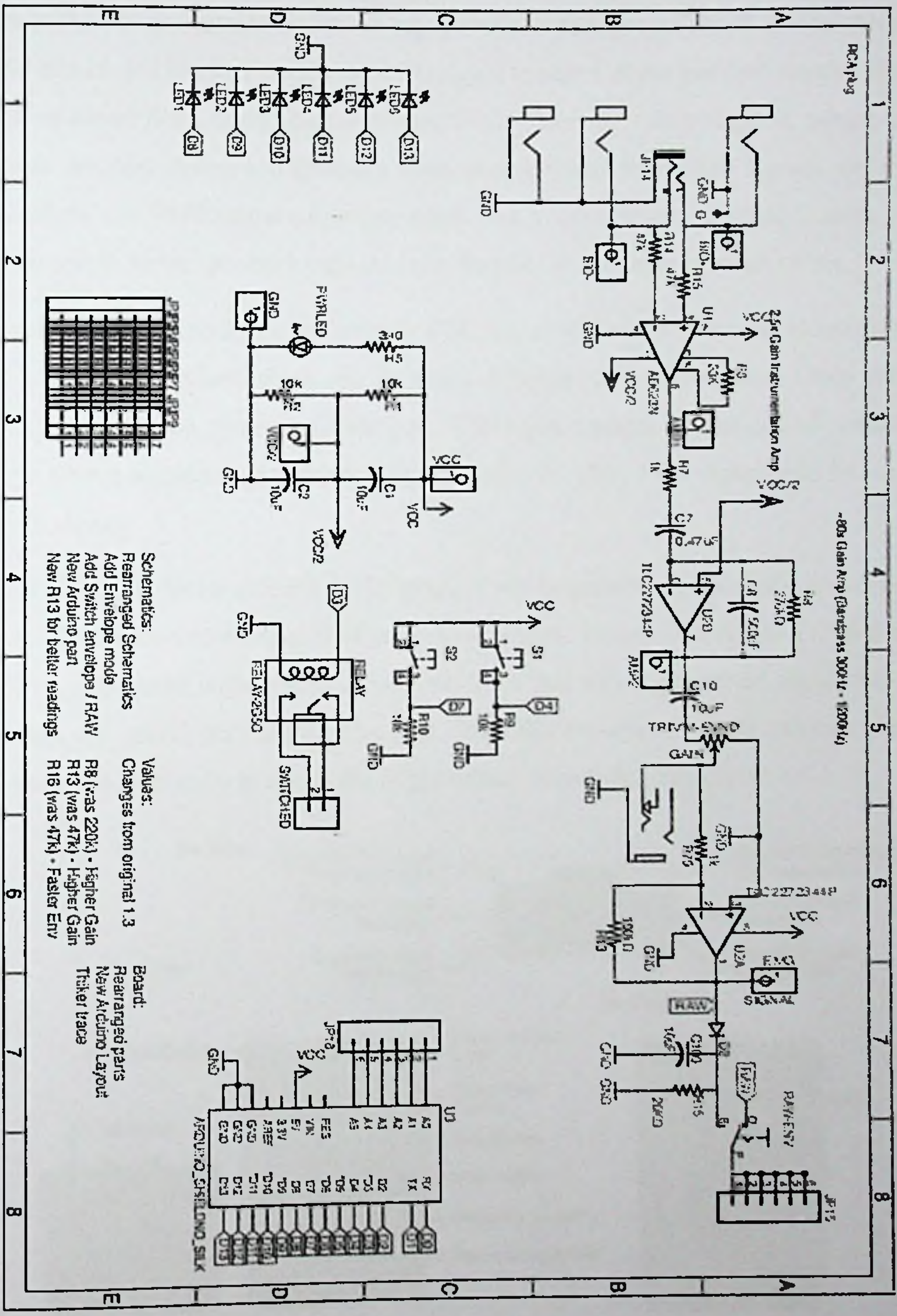


Figure 5.3 – Arduino EEG Shield

5.4 Data Processing

In this section of the application, EEG signals will be identified, translated and will generate input to TENS device, so that TENS device can stimulate muscle of the target human. First part of the data processing is done using the Arduino board, and second part of the data processing is done using the built-in features of TENS device. Arduino EEG Shield will be used modify the EEG signal to match & cut back un-wanted signals such as blood flow, small muscle contractions. There are some humans, whose EEG signals are very strong and there are some people where their EEG signals are week. Therefore, the EEG signal saturation mark was implemented, so when human EEG signal hits to minimum mark only Arduino instruct Target human to stimulate.

Arduino board is configured to identify EEG signal and classify the signal into 100 to 400 range. EEG cut-off mark will be decided based on historical data. Once cut-off mark value decided, system will compare EEG input strength against cut-off value and if the source signal range is more than the cuff-off value, EEG signal will be sent to TENS device.

Once TENS device receive the EEG signal, it will be generating the signal and direct it to the Human nerves we identified on earlier chapter. However, initially TENS device will be configured with low frequency value so that newly generated signal will not damage the nerves on the target human. Once the experiment starts, we can slowly increase the frequency to match the target human biological frequency.

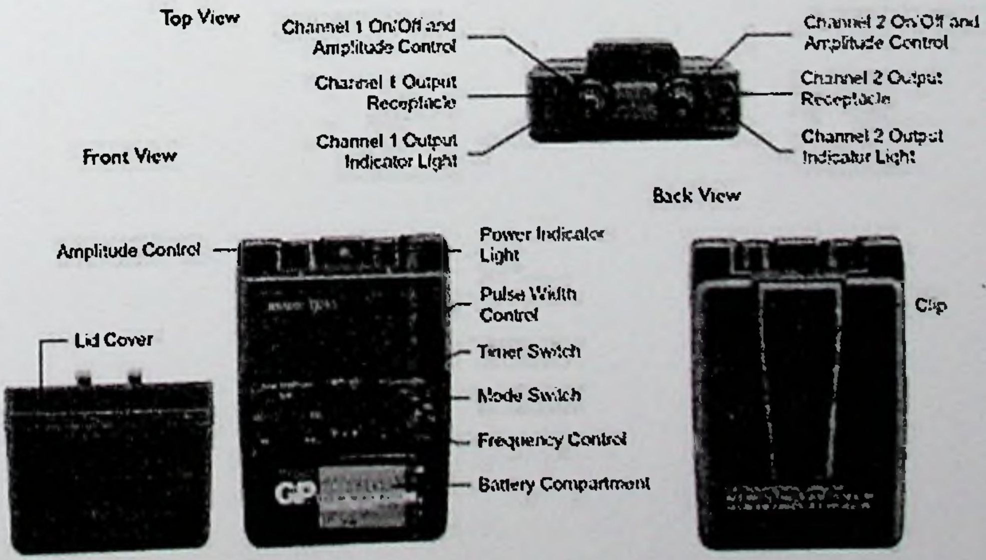


Figure 5.4 – TENS Device

5.6 Summary

In this chapter discussed about the human nerve system and identified the nerve to stimulate to generate action potentials in each muscle. Then EEG signal detection, identification & classification was discussed. To capture EEG signals, Arduino EEG Shield was proposed to use due to its commercial availability. At the end of this chapter, TENS device was explained in detail and its operational flow. The TENS device identified as the best way to induce electrical signal to muscle using non-invasive methods.

Chapter 6

Implementation of H2H Interface Program

6.1 Introduction

This Chapter discuss how implementation was carried out, providing a detailed explanation of the implementation and tools used. The Implementation was divided into three steps, where detecting the EEG signal, analysis of the EEG signal and generating artificial stimulation using TENS Device. An Electrodes will carefully place directly above the Median Nerve, and will capture Neural Impulses of human and convert them to Electrical current pulses. Then Convertor will use to adjust these electrical current pulses to frequency of second human's neuro-muscle system. This modified signal will then sent through Transcutaneous Electrical Nerve Stimulation device to generate neural impulses matching to second human. This process will be explained further in on this chapter.

6.2 Capturing EEG Signal

This section discuss how EEG Signals are captured from the Human Arm. During the Analysis and Design, it was planned to capture Neural Impulses from both Median & Ulnar nerves. However to ease the implementation and reduce the complexity of the project, it was decided to restrict EEG signal capturing only to Median Nerve. To capture neural impulses on Median nerve, two electrodes were used and they were placed predefined locations on the human arm intercepting Median nerve. These electrodes capture neural impulses and then transferred them to Arduino EEG Shield via Arduino UNO board for interpretation. The Arduino UNO board does not accommodate EEG connectors, therefore Arduino shield was used to handle this type of connection and interface them to Arduino. Earlier plan was to create the EEG Arduino Shield, but due to commercial availability of the Shield, customized version of EEG Arduino Shield from OpenBCI was used.

Each different human has unique characteristics of their EEG Signals. There are some humans, whose EEG signals are very strong and there are some people where their EEG signals are weak. Therefore, the EEG signal saturation mark was implemented, so when human EEG signal hits to minimum mark only, Arduino instruct Target human to stimulate. Arduino board is configured to identify EEG signal and classify the signal into 100 to 400 range. EEG cut-off mark was decided based on historical data. Once cut-off mark value decided, system will compare EEG input strength against cut-off value and if the source signal range is more than the cuff-off value, EEG signal will be sent to TENS device.

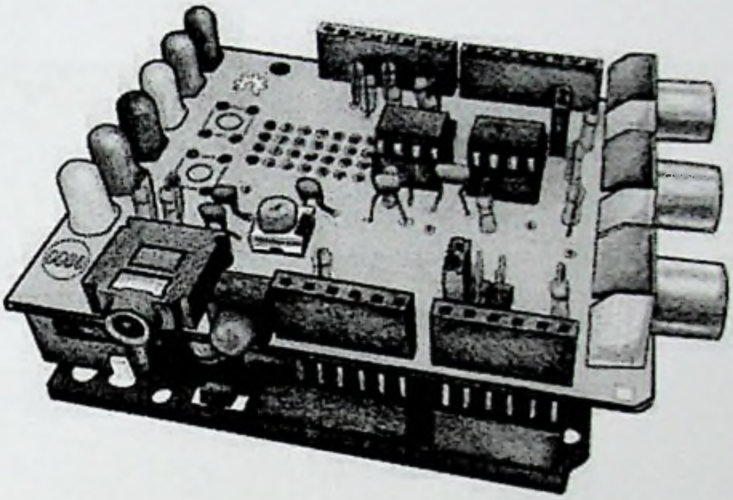


Figure 6.1 – EEG Arduino Shield

There are two types of electrode arrangement known as Monopolar and Bipolar configuration. Regardless of the configuration, One Electrode is always referred to as Active Electrode, which place near the nerve to be stimulated. However in Monopolar Stimulation, the other electrode is known as in-different or return electrode, which place in the less excitable tissues. Usually reference electrode has a larger surface area than the active electrode. In Bipolar stimulation, the reference electrode is placed near the active electrode. In bipolar stimulation, the reference electrode is placed close to the active electrode. Multichannel monopolar systems cut back the amount of electrodes and leads needed by using only one remote return electrode with many active electrodes placed close to motor points or nerves targeted for excitation. In order to capture EEG signals using non-invasive methods, three electrodes were employed using the Multichannel monopolar System. Two Electrodes were actually placed directly above the nerves and one electrode was placed on another area of the arm to detect and compare signal with ground. The Return Electrode was placed on the palm of the hand or sometimes attached to wrist watch. The wrist watch option were seems to be a good option, as it touch the wide range of the skin, where palm could have static electrical energy which may disrupt the reference node, which distort the signals captured from the active electrodes.

Then these electrodes were connected to the EEG Shield, which connected to Arduino UNO board. With the default code comes with the EEG Shield, we can see the EEG Arduino shield's Analogue pin is changing its value when the Muscles are contracted and retracted. However this was rudimentary value, where it cannot be used on our purpose.

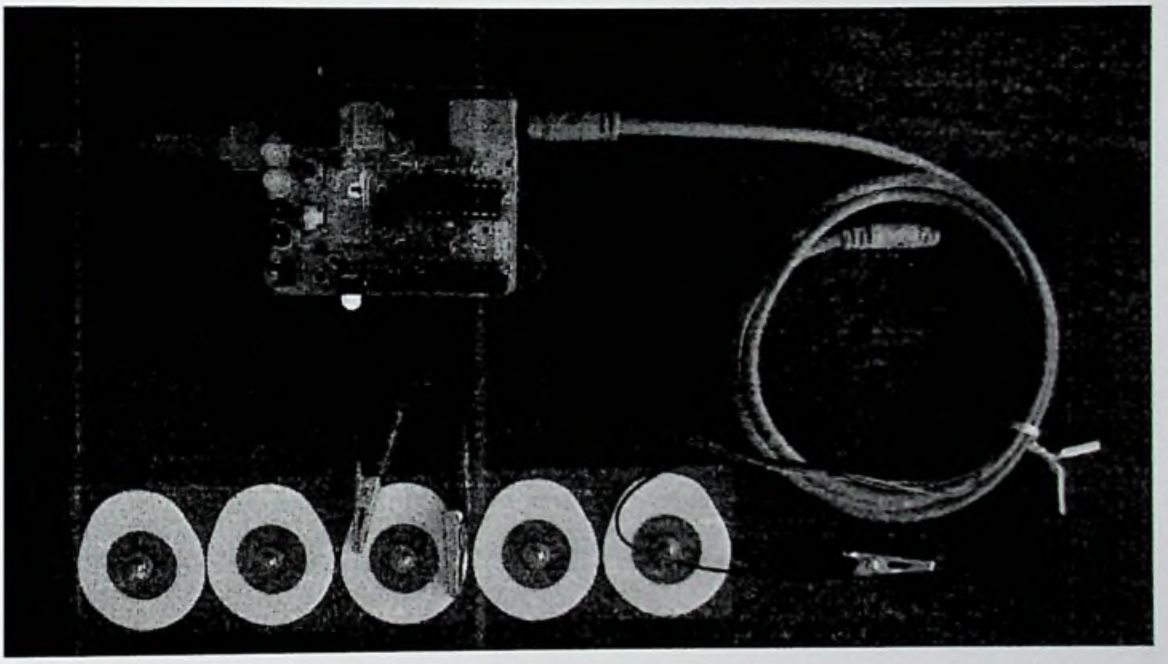


Figure 6.2 – EEG Capturing Configuration

Therefore default Arduino program [Appendix A] which comes with the EEG Shield has been altered to suite our need of generating artificial stimulation. One of the major change done to the default code is, writing the values to USB rather than Digital Output so this signal can be used to plot the graphs for analysis purposes.

Digital to Analogue conversion was done on the Arduino code using standard Arduino library. Arduino board is configured to identify EEG signal and classify the signal into 100 to 400 range.

6.3 Visualization of EEG Signal

The unprecedented increase in computational power over the last 20 years has led to extensive developments of advanced data processing and visualization methods. In the recent years, visualization of non-invasive EEG Signals were rapidly increased due to this. This section discuss the application that created to View and Analyse the EEG signal captured from the Arduino Shield. This is a vital exercise as this visualize the EEG signal to end user. There were plenty of commercial use of the visualization of EEG signals, especially in the medical field. Usually EEG signals are analysed in visual sense to identify the brain activity of a patient. In most cases, EEG visualization is used to identify the patients with brain death as well as the physiological disorders that cannot be diagnosed using the conventional medical procedures.

To visualize the EEG signals, new Windows Application called "EEG Capturing System" has been designed using C#.NET. This application connect to Arduino UNO board using its USB port with Baud rate of 9600 and then it reads the values written to USB from the Arduino EEG Shield using the program mentioned on Appendix A.

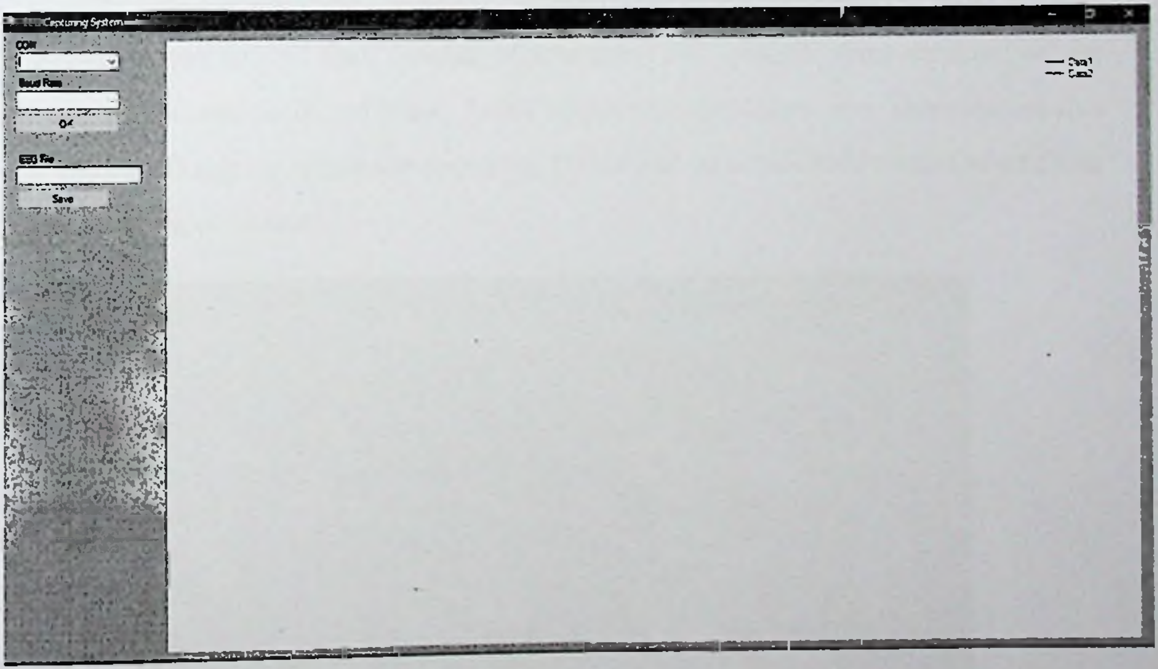


Figure 6.3 – EEG Capture/View Application

This application was designed in a foam that it can save the EEG signal file for future use. Intention was record the EEG signals of one human doing some task and later it can be shipped to somebody in remote location to use it to generate the EEG signal and artificially stimulate the muscles.

However in this research more focus was placed to live capturing of EEG signals from a Human and generate artificial stimulation on second human without storing the captured EEG Signal. Above mentioned EEG Signal storing for future use and generate Artificial Stimulation from the Stored EEG signal is considered as future work.

6.4 Generation of new EEG Signal

This section discuss the steps involved in generating an artificial stimulation on the second human. Earlier in this chapter, separate Arduino program loaded to Arduino UNO board to read the EEG values captured and display them on a graph. However in order to generate artificial Stimulation, earlier captured Analogue EEG Signal needs to be converted to Digital Signal.

It's not possible to extend earlier Arduino program as earlier program write the analogue EEG Signal to USB, but in this case captured Analogue EEG signal must be converted to Digital Signal and send it to TENS device to generate the stimulation. Therefore new Arduino program [Appendix B] is created which read the EEG signals from the Electrodes and convert the signal to Digital Signal.

However it was noticed that, number of unwanted EEG signals were detected on the human arm related to Blood Flow, Small Muscle Contractions, etc. Therefore in this research, EEG signals which are above the 10 Hz will be considered when the artificial signal is being constructed.

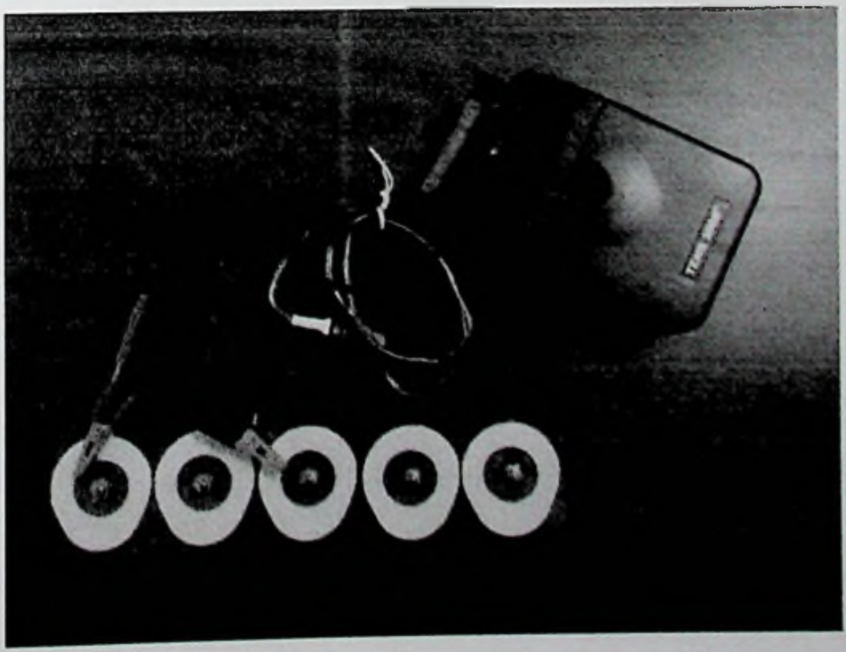


Figure 6.4 – TENS Device Setup

When signal is being transmitted to TENS device from the Arduino Output Pin, it send only a value of the EEG Signal varying from 10 Hz to 25 Hz. However it was noticed that different EEG signals were generated from same motor action performed by different humans. For an example, when Subject 01 was holding a pen, his muscles generated EEG signals varying from 15 Hz to 18 Hz, but when Subject 02 doing the same action, EEG signal was showing 17Hz to 20 Hz frequency range.

Therefore potentiometer in the Arduino board has been used to match the frequency of the Source and Destination Human. Apart from that, TENS device has capability of generating the signal in predefined characteristics. For this experiment, we use default settings on the TENS device as below.

Figure 6.5 – TENS Device Configuration

6.5 Summary

This Chapter discussed how the implementation was carried out, providing a detailed explanation of the implementation and tools used. The Implementation was divided into three steps, where Capturing the EEG signal, Visualization of the EEG signal and Generating artificial stimulation using TENS Device. An Electrodes were carefully place directly above the Median Nerve, and captured Neural Impulses of human and convert them to Electrical pulses. Then Convertor was use to adjust these electrical current pulses to frequency of second human's neuro-muscle system. Then the modified signal sent through Transcutaneous Electrical Nerve Stimulation device to generate neural impulses matching to second human.

Chapter 7

Evaluation

7.1 Introduction

This section discuss about the experiments conducted to evaluate the Human to Human Interface designed. However due to the nature of this experiment, Test Subjects were very hard to find, therefore few Test Subjects only used.

7.2 Experimental Design

This study used 3 test subjects (2 men and 1 women), with an average age of 28 years (the subjects' age ranged between 26 and 30 years).No EEG related training has been provided to the subjects prior to this experiment.

In this experiment, all three subjects were used as the Source of capturing EEG signals, and two men were used to artificially stimulate nerves on the arm. However we restricted the experiment for Median Nerve since Ulnar & Radial Nerve stimulation needs to increase the electrode count to six. To understand and evaluate the EEG Signal captured and artificial stimulation, Piano Playing activity is used. All three Test subjects were having brief experience on Playing a Keyboard, but for this experiment, we used different Keyboard notations for each person.

Following Table depicts the design of the experiment.

Source Human	**Destination Human**	**Notation**	**Duration**	**Method**
Man 01	Man 02	Notation 01	10mins	Human to Human Interface
Man 01	Woman 01	Notation 01	10mins	Human to Human Interface
Man 01	Man 02	Notation 02	10mins	Human to Human Interface
Man 01	Woman 01	Notation 02	10mins	Human to Human Interface

Man 02	Man 01	Notation 03	10mins	Human to Human Interface
Man 02	Woman 01	Notation 03	10mins	Human to Human Interface
Man 02	Man 01	Notation 04	10mins	Human to Human Interface
Man 02	Woman 01	Notation 04	10mins	Human to Human Interface
Woman 01	Man 01	Notation 05	10mins	Human to Human Interface
Woman 01	Man 02	Notation 05	10mins	Human to Human Interface
Woman 01	Man 01	Notation 06	10mins	Human to Human Interface
Woman 01	Man 02	Notation 06	10mins	Human to Human Interface
Man 01		Notation 07	10mins	Manual
Man 01		Notation 08	10mins	Manual
Man 02		Notation 09	10mins	Manual
Man 02		Notation 10	10mins	Manual
Woman 01		Notation 11	10mins	Manual
Woman 01		Notation 12	10mins	Manual
Man 01		Notation 13	10mins	Manual
Man 01		Notation 14	10mins	Manual

Man 02		Notation 15	10mins	Manual
Man 02		Notation 16	10mins	Manual
Woman 01		Notation 17	10mins	Manual
Woman 01		Notation 18	10mins	Manual

Table 7.1 – Experiment Design

For each iteration of the experiment, number of mistakes happened during the performance is recorded.

7.3 Experimental Result

The experiment results were collected for nearly 2 months period, and number of mistakes occurred on the each iteration is recorded. Before starting experiment on each day, test subjects were give a music to listen to and relax their mind and body. This was done to ensure that Test Subjects' daily activities will not introduce any impact on the experiment.

Following table depicts the results of the experiment conducted.

Source Human	Destination Human	Notation	Duration	Method	No of Mistake
Man 01	Man 02	Notation 01	10mins	Human to Human Interface	1
Man 01	Woman 01	Notation 01	10mins	Human to Human Interface	0
Man 01	Man 02	Notation 02	10mins	Human to Human Interface	1

Man 01	Woman 01	Notation 02	10mins	Human to Human Interface	0
Man 02	Man 01	Notation 03	10mins	Human to Human Interface	2
Man 02	Woman 01	Notation 03	10mins	Human to Human Interface	2
Man 02	Man 01	Notation 04	10mins	Human to Human Interface	2
Man 02	Woman 01	Notation 04	10mins	Human to Human Interface	2
Woman 01	Man 01	Notation 05	10mins	Human to Human Interface	0
Woman 01	Man 02	Notation 05	10mins	Human to Human Interface	2
Woman 01	Man 01	Notation 06	10mins	Human to Human Interface	2
Woman 01	Man 02	Notation 06	10mins	Human to Human Interface	3
Man 01		Notation 07	10mins	Manual	2
Man 01		Notation 08	10mins	Manual	2
Man 02		Notation 09	10mins	Manual	2
Man 02		Notation 10	10mins	Manual	2

Woman 01		Notation 11	10mins	Manual	3
Woman 01		Notation 12	10mins	Manual	2
Man 01		Notation 13	10mins	Manual	1
Man 01		Notation 14	10mins	Manual	2
Man 02		Notation 15	10mins	Manual	3
Man 02		Notation 16	10mins	Manual	1
Woman 01		Notation 17	10mins	Manual	1
Woman 01		Notation 18	10mins	Manual	2

Table 7.2 – Experiment Results

7.4 Conclusion

From above experimental results, it was noticed that when the Human to Human Interface deployed for assisting the second human with the Piano Playing, error rate is 17/12 = 1.4167 and when using the conventional learning method is used, error rate is 26/12 = 2.1666. In conclusion, Human to Human interface assisted learning was 32.69% error free than the conventional method of learning.

According to the results set acquired, it was visible that Human to Human Interface is making slight improvement and reducing no of mistakes occurring. However it's visible that even using the assistance of Human to Human Interface, some cases it reported no of mistakes occurring on an iteration is range between 2-3. It was noticed that this was due to error rate of the artificial Stimulated signal.

In this research, non-invasive method is used while recording/capturing the EEG signals, which introduced the signal noise to the original EEG Signal. Also after capturing the EEG signal, we introduced a cap at 10 Hz, where all the signals less than 10 Hz was ignored to reduce the error rate. However this cap, also reduce the accuracy of the EEG signals impacting to the accuracy of the Artificial Stimulated Signal.

7.5 Summary

In this chapter we have evaluated the application using proper experiment mechanism. We have designed and conducted experiment to measure the success of the application and presented the results. The Application has been tested using three test subjects (2 men and 1 women), with an average age of 28 years, and results show rapid increase in the effectiveness of the learning when this interface is used. The Error rate while playing piano using Human-to-Human interface was 1.4167 where conventional method showed rate of 2.1666. In conclusion, Human to Human interface assisted learning was 32.69% error free than the conventional method of learning.

Chapter 8

Conclusion & Further Work

8.1 Introduction

This report presents results that shows that it is possible to build a Human to Human Interface system that allows to capture EEG signals generating from the Motor Actions and feed the Signals to second human that exact same motor actions can be performed, which can eventually develop muscle memory on the second human.

This is accomplished using the Arduino UNO board, Arduino EEG Shield and TENS Device. The EEG Electrodes were placed on the arm to detect EEG Signals generated as a result from the Muscle Contractions happening on the Median Nerve. Then these signals were used to create artificial Stimulation on another human using TENS device. This is effectively recording motor actions perform exact motor actions on the second human, effectively reducing the time to develop muscle memory.

8.2 Conclusion

This research showed that it is possible to build a Human to Human Interface system that allows to capture EEG signals generating from the Motor Actions and feed the Signals to second human that exact same motor actions can be performed. In order to achieve the objectives, Neural Impulses of Professional piano player was captured from median nerve and sent through signal processing system which convert the signal to the characteristics matching to the student on end of the system. This signals was then fed to TENS device to generate artificial neural impulses. These artificial neural impulses then fed to the student's median nerve using Non-Invasive Methods by placing two electrodes directly above the median nerve. To capture neural impulses from Median nerve, two electrodes were used and they were placed predefined locations on the human arm intercepting Median nerve. These electrodes captured neural impulses and then transferred them to Arduino board for interpretation. The modified Signal using Arduino then fed to TENS device which altered the signal's characteristics such as Frequency, Pulse Width, etc. to match to the characteristics of the target human's neuro-muscle system.

However it was noticed that due to non-Invasive nature of the EEG electrodes, signal captured were not error free. Therefore this system cannot be used to generate exact, error free, motor actions, but rather it provides guidance to the second human when they wanted to perform the exact same motor actions.

The aim of this project is to develop a Muscle Memory using human to human interface, however after analysing the results of the experiment, it was evident that this application has improved the learning process and reduce the time to develop muscle memory. Therefore the aim of the project is successfully achieved.

Following objectives mentioned early in the thesis are met successfully during the execution of the research.

- To critically study the domain of EEG & BCI field.
- To critically analyze the current approaches on duplicating Motor actions on another human.
- Recognizing the suitable technology to develop Human to Human Interface.
- Design & Develop Human to Human Interface, This has been discussed on the Chapter 02 & 03.
- Evaluate the Solution

In the first chapter problem was introduced with the surrounding issues on this research area and the novel solution. During the second and third chapters, related technologies such as Brain Computer Interfacing methods, pros and cons of the BCI, EEG, FES, and TENS were discussed. This review of the related technologies helped to select appropriate technology to design Human to Human Interface. The Third chapter explained in detail, the method used to analyze and design the Human to Human Interface.

In the Evaluation Chapter, Acceptable evaluation mechanism was devised to evaluate the system. Finally the solution was tested using three test subjects (2 men and 1 women), with an average age of 28 years, and results showed rapid increase in the effectiveness of the learning when the Human to Human interface is used. The Error rate while playing piano using Human-to-Human interface was 1.4167 where conventional method showed rate of 2.1666. In conclusion, Human to Human interface assisted learning, was 32.69% error free than the conventional method of learning, which supported the original hypothesis.

8.3 Further Work

In this project, we were able to design and develop human to human interface which assist on the developing a Muscle memory on another human. However due to non-invasive nature of the EEG signal capturing stage, lot of noise was introduced to the captured signal. This signal noises were due to blood flow, and small muscle contractions happening on human body. In order to remove this noise, we introduced a 10 Hz cap on the EEG signal frequency.

This Human to Human Interface can be further enhanced to capture EEG data more accurately, by using the Invasive methods. This needs medical doctor to operate human arm, and place electrodes inside the arm. This was EEG signals will be more accurate and noise free.

Apart from that, if the number of electrodes, employed on Source and Destination human can be increased, it will generate more accurate EEG signals effectively generating more accurate motor actions.

8.4 Summary

In this chapter we have discussed the conclusions that we can finally derived from the research, Problems encountered, limitations of the solution and further work was also discussed in here. The Experimental Result was thorough and results supported the hypothesis. Further work on the related areas were identified and documented.

Reference

[1] Brent J. Lance, Member IEEE, Scott E. Kerick, Anthony J. Ries, Kelvin S. Oie, and Kaleb McDowell, Senior Member IEEE "Brain–Computer Interface Technologies in the Coming Decades" 2012

[2] F. Quandt and F. C. Hummel, "The influence of functional electrical stimulation on hand motor recovery in stroke patients," 2014.

[3] C. Postelnicu, D. Talaba, and M. Toma, "Brain computer interfaces for medical applications," *Bull. Transilv. Univ. Brasov*, vol. 3, p. 52, 2010.

[4] G. E. Loeb, "NEUROPROSTHETIC INTERFACES–THE REALITY BEHIND BIONICS AND CYBORGS."

[5] G. E. Loeb, F. J. Richmond, and L. L. Baker, "The BION devices: injectable interfaces with peripheral nerves and muscles," *Neurosurg. Focus*, vol. 20, no. 5, pp. 1–9, 2006.

[6] D. J. Weber, R. B. Stein, K. M. Chan, G. E. Loeb, F. J. . Richmond, R. Rolf, K. James, S. L. Chong, A. K. Thompson, and J. Misiaszek, "Functional electrical stimulation using microstimulators to correct foot drop: a case study," *Can. J. Physiol. Pharmacol.*, vol. 82, no. 8–9, pp. 784–792, Jul. 2004.

[7] K. Warwick, M. Gasson, B. Hutt, I. Goodhew, P. Kyberd, H. Schulzrinne, and X. Wu, "Thought communication and control: a first step using radiotelegraphy," *IEE Proc. - Commun.*, vol. 151, no. 3, p. 185, 2004.

[8] "Transcutaneous Electrical Nerve Stimulation (TENS) - Physiopedia, universal access to physiotherapy knowledge." [Online]. Available: http://www.physio-pedia.com/Transcutaneous_Electrical_Nerve_Stimulation_(TENS). [Accessed: 18-Aug-2015].

[9] Virgílio Bento, Luís Paula, António Ferreira, Nuno Figueiredo, Ana Tomé, Filipe Silva, João Paulo Cunha and Pétia Georgieva "Advances in EEG-based Brain-Computer Interfaces for Control and Biometry", 2006.

[10] L. Brosseau, K. Yonge, V. Welch, S. Marchand, M. Judd, G. A. Wells, and P. Tugwell, "Transcutaneous electrical nerve stimulation (TENS) for the treatment of rheumatoid arthritis in the hand," in *Cochrane Database of Systematic Reviews*, The Cochrane Collaboration, Ed. Chichester, UK: John Wiley & Sons, Ltd, 2003.

[11] J.R. Wolpaw, G.E. Loeb, B.Z. Allison, E. Donchin, O.F. do Nascimento, W.J. Heetderks, F.Nijboer, W.G. Shain, and J.N. Turner, BCI Meeting 2005 – workshop on signals and recording methods, IEEE Trans Neural Syst Rehabil Eng: A Pub IEEE Eng Med Biol Soc. 14, Jun.,138–141, (2006).

[12] X. Gao, D. Xu, M. Cheng, and S. Gao, A BCI-based environmental controller for the motiondisabled. IEEE Trans Neural Syst Rehabil Eng, 11, Jun., 137–140, (2003).

[13] S.G. Mason, A. Bashashati, M. Fatourechi, K.F. Navarro, and G.E. Birch, A comprehensive survey of brain interface technology designs. Ann Biomed Eng, 35, Feb., 137–169, (2007)

[14] R. Leeb, D. Friedman, G.R. Müller-Putz, R. Scherer, M. Slater, and G.Pfurtscheller, SelfPaced (Asynchronous) BCI control of a wheelchair in virtual environments: A case study with a Tetraplegic. Comput Intell Neurosci, 79642,(2007)

[15] Anand Nayyar, Vikram Puri A review of Arduino board's, Lilypad's & Arduino shields, Computing for Sustainable Global Development (INDIACom), 2016

[16] Dr Suzy Duckworth, GPST1 Norwich; Shamsher Diu FES for foot drop in stroke and MS policy Norfolk County Council 2012

[17] Fung, J., & Barbeau, H. (1994). Effects of conditioning cutaneomuscular stimulation on the soleus H-reflex in normal and spastic paretic subjects during walking and standing. Journal of Neurophysiology, 72(5), 2090-2104.

[18] Morten L. Kringelbach, Ned Jenkinson, Sarah L.F. Owen & Tipu Z. Aziz (2007). Translational principles of deep brain stimulation. Nature Reviews Neuroscience 8, 623-635 (August 2007).

Appendix A

EEG Capturing using Arduino Shield

A.1 Introduction

Arduino Board comes with the Default code which detect the analogue signal from the Electrodes. This code allows captured data to be written to USB port.

A.2 Arduino Source Code

Following Arduino code has been slightly altered to make the signal more strong and accurate.

```
#define EKG A0
#define BUFFER_SIZE 100
#define SIZE_OF_COMMAND_BUFFER 30
#ifndef cbi
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif
#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif

int buffersize = BUFFER_SIZE;
int head = 0;
int tail = 0;
byte writeByte;
char commandBuffer[SIZE_OF_COMMAND_BUFFER];
byte reading[BUFFER_SIZE];
int interrupt_Number=1999;
int numberOfChannels = 1;
int tempSample = 0;
int commandMode = 0;

void setup(){
  Serial.begin(230400);
  delay(300);
  Serial.println("StartUp!");

  cli();

  sbi(ADCSRA,ADPS2);
  cbi(ADCSRA,ADPS1);
  cbi(ADCSRA,ADPS0);


  TCCR1A = 0;
  TCCR1B = 0;
```

```
 TCNT1  = 0;

 OCR1A = interrupt_Number;

 TCCR1B |= (1 << WGM12);

 TCCR1B |= (1 << CS11);

 TIMSK1 |= (1 << OCIE1A);

 sei();

}


ISR(TIMER1_COMPA_vect) {
  if(commandMode!=1)
  {
     tempSample = analogRead(A0);
     reading[head] =  (tempSample>>7)|0x80;
     head = head+1;
     if(head==BUFFER_SIZE)
     {
      head = 0;
     }
     reading[head] =  tempSample & 0x7F;
     head = head+1;
     if(head==BUFFER_SIZE)
     {
      head = 0;
     }
   }
}

void serialEvent()
{
 commandMode = 1;
 TIMSK1 &= ~(1 << OCIE1A);
 String inString = Serial.readStringUntil('\n');

 inString.toCharArray(commandBuffer, SIZE_OF_COMMAND_BUFFER);
 commandBuffer[inString.length()] = 0;

 char* command = strtok(commandBuffer, ";");
 while (command != 0)
 {
    char* separator = strchr(command, ':');
    if (separator != 0)
    {
       *separator = 0;
```

```
        --separator;
        if(*separator == 'c')
        {
          separator = separator+2;
          numberOfChannels = atoi(separator);
        }
         if(*separator == 's')
        {

        }
    }

    command = strtok(0, ";");
  }

  OCR1A = (interrupt_Number+1)*numberOfChannels - 1;
  TIMSK1 |= (1 << OCIE1A);
  commandMode = 0;
}


void loop(){

    while(head!=tail && commandMode!=1)
    {
      Serial.write(reading[tail]);
      tail = tail+1;
      if(tail==BUFFER_SIZE)
      {
        tail = 0;
      }
    }
}
```

Appendix B

EEG Analogue to Digital Conversion using Arduino Shield

A.1 Introduction

This Arduino code has been written to Convert EEG Analogue Signal to Digital Signal and output via the Arduino UNO Digital pins.

A.2 Arduino Source Code

```
#include <Servo.h>
Servo ServoGripper;

#define NUM_LED 6
#define MAX_Low 100
#define MAX_High 254
#define Threshold 3
#define threshold_degrees 10

int reading[10];
int finalReading;
int StimPin = 3;
int SwitchPin = 4;
int SwitchThreshold = 7;
int GripPin = 2;
int SwitchState = 0;
int SwitchThresholdState = 0;
int MAX = 0;
byte litLeds = 0;
byte multiplier = 1;
byte leds[] = {8, 9, 10, 11, 12, 13};
int aQ1 = 11;
int aQ2 = 13;
int aQ3 = 8;

const int UpdateTime = 200;
unsigned long OldTime = 0;
int old_degrees = 0;
int new_degrees = 0;

void setup(){
  Serial.begin(9600);
  ServoGripper.attach(GripPin);
  pinMode(StimPin, OUTPUT);
  pinMode(SwitchPin, INPUT);
  pinMode(SwitchThreshold, INPUT);
```

```
 for(int i = 0; i < NUM_LED; i++){
  pinMode(leds[i], OUTPUT);
 pinMode(aQ1, OUTPUT);

 }
 MAX = MAX_High;


}

void loop(){

  SwitchThresholdState = digitalRead(SwitchThreshold);
  if (SwitchThresholdState == HIGH){
  if (MAX == MAX_High){ MAX = MAX_Low; digitalWrite(aQ3, HIGH);}
  else{
   MAX = MAX_High; digitalWrite(aQ2, HIGH); }
  while (SwitchThresholdState == HIGH) {
  SwitchThresholdState = digitalRead(SwitchThreshold);
  delay(10); }
  }

  SwitchState = digitalRead(SwitchPin);
  if (SwitchState == HIGH){
  digitalWrite(StimPin, HIGH), digitalWrite(aQ1, HIGH);
   while (SwitchState == HIGH) {
  SwitchState = digitalRead(SwitchPin);
  delay(10); }
  }

 for(int i = 0; i < 10; i++){
  reading[i] = analogRead(A0) * multiplier;
  delay(2);
 }
 for(int i = 0; i < 10; i++){
  finalReading += reading[i];
 }
 finalReading /= 10;
 for(int j = 0; j < NUM_LED; j++){

  digitalWrite(leds[j], LOW);
  digitalWrite(StimPin, LOW);
 }
 Serial.println(finalReading);
 finalReading = constrain(finalReading, 0, MAX);
 litLeds = map(finalReading, 0, MAX, 0, NUM_LED);

 for(int k = 0; k < litLeds; k++){
  digitalWrite(leds[k], HIGH);
  if (k >= Threshold){
```

```
    digitalWrite(StimPin, HIGH);
   }
  }

  new_degrees = map(finalReading, 0 ,MAX, 165, 0);

if (millis() - OldTime > UpdateTime){
  if(abs(new_degrees-old_degrees) > threshold_degrees){
   ServoGripper.write(new_degrees);
  }
  OldTime = millis();
  old_degrees = new_degrees;
}
}
```

A.3 Arduino Source Code – EEG Signal Detection

```
#define EKG A0
#define BUFFER_SIZE 100
#define SIZE_OF_COMMAND_BUFFER 30
#ifndef cbi
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif
#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif

int buffersize = BUFFER_SIZE;
int head = 0;
int tail = 0;
byte writeByte;
char commandBuffer[SIZE_OF_COMMAND_BUFFER];
byte reading[BUFFER_SIZE];
int interrupt_Number=1999;
int numberOfChannels = 1;
int tempSample = 0;
int commandMode = 0;

void setup(){
  Serial.begin(9600);
  delay(100);
  Serial.println("StartUp!");

  cli();

  sbi(ADCSRA,ADPS2);
  cbi(ADCSRA,ADPS1);
  cbi(ADCSRA,ADPS0);
```

```
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1  = 0;

  OCR1A = interrupt_Number;

  TCCR1B |= (1 << WGM12);

  TCCR1B |= (1 << CS11);

  TIMSK1 |= (1 << OCIE1A);

  sei();

}


ISR(TIMER1_COMPA_vect) {
   if(commandMode!=1)
   {
      tempSample = analogRead(A0);
      reading[head] =  (tempSample>>7)|0x80;
      head = head+1;
      if(head==BUFFER_SIZE)
      {
       head = 0;
      }
      reading[head] =  tempSample & 0x7F;
      head = head+1;
      if(head==BUFFER_SIZE)
      {
       head = 0;
      }
    }
}

void serialEvent()
{
  commandMode = 1;
  TIMSK1 &= ~(1 << OCIE1A);
  String inString = Serial.readStringUntil('\n');

  inString.toCharArray(commandBuffer, SIZE_OF_COMMAND_BUFFER);
  commandBuffer[inString.length()] = 0;

  char* command = strtok(commandBuffer, ";");
  while (command != 0)
  {
     char* separator = strchr(command, ':');
     if (separator != 0)
```

```
      {
          *separator = 0;
          --separator;
          if(*separator == 'c')
          {
            separator = separator+2;
            numberOfChannels = atoi(separator);
          }
           if(*separator == 's')
          {

          }
      }

      command = strtok(0, ";");
  }

  OCR1A = (interrupt_Number+1)*numberOfChannels - 1;
  TIMSK1 |= (1 << OCIE1A);
  commandMode = 0;
}


void loop(){

    while(head!=tail && commandMode!=1)
    {
      Serial.write(reading[tail]);
      tail = tail+1;
      if(tail==BUFFER_SIZE)
      {
        tail = 0;
      }
    }
}
```

A.3 C#.NET Code – EEG Signal Visualization

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading.Tasks;
using System.IO.Ports;
using System.Windows.Forms.DataVisualization.Charting;

namespace EEGC
{
    public partial class Form1 : Form
    {
        private SerialPort sensport;
        private string data;

        public Form1()
        {
            InitializeComponent();
            Control.CheckForIllegalCrossThreadCalls = false;


        }

        private void Form1_Load(object sender, EventArgs e)
        {
            comboBox1.DataSource = SerialPort.GetPortNames();
            timer1.Start();


        }
```

```
double rt = 0;
Boolean i = false;
private void timer1_Tick(object sender, EventArgs e)
{
    rt = rt + 0.1;
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    sensport.Close();
}

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        if (comboBox2.Text == "")
        {
            MessageBox.Show("Select a Baud Rate");
        }

        else
        {
            sensport = new SerialPort();
            sensport.BaudRate = int.Parse(comboBox2.Text);
            sensport.PortName = comboBox1.Text;
            sensport.Parity = Parity.None;
            sensport.DataBits = 8;
            sensport.StopBits = StopBits.One;
            sensport.Handshake = Handshake.None;
            sensport.DataReceived += sensport_DataReceived;

            try
```

```
            {
                sensport.Open();
                textBox1.Text = "";
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error");
            }
        }
    }
    catch(Exception E)
    {
        MessageBox.Show(E.Message, "Error");
    }

}

void sensport_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    try
    {
        if (i == false)
        {
            rt = 0;
            i = true;
        }

        data = sensport.ReadLine();
        double datas;
        if ((double.TryParse(data, out datas)) && datas < 1200)
        {
            this.chart1.Series["Data1"].Points.AddXY(rt, datas);
            textBox2.Text = textBox2.Text + data + Environment.NewLine;
        }
```

```
        }
        catch(Exception ex)
        {
            MessageBox.Show(ex.Message, "Error");
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        string filename = textBox1.Text;

        if (filename == "")
        {
            MessageBox.Show("Please enter file location");
        }

        else {
            try
            {

                string fileurl = @"" + filename + "";
                this.chart1.SaveImage(("+harsha+"), ChartImageFormat.Png);

                MessageBox.Show("File Saved " + fileurl);
            }
            catch (Exception ex3)
            {
                MessageBox.Show(ex3.Message, "Error");
            }

        }

    }
```

```
        private void label4_Click(object sender, EventArgs e)
        {

        }

        private void chart1_Click(object sender, EventArgs e)
        {

        }
    }
}
```

Designer.cs

```
namespace EEGC
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
```

```
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.components = new System.ComponentModel.Container();
        System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea2 =
new System.Windows.Forms.DataVisualization.Charting.ChartArea();
        System.Windows.Forms.DataVisualization.Charting.Legend legend2 = new
System.Windows.Forms.DataVisualization.Charting.Legend();
        System.Windows.Forms.DataVisualization.Charting.Series series3 = new
System.Windows.Forms.DataVisualization.Charting.Series();
        System.Windows.Forms.DataVisualization.Charting.Series series4 = new
System.Windows.Forms.DataVisualization.Charting.Series();
        this.comboBox1 = new System.Windows.Forms.ComboBox();
        this.timer1 = new System.Windows.Forms.Timer(this.components);
        this.chart1 = new
System.Windows.Forms.DataVisualization.Charting.Chart();
        this.button1 = new System.Windows.Forms.Button();
        this.button2 = new System.Windows.Forms.Button();
        this.label5 = new System.Windows.Forms.Label();
        this.label6 = new System.Windows.Forms.Label();
        this.comboBox2 = new System.Windows.Forms.ComboBox();
        this.textBox1 = new System.Windows.Forms.TextBox();
        this.label1 = new System.Windows.Forms.Label();
        this.textBox2 = new System.Windows.Forms.TextBox();
        ((System.ComponentModel.ISupportInitialize)(this.chart1)).BeginInit();
        this.SuspendLayout();
        //
```

```
        // comboBox1
        //
        this.comboBox1.FormattingEnabled = true;
        this.comboBox1.Location = new System.Drawing.Point(15, 25);
        this.comboBox1.Name = "comboBox1";
        this.comboBox1.Size = new System.Drawing.Size(121, 21);
        this.comboBox1.TabIndex = 0;
        //
        // chart1
        //
        this.chart1.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.T
op | System.Windows.Forms.AnchorStyles.Bottom)
        | System.Windows.Forms.AnchorStyles.Left)
        | System.Windows.Forms.AnchorStyles.Right)));
        this.chart1.BackColor = System.Drawing.SystemColors.Window;
        chartArea2.Name = "ChartArea1";
        this.chart1.ChartAreas.Add(chartArea2);
        legend2.Name = "Legend1";
        this.chart1.Legends.Add(legend2);
        this.chart1.Location = new System.Drawing.Point(192, 12);
        this.chart1.Name = "chart1";
        this.chart1.Palette =
System.Windows.Forms.DataVisualization.Charting.ChartColorPalette.Fire;
        series3.BorderColor = System.Drawing.Color.Yellow;
        series3.BorderWidth = 2;
        series3.ChartArea = "ChartArea1";
        series3.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.FastLine;
        series3.Color = System.Drawing.Color.Indigo;
        series3.Legend = "Legend1";
        series3.Name = "Data1";
        series3.Palette =
System.Windows.Forms.DataVisualization.Charting.ChartColorPalette.Fire;
```

```
        series4.BorderWidth = 2;
        series4.ChartArea = "ChartArea1";
        series4.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.FastLine;
        series4.Color = System.Drawing.Color.Red;
        series4.Legend = "Legend1";
        series4.Name = "Data2";
        this.chart1.Series.Add(series3);
        this.chart1.Series.Add(series4);
        this.chart1.Size = new System.Drawing.Size(1125, 628);
        this.chart1.TabIndex = 1;
        this.chart1.TabStop = false;
        this.chart1.Text = "chart1";
        this.chart1.Click += new System.EventHandler(this.chart1_Click);
        //
        // button1
        //
        this.button1.Location = new System.Drawing.Point(15, 97);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(121, 23);
        this.button1.TabIndex = 3;
        this.button1.Text = "OK";
        this.button1.UseVisualStyleBackColor = true;
        this.button1.Click += new System.EventHandler(this.button1_Click);
        //
        // button2
        //
        this.button2.Location = new System.Drawing.Point(15, 615);
        this.button2.Name = "button2";
        this.button2.Size = new System.Drawing.Size(110, 23);
        this.button2.TabIndex = 4;
        this.button2.Text = "Save";
        this.button2.UseVisualStyleBackColor = true;
        this.button2.Click += new System.EventHandler(this.button2_Click);
```

```
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(12, 9);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(53, 13);
this.label5.TabIndex = 10;
this.label5.Text = "COM Port";
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(15, 53);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(58, 13);
this.label6.TabIndex = 11;
this.label6.Text = "Baud Rate";
//
// comboBox2
//
this.comboBox2.FormattingEnabled = true;
this.comboBox2.Items.AddRange(new object[] {
"4800",
"9600",
"14400",
"19200",
"28800",
"38400",
"57600",
"115200"});
this.comboBox2.Location = new System.Drawing.Point(15, 70);
this.comboBox2.Name = "comboBox2";
this.comboBox2.Size = new System.Drawing.Size(121, 21);
```

```
this.comboBox2.TabIndex = 12;
//
// textBox1
//
this.textBox1.Location = new System.Drawing.Point(15, 589);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(147, 20);
this.textBox1.TabIndex = 5;
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(15, 573);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(67, 13);
this.label1.TabIndex = 6;
this.label1.Text = "File Location";
//
// textBox2
//
this.textBox2.Location = new System.Drawing.Point(18, 126);
this.textBox2.Multiline = true;
this.textBox2.Name = "textBox2";
this.textBox2.ScrollBars = System.Windows.Forms.ScrollBars.Both;
this.textBox2.Size = new System.Drawing.Size(100, 320);
this.textBox2.TabIndex = 13;
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.SystemColors.Window;
this.ClientSize = new System.Drawing.Size(1334, 652);
this.Controls.Add(this.textBox2);
```

```
            this.Controls.Add(this.comboBox2);
            this.Controls.Add(this.label6);
            this.Controls.Add(this.label5);
            this.Controls.Add(this.label1);
            this.Controls.Add(this.textBox1);
            this.Controls.Add(this.button2);
            this.Controls.Add(this.button1);
            this.Controls.Add(this.chart1);
            this.Controls.Add(this.comboBox1);
            this.Name = "Form1";
            this.Text = "EEG Capturing System";
            this.Load += new System.EventHandler(this.Form1_Load);
            ((System.ComponentModel.ISupportInitialize)(this.chart1)).EndInit();
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.ComboBox comboBox1;
        private System.Windows.Forms.Timer timer1;
        private System.Windows.Forms.DataVisualization.Charting.Chart chart1;
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.Button button2;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.ComboBox comboBox2;
        private System.Windows.Forms.TextBox textBox1;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TextBox textBox2;
    }
}
```

csproj

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build"
xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <Import
Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.p
rops"
Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.
Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{BBDBCA9D-0CA8-49F7-86C1-1643DD829291}</ProjectGuid>
    <OutputType>WinExe</OutputType>
    <AppDesignerFolder>Properties</AppDesignerFolder>
    <RootNamespace>EEGC</RootNamespace>
    <AssemblyName>EEGC</AssemblyName>
    <TargetFrameworkVersion>v4.5.2</TargetFrameworkVersion>
    <FileAlignment>512</FileAlignment>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>false</Optimize>
    <OutputPath>bin\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
  </PropertyGroup>
```

```
 <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
   <PlatformTarget>AnyCPU</PlatformTarget>
   <DebugType>pdbonly</DebugType>
   <Optimize>true</Optimize>
   <OutputPath>bin\Release\</OutputPath>
   <DefineConstants>TRACE</DefineConstants>
   <ErrorReport>prompt</ErrorReport>
   <WarningLevel>4</WarningLevel>
 </PropertyGroup>
 <ItemGroup>
  <Reference Include="System" />
  <Reference Include="System.Core" />
  <Reference Include="System.Windows.Forms.DataVisualization" />
  <Reference Include="System.Xml.Linq" />
  <Reference Include="System.Data.DataSetExtensions" />
  <Reference Include="Microsoft.CSharp" />
  <Reference Include="System.Data" />
  <Reference Include="System.Deployment" />
  <Reference Include="System.Drawing" />
  <Reference Include="System.Net.Http" />
  <Reference Include="System.Windows.Forms" />
  <Reference Include="System.Xml" />
</ItemGroup>
<ItemGroup>
 <Compile Include="Form1.cs">
   <SubType>Form</SubType>
 </Compile>
 <Compile Include="Form1.Designer.cs">
   <DependentUpon>Form1.cs</DependentUpon>
 </Compile>
 <Compile Include="Program.cs" />
 <Compile Include="Properties\AssemblyInfo.cs" />
 <EmbeddedResource Include="Form1.resx">
```

```
      <DependentUpon>Form1.cs</DependentUpon>
    </EmbeddedResource>
    <EmbeddedResource Include="Properties\Resources.resx">
      <Generator>ResXFileCodeGenerator</Generator>
      <LastGenOutput>Resources.Designer.cs</LastGenOutput>
      <SubType>Designer</SubType>
    </EmbeddedResource>
    <Compile Include="Properties\Resources.Designer.cs">
      <AutoGen>True</AutoGen>
      <DependentUpon>Resources.resx</DependentUpon>
    </Compile>
    <None Include="Properties\Settings.settings">
      <Generator>SettingsSingleFileGenerator</Generator>
      <LastGenOutput>Settings.Designer.cs</LastGenOutput>
    </None>
    <Compile Include="Properties\Settings.Designer.cs">
      <AutoGen>True</AutoGen>
      <DependentUpon>Settings.settings</DependentUpon>
      <DesignTimeSharedInput>True</DesignTimeSharedInput>
    </Compile>
  </ItemGroup>
  <ItemGroup>
    <None Include="App.config" />
  </ItemGroup>
  <Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
  <!-- To modify your build process, add your task inside one of the targets below and
uncomment it.
       Other similar extension points exist, see Microsoft.Common.targets.
  <Target Name="BeforeBuild">
  </Target>
  <Target Name="AfterBuild">
  </Target>
  -->
</Project>
```