UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEVELOPMENT OF MICROSLEEP DETECTION SYSTEM BY USING BIOSENSOR MODULE AND PULSE RATE

This report submitted in accordance with the requirement of the Universiti Teknikal Malaysia Melaka (UTeM) for the Bachelor's Degree in Electronics Engineering Technology (Telecommunications) (Hons).

by

NUR MIMI WAHEDA BINTI MD YUSOFF

B071310692

FACULTY OF ENGINEERING TECHNOLOGY

2016

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS LAPORAN PROJEK SARJANA MUDA

TAJUK: **DEVELOPMENT OF MICROSLEEP DETECTION SYSTEM BY USING BIOSENSOR MODULE AND PULSE RATE**

SESI PENGAJIAN: **2016/17 Semester 1**

Saya **NUR MIMI WAHEDA BINTI MD YUSOFF**

mengaku membenarkan Laporan PSM ini disimpan di Perpustakaan Universiti Teknikal Malaysia Melaka (UTeM) dengan syarat-syarat kegunaan seperti berikut:

1. Laporan PSM adalah hak milik Universiti Teknikal Malaysia Melaka dan penulis.
2. Perpustakaan Universiti Teknikal Malaysia Melaka dibenarkan membuat salinan untuk tujuan pengajian sahaja dengan izin penulis.
3. Perpustakaan dibenarkan membuat salinan laporan PSM ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (✓)

☐	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia sebagaimana yang termaktub dalam AKTA RAHSIA RASMI 1972)
☐	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
☐	TIDAK TERHAD	

Disahkan oleh:

__________________________ __________________________

Alamat Tetap:

Cop Rasmi:

Tarikh: __________________________

Tarikh: __________________________

** Jika Laporan PSM ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh laporan PSM ini perlu dikelaskan sebagai SULIT atau TERHAD.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FAKULTI TEKNOLOGI KEJURUTERAAN
Tel : +606 234 6623 | Faks : +606 23406526

Rujukan Kami (Our Ref) :
Rujukan Tuan (Your Ref) :

28 JAN XXXX - Cth

Pustakawan
Perpustakaan UTeM
Universiti Teknikal Malaysia Melaka
Hang Tuah Jaya,
76100 Durian Tunggal,
Melaka.

Tuan/Puan,

PENGKELASAN LAPORAN PSM SEBAGAI SULIT/TERHAD LAPORAN PROJEK SARJANA MUDA TEKNOLOGI KEJURUTERAAN PEMBUATAN (COURSE NAME): AAA BIN BBB

Sukacita dimaklumkan bahawa Laporan PSM yang tersebut di atas bertajuk ***"Quality and Performance Investigation of TiAlN Coated Drill Tools"*** mohon dikelaskan sebagai *SULIT / TERHAD untuk tempoh LIMA (5) tahun dari tarikh surat ini.

2. Hal ini adalah kerana IANYA MERUPAKAN PROJEK YANG DITAJA OLEH SYARIKAT LUAR DAN HASIL KAJIANNYA ADALAH SULIT.

Sekian dimaklumkan. Terima kasih.

Yang benar,

____
Tandatangan dan Cop Penyelia

* Potong yang tidak berkenaan

NOTA: BORANG INI HANYA DIISI JIKA DIKLASIFIKASIKAN SEBAGAI SULIT DAN TERHAD. JIKA LAPORAN DIKELASKAN SEBAGAI **TIDAK TERHAD**, MAKA BORANG INI **TIDAK PERLU DISERTAKAN** DALAM LAPORAN PSM.

DECLARATION

I hereby, declared this report entitled “The Development of Microsleep Detection System by Using Biosensor and Pulse Rate” is the results of my own research except as cited in references.

Signature	:	………………………………………….
Author’s Name	:	NUR MIMI WAHEDA BINTI MD YUSOFF
Date	:	…………………………………………

APPROVAL

This report is submitted to the Faculty of Engineering Technology of UTeM as a partial fulfillment of the requirements for the degree of Bachelor of in Electrical Engineering Technology in Telecommunications (Hons). The member of the supervisory is as follow:

………………………………

(AHMAD FAUZAN BIN KADMIN)

ABSTRAK

Pada masa kini, bilangan kemalangan meningkat dari hari ke hari. Banyak kemalangan berlaku kepada mana-mana generasi rakyat kerana banyak faktor. Salah satu yang paling sering faktor adalah pemandu dia berasa mengantuk apabila memandu atau yang mempunyai tingkah laku yang microsleep. Apabila mereka mempunyai microsleep tidak ramai yang dapat membantu kemudian apabila di tempat duduk penumpang kosong. Projek ini dibina untuk membantu mereka yang mempunyai microsleep atau perasaan apabila memandu mengantuk atau membuat kerja-kerja yang memerlukan tumpuan. Skop projek ini adalah bukan sahaja untuk pemandu tetapi kepada semua orang yang melakukan kerja-kerja yang memerlukan tumpuan penuh seperti guru, pelajar, dan juga atlet. Tujuan projek ini adalah untuk Microsleep Sistem Pengesanan dengan menggunakan Biosensor Modul dan kadar Pulse dengan menggunakan arduino sebagai peralatan yang penting untuk mengawal kadar nadi yang akan dibaca dengan menggunakan sensor ECG. Kod sumber arduino akan digunakan C atau C ++ Bahasa dalam persekitaran pembangunan bersepadu (IDE). The Matlab sotware juga digunakan dalam projek ini untuk melakukan klasifikasi kadar nadi dengan menggunakan Fungsi Radial Asas (RBF) Rangkaian neural. Kesimpulannya, produk ini bukan sahaja untuk pemandu tetapi juga untuk orang yang bekerja atau pelajar kerana ia boleh memberi isyarat kepada rakyat apabila mereka mempunyai microsleep itu.

ABSTRACT

Nowadays, the number of accident increased from day to day. Many accident happen to any generation of people because of many factors. One of the most often factor is he driver feels sleepy when drive or having the microsleep behaviour. When they are having microsleep not many people are be able to help then when on the passenger seat are empty. This project is built to help those who have microsleep or feeling sleepy when drive or do some work that need concentration. The scope of this project are not only for the driver but to all people that do the work who need a full concentration such as teacher, student, and also athletes. The aim of this project is to Microsleep Detection System by using Biosensor Module and Pulse rate by using the arduino as the important equipment to control the pulse rate that will be read by using the ECG Sensor. The source code of the arduino will be use C or C++ Language in the integrated development environment (IDE). The Matlab sotware is also being used in this project to do the classification of the pulse rate by using the Radial Basis Function (RBF) neural network. As conclusion, this product are not only for the driver but also for the working people or even student because it can give the signal for the people when they are having the microsleep.

DEDICATION

This humble effort specially dedicated to my beloved parents, family, friends and lecturers whose love can never be forgotten for their support, guidance and encouragement upon completing this project and report.

Special dedicated to my family

MD YUSOFF BIN ABDULLAH

RUZINA BINTI DERAMAN

ACKNOWLEDGEMENT

In the name of Allah SWT, the most Almighty and the most Merciful, I am thankful to Allah for enabling me to complete this report.

First and foremost, I would like to thanks to Ahmad Fauzan bin Kadmin as my supervisor, lecturer of Faculty Technology Engineering for giving me a full cooperation and supervision during making of this final report and project. Not to be forgotten, I would like to thanks my family for always supporting me in whatever I do. Besides, thanks to my friends for always encourage me by giving moral support and contributing their ideas in completing this report.

Last but not least, I wish to say deeply gratitude to party directly involved and indirectly involved to complete my project. All good offices you very I valued.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS, SYMBOLS AND NOMENCLATURE

RBF	-	Radial Basis Functions
MS	-	Microsleep
BMs	-	Behavioural Microsleeps
IDE	-	Integrated Development Environment
BPM	-	Beats per Minutes
fMRI	-	Functional Magnetic Resonance Image
EEG	-	Electroencephalography
PSG	-	Polysomnography
EOG	-	Electrooculogram
FD	-	Fractal Dimension
LSTM	-	Long Short-Term Recollection
fROI	-	Face Region of Interest
ESN	-	Echo State Network

CHAPTER 1
INTRODUCTION

1.0 Introduction

In this chapter, it covers the background of the microsleep, problem statements, objectives, scope, limitation, project's methodology and project expectation. This chapter presents all of the general knowledge the project and overall knowledge about the microsleep. The problem statements that stated on this chapter is according to the current issue and the facts from the data of accident which occur because of the microsleep. The general method that will be used is also stated in this chapter.

1.1 Background

The development of microsleep detection system by using biosensor module and pulse rate is a system where the ECG sensor takes the role to control the detection of the microsleep. Nowadays, many people having a problem to concentrate to do things that need the great concentration without getting sleepy or microsleep. Sometimes they need other people to help them to wake up them in the middle of having microsleep or help them to cheer them up to avoid having microsleep.

This project will train in a new way to see the performance of microsleep without getting helps from the others. The development of microsleep detection system by using biosensor module and pulse rate consists of ECG sensor to read the pulse rate of the person that used it.the pulse rate that the person who are having microsleep is in condition of slow heartbeat with rates around 40–50 BPM. However, the overall microsleep detection system operation is using arduino which can control the whole process by using suitable coding.

1.2 Problem Statement

Nowadays, from the police and forensic report, most the accidents happened because of microsleep behavior from the driver. The accidents that occur are not cause by the older driver, but also teenagers because the microsleep states happen to everyone who is doing the activity that need concentration. As we know, microsleep (MS) is a fleeting scene of rest which may keep going for a transitory of a moment or up to 30 seconds. In this type of condition, a person could not react to some subjective tactile info and gets to be distinctly oblivious respond to some arbitrary sensory input and becomes unconscious. By developing the microsleep detection system by using biosensor module and pulse rate, the accidents can be prevented as the pulse rate will detect the changing of pulse behavior when microsleep. So, in the end when the pulse rate is detected, it will trigger the driver and realize that they are having microsleep condition. However, the application of this project is focus for vehicle driver and students.

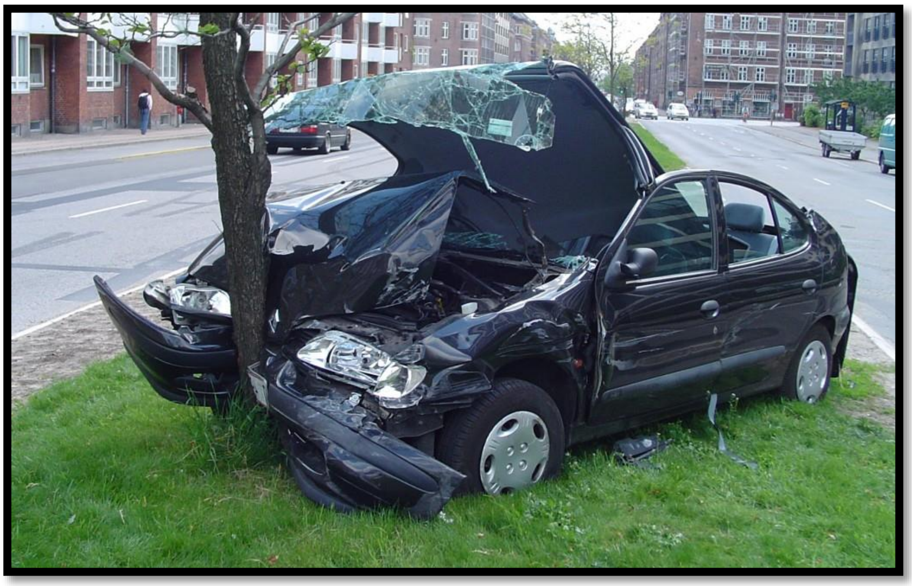


Figure 1.2 : Example of accidents happen

1.3 Objectives

The project is implemented in order to achieve the following objectives which are:

i. To develop the microsleep detection by using biosensor module and pulse rate.

ii. To analyse the performance of microsleep detection accuracy by using biosensor module and pulse rate.

1.4 Scope of Project

The scope of the project is to study and develop microsleep detection system by using biosensor module and pulse rate. Besides, the scopes of the project are to study about microsleep, the types of systems that had been used and understand and develop the biosensor module and pulse rate. Moreover, the criteria of the project scopes are including man, material and methods.

1.4.1 Man

Males and females drivers (lectures, students, workers, athletes and etc) with no certain of ages and personality.

1.4.2 Material

There are five main components that will be used in this projects which are ECG sensor, arduino and pulse rate**.**

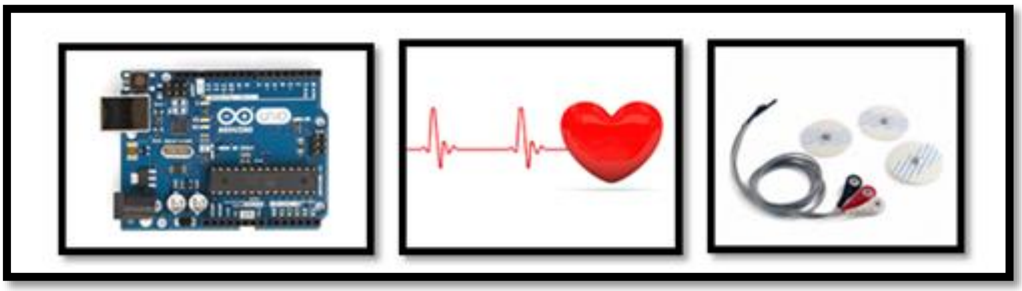


Figure 1.4a : The materials in this projects from left arduino, pulse rate and ECG Sensor

1.4.3 Methods

The methods that will be used are consists of one important method which is by using the Matlab of Radial Basis Function (RBF) neural network to train the pulse rate and to detect the accuracy of the pulse rate.

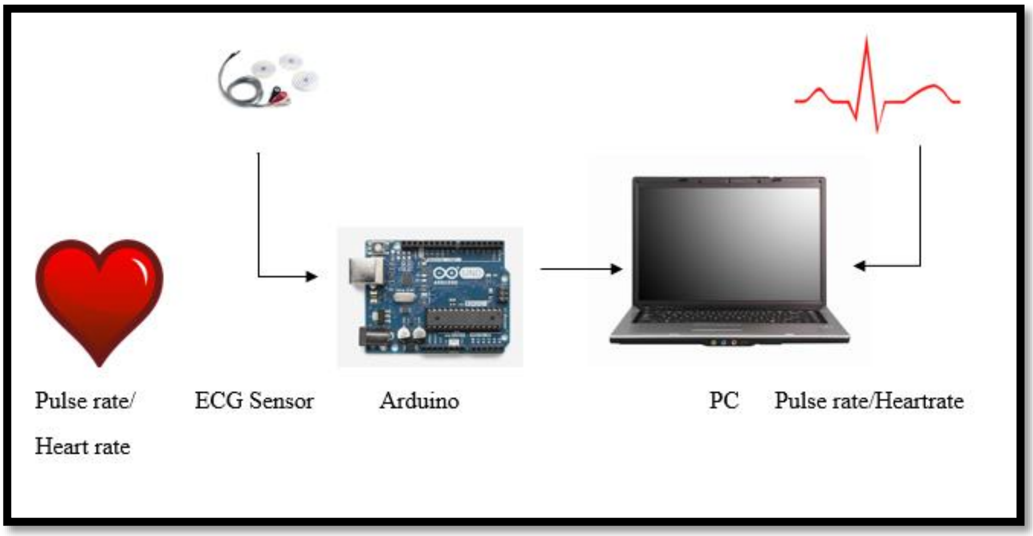


Figure 1.4b : The overview of project diagram

1.5 Project Methodology

In order to produce a good project, there are several procedures that must be followed. Initially, information about the microsleep problem must be identified. Then, more information about the developments of the microsleep problem are gathered from the journals, internet, books and also articles. Besides that, the research continues with the search on the type of the basic application of the arduino that will be used in this project and also search on coding on the arduino to be programmed by using integrated development environment (IDE). Next, after finishing the research, the coding will be simulated in arduino in order to identify whether the coding can be simulated without any error. The pulse rate then will display at the pulse rate amped visualizer to show the heart beat according to the pulse rate read. After that, the hardware for microsleep detection system will be designed. Lastly, the hardware will be combined with the coding to get the complete microsleep detection system by using biosensor module and pulse rate. The microsleep detection system is also develop by using the buzzer to alarm people if the microsleep condition is detected.

1.6 Thesis Structure

Chapter 1:

The first chapter introduces the summary of the idea of the project. It focused on the overview of the project flow, satisfying the objectives, the problem statement, scope and result of the project.

Chapter 2:

In this chapter, the project background is discussed. The methodology of the concept, theory, and several characteristic of components of hardware that are used in this project are also discussed in this chapter. This chapter also defines the terms that is being used in this project and discussed the concept of the research and how it is related with the theory.

Chapter 3:

Chapter 3 describes the detail of the methodology used in this project. The time table or steps that need to be completed and the detailed reports regarding the studies about the project to achieve the objectives of the project are presented.

Chapter 4:

Chapter four shows the result and discussion. All the simulations, data of the projects and results analysis obtained will be discussed in detail on every part of the process. The results will be compared with the aims outlined in order to satisfy for some hypothesis and conclusion.

Chapter 5:

Chapter five discuss about the conclusion and future work that can be undertaken in the future as the seed to design new project. Some recommendations and suggestion on how to surpass the performance of the system based on the desired results will be given.

CHAPTER 2
LITERATURE REVIEW

2.0 Introduction

This chapter introduces the summary of literature review on the development of the microsleep detection system by using biosensor module and pulse rate. The development of the different control system of the microsleep that aims to help people stay alert while doing their work is discussed. The advantages and disadvantages of each microsleep development are also presented and compared. This chapter also presents the development of the control system as main system that will be used will be utilized in this project.

2.1 History of Microsleep

A microsleep (MS) is a fleeting scene of rest which may keep going for a transitory of a moment or until 30 seconds. In this state, the person would not be able to react to some subjective tactile info and gets to be distinctly oblivious respond to discretionary tactile inputs and becomes unconscious. Microsleep is transpire when a person unable to find his cognizance and eventually receives cognizance after a moment in cognizance, or when there are sudden switches between conditions of alertness and sleep. In comportment terms, microsleep is striking as sagging visual perceivers, moderate visual cover conclusion, and head gesturing.

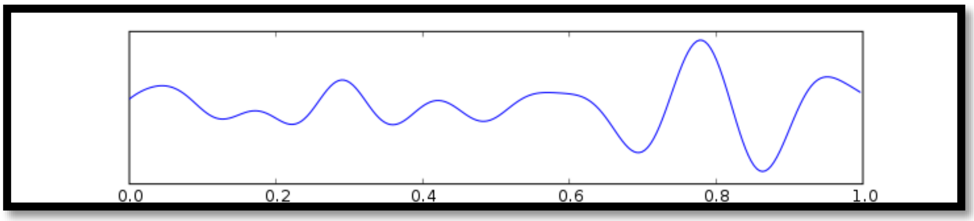


Figure 2.1a : Example of an EEG theta wave

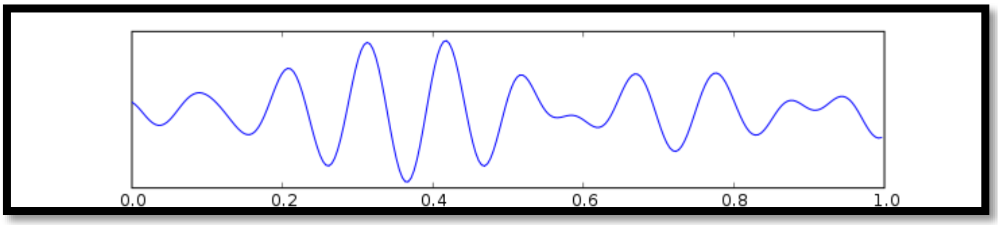


Figure 2.1b : Example of an EEG alpha wave

There are many confused and costly approaches to identify microsleep incorporate EEG, fMRI, EOG, and PSG associate with many sorts of programming stages. At the point when various tests are utilized as a part of relating, recognition of microsleep are likely will turn out to be more exact.

Method	**Description or examples**
PSG	• PSG detects many body functions • Ex: brain (EEG), eye movements (EOG), muscle activity or skeletal muscle activation (EMG) and heart rhythm (ECG) during sleep.
EEG	• EEG states the brain spontaneous activity over a short period of time for 20–40 minutes • Microsleeps make EEG shifts to lower frequencies from alpha to theta waves.

fMRI	• A functional neuroimaging procedure by using MRI technology • Measures brain activity by detecting associated changes in blood flow
Psychological tests	• React to test time • Ex : Karolinska Sleepiness Scale (KSS), Maintenance of Wakefulness Test (MWT)
EOG	EOG is a methods for measuring the resting potential of the retina in the human eye.
Eye-video test	Measures eyes flickering and eye developments movements to identify microsleep events.
Mouth yawning test	Counts number of yawns over a period of time.

Table 2.1 : The types of methods to detects microsleep and its example

2.2 Previous Research of Microsleep Detection System

M. T. R. Peiris, R. D. Jones, P. R. Davidson, P. J. Bones, D. J. Myall (2005) have been studied on the fractal measurement (FD) of EEG that has been shown to be of significant worth in the discovery of epileptic seizures. The exploration has been recorded for one hour of a consistent following assignment while EEG, EOG and facial video are recorded. Higuchi's calculation was habituated to ascertain the FD of the EEG. A subset of information is evaluated autonomously by three human raters watching both following execution and the video rating to recognize behavioral microsleep occasions. Video failures were scored autonomously from following execution by a human rater. The normal point biserial relationship amongst's FD and the mean human rating was - 0.213 signifying unassuming accidence. [1].

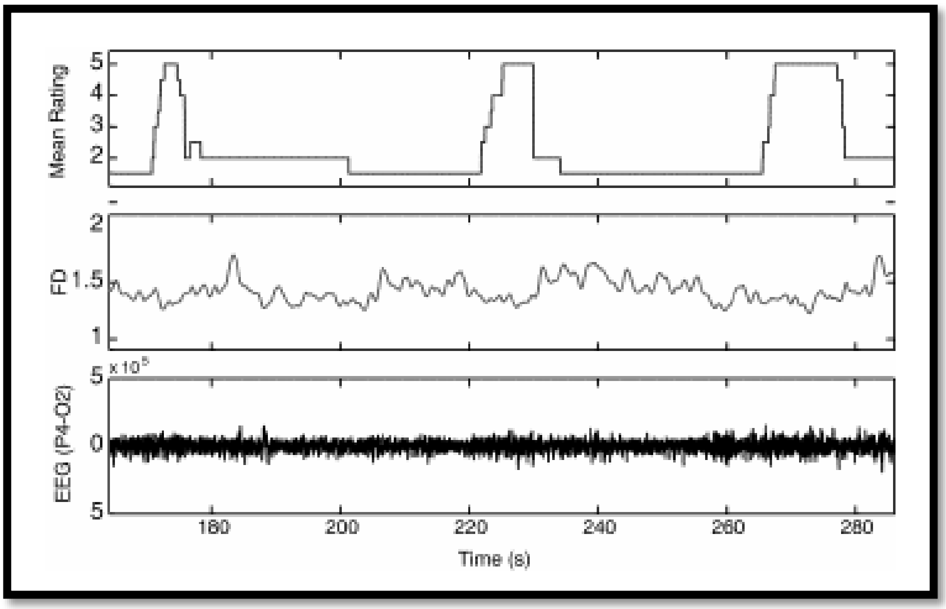


Figure 2.2a : A 2 min sample segment of the mean rating, Higuchi FD, and EEG

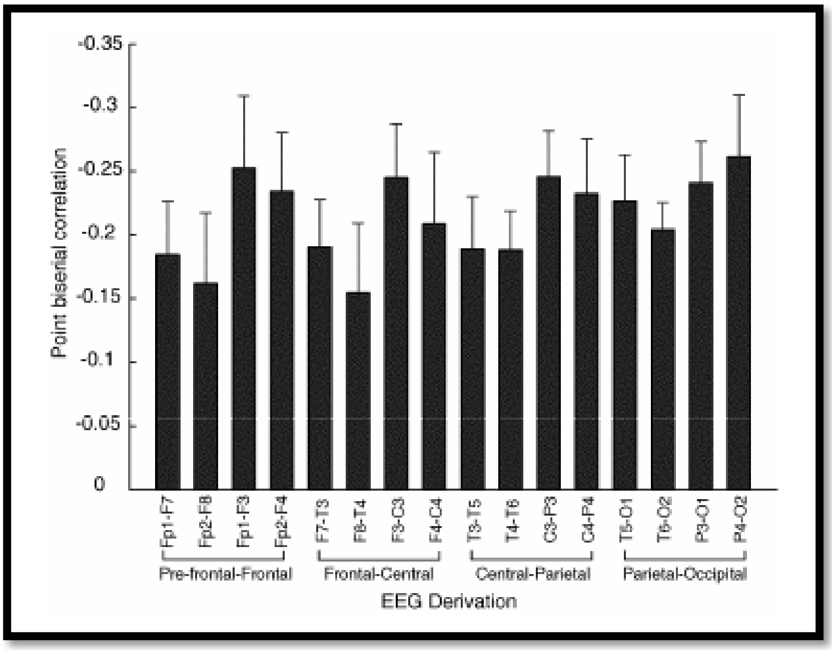


Figure 2.2b : Mean point-biserial correlation between the continuous FD and the binary lapse index over all EEG derivations

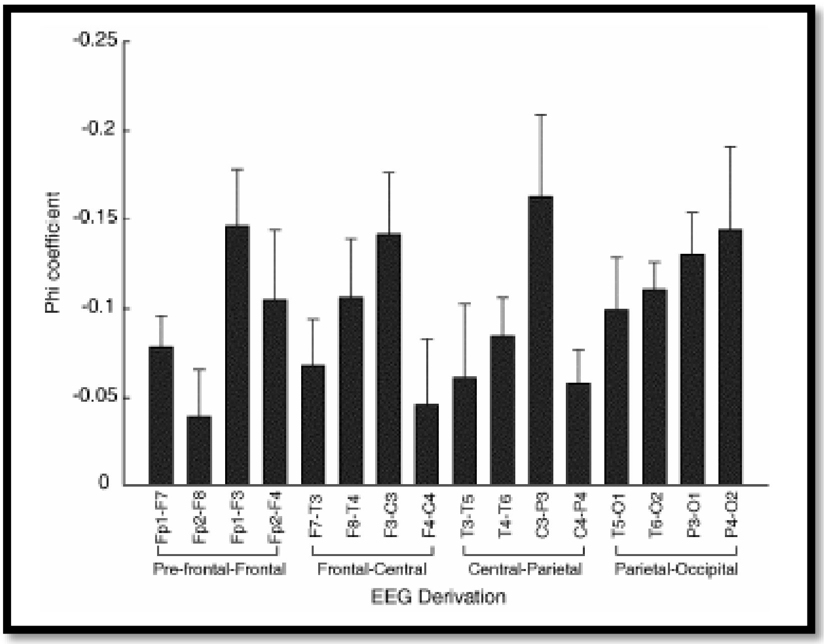


Figure 2.2c : Mean Φ coefficients between the FD and the Lapse Index over all EEG derivations

P. R. Davidson, R. D. Jones and M. T. R. Peiris (2005) did the research that focused on Lapses in visuomotor performance are customarily connected with behavioral microsleep events. Encountering a pass of this sort while while performing a paramount task can have catastrophic consequences. A caveat system capable of reliably detecting patterns in EEG occurring afore or during a lapse has the potential to preserve many lives. To prepare and approve the framework the system, EEG, facial video and tracking data were amassed from 15 subjects performing a visuomotor tracking task for 2 1-hour sessions. They are developing a behavioral microsleep detection system which employs Long Short-Term Recollection (LSTM) recurrent neural networks. This provided behavioural information on lapse events with good temporal resolution. They developed an automated demeanor rating system and trained it to estimate the mean opinion of 3 human raters on the likelihood of a lapse. They then trained an LSTM neural network to evaluate the yield of the lapse rating framework given only EEG spectral data. The detection system was designed to operate in authentic-time without calibration for individual subjects [2].

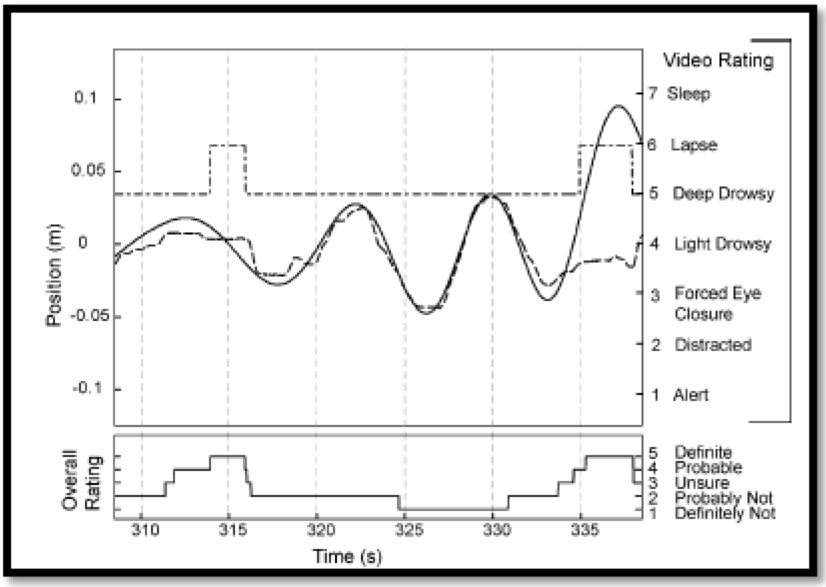


Figure 2.2d : Interface used for lapse rating study

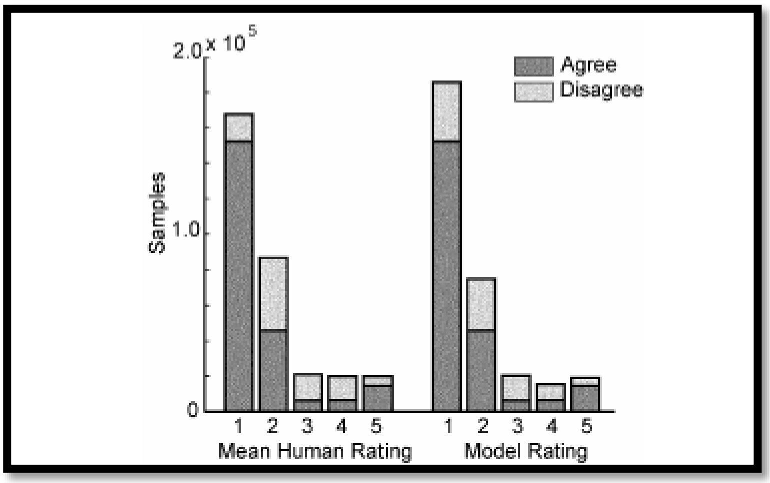


Figure 2.2e : Histogram comparing mean human rating and model output on the test set.

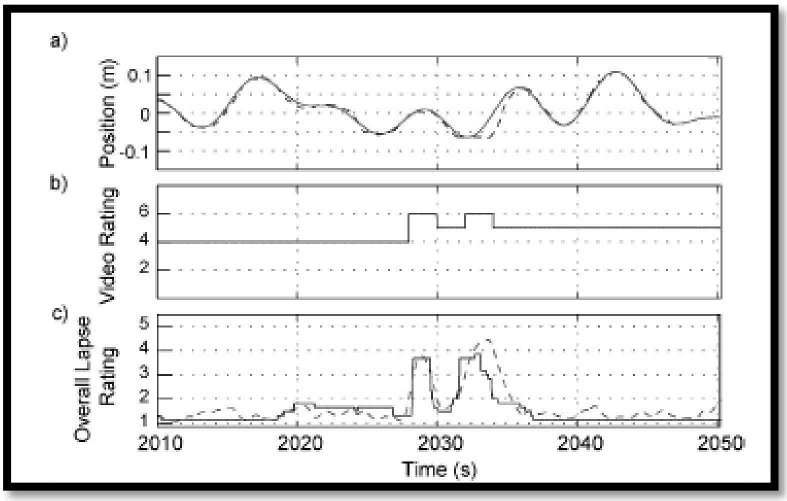


Figure 2.2f : Typical performance of rating

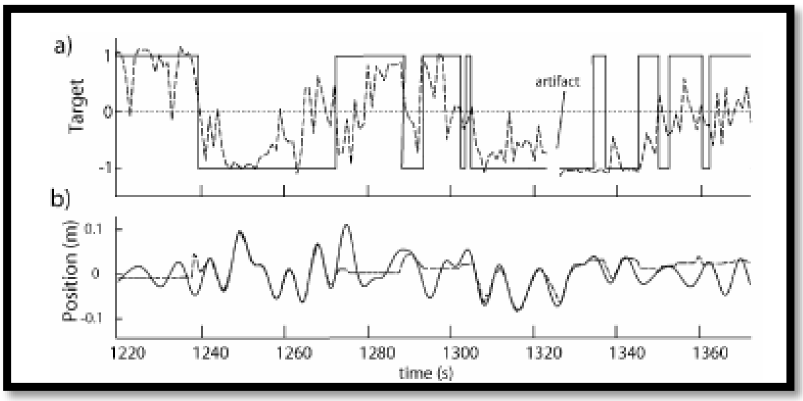


Figure 2.2g : Example of lapse detector performance

Malik T. R. Peiris, Richard D. Jones, Paul R. Davidson, and Philip J. Bones (2006) have been developed the behavioural microsleep detector from EEG Spectra. The research is focused on EEG Spectra which is to correlate with level of arousal and vigilantness in humans. This researched was fixate on eight non-sleep denied unremarkable subjects performed two 1-hour sessions of a perpetualtracking task while EEG and facial video were recorded. BMs are distinguished free of following execution by a human rater by viewing the video recordings. Spectral puissance, normalized spectral puissance, and power ratios in the standard EEG bands are calculated utilizing the Burg method on 16 bipolar derivations to compose an EEG feature matrix. PCA is utilized to reduce the dimensionality of the feature matrix and linear discriminant analysis used to compose a classifier for each subject [3].

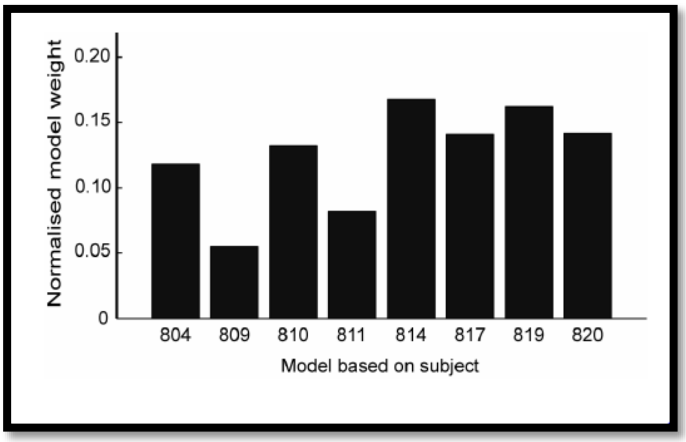


Figure 2.2h : Model weights calculated according to the generalization performance of each model obtained by classifying the data from the remaining subjects.

Wai Yie Leong, Danilo P. Mandic, Martin Golz and David Sommer (2007) have been developed the microsleep events for blind extraction. The aim of this study is to recognise the occurrence of microsleep events in an overnight driving task. They proposed a biosignal analysis method for the detection and extraction of microsleep events. This is achieved by employing blind source extraction method based on a cascaded nonlinear estimator to extract the relevant microsleep events. The cascaded nonlinear estimator jointly estimate the kurtosis and measure the nonlinearity and noise effects within the biosignal. This proposed method is applied to the electroencephalogram and electroculogram recorded of 12 young volunteers while performing monotonic overnight driving in a real car driving simulation laboratory. The extracted microsleep signals can then be used for driving simulation pilot studies for alertness monitoring, and to trigger the activation of alertness counter measures [4].

Govinda R. Poudel, Richard D. Jones, Carrie R. H. Innes, Richard Watts, T. Leigh Signal, Philip J. Bones (2009) is focus on the fMRI analysis of BMs revealed vicissitudes in haemodynamic activity in several cortical and sub-cortical regions associated with visuomotor control and arousal. The method that had been used was a multimodal method to investigate the neural correlates of BMs utilizing simultaneous recording of fMRI, eyevideo, VEOG, and perpetual visuomotor replication is presented. The data were accumulated from 20 salubrious volunteers while they performed a perpetual visuomotor tracking task inside an MRI scanner for 50 min. fMRI analysis of BMs revealed transmutations in haemodynamic activity in several cortical and sub-cortical regions associated with visuomotor control and arousal [5].

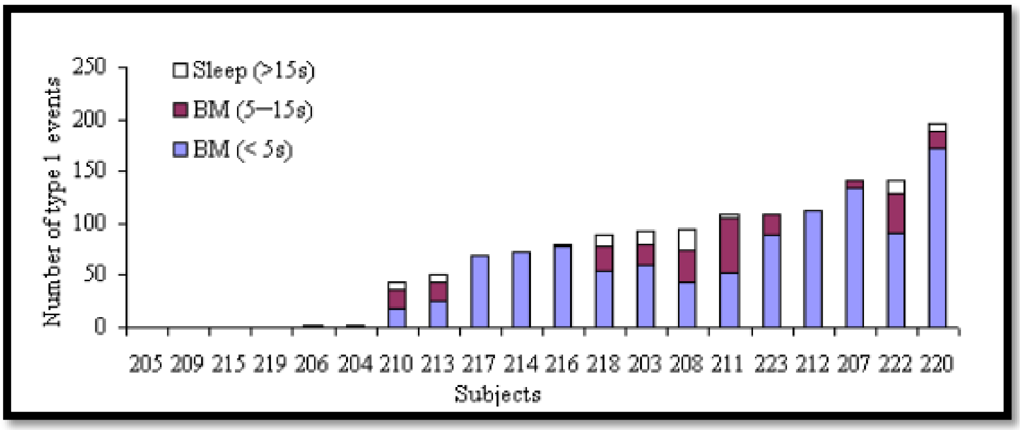


Figure 2.2i : Distribution of type 1 events across subjects. Numbers of short BMs, long BMs, and sleep episodes are plotted for each subject.

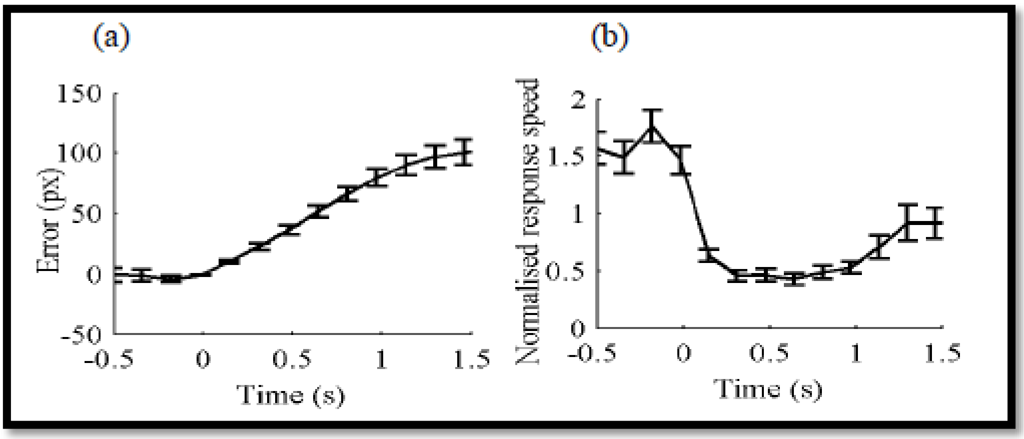


Figure 2.2j : Mean ± SEM of response error and normalized response speed. Normalised response speed.

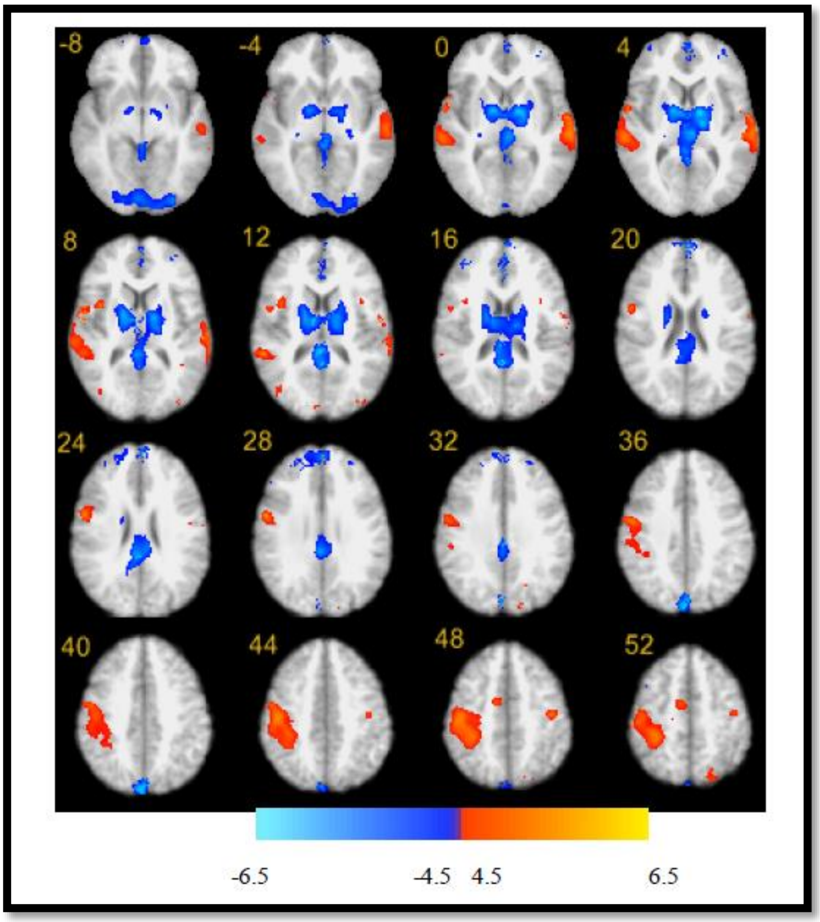


Figure 2.2k : The spatial pattern of group level BOLD activation (red) and deactivation (blue) during short BMs (< 5 s)

Carrie R. H. Innes, Govinda R. Poudel, T. Leigh Signal, and Richard D. Jones (2010) did the research on the behavioural microsleep in normally rested people. Slumber deprived people, or those performing elongated monotonous tasks, frequently have brief episodes when performance is suspended and they appear to fall asleep momentarily, behavioural microsleeps. Unintentional suspension of performance is categorically concerning in vocations in which public safety depends on elongated unimpaired functioning (e.g., drivers, air traffic controllers, health professionals, and process control workers). This study aimed to determine whether there is a relationship between propensity for BMs and measures of slumber. Subjects undertook a perpetual 50-min 2-D tracking task and BMs were identified with high temporal precision predicated on simultaneous analysis of visuomotor replication, tracking haste, tracking error, vertical electrooculogram, and ocular perceiver-video. BM rates and durations were correlated with measures of slumber [6].

SUBJECTIVE SLEEP PROPENSITY, SLEEP QUALITY, CIRCADIAN TYPE, AND HOURS AND EFFICIENCY OF SLEEP AS MEASURED WITH ACTIGRAPHY	Mean ± SD	Range
Epworth Sleepiness Scale score	6.7 ± 3.3	0–12
Pittsburgh Sleep Quality Index global score	4.3 ± 1.8	2–9
Horne-Ostberg Morningness-Eveningness Questionnaire total score	55.5 ± 9.4	31.0–71.0
Mean time in bed – 5 nights (h)	8.2 ± 0.7	7.1–9.4
Mean assumed sleep duration[a] – 5 nights (h)	7.8 ± 0.8	6.7–9.1
Mean actual sleep duration[b] – 5 nights (h)	6.8 ± 0.8	5.3–8.2
Mean sleep efficiency[c] – 5 nights (%)	82.8 ± 6.1	68.2–91.9
Actual sleep duration – night before study (h)	7.0 ± 1.1	4.9–8.8

Table 2.2a : The mean scores and ranges for the 20 participants on the ESS, PSQI, and Horne-Ostberg MEQ.

M. Golz and D. Sommer (2010) have studied that focused on monitoring and detection of drowsiness and microsleep (MS) during driving simulation. At first a framework is presented how to utilize methods of pattern recognition and of computational intelligence in order to detect and to predict MS. Secondly, different bio signals are compared due to their value for MS detection. Thirdly, results of different relegation algorithms are compared and the advantages as well as disadvantages of Support-Vector Machines for this task are highlighted. Fourthly, investigations of automatic pertinence tenacity are introduced. Fifthly, the quandary of the inter- and intra-individual variability is addressed. Cross validation analysis revealed that the EEG characteristics during MS are very concrete from individual to individual. Sixthly, one has to keep in mind that the MS detection algorithm has been established by learning from data. For this, only clear-cut examples of MS and Non-MS were culled. Seventhly, the MS density is presented as an incipient measure of lethargy and vigorous central fatigue. Eighthly, the application of their methodology to evaluate commercial fatigue monitoring technologies (FMT) is presented [7].

Amol M. Malla, Paul R. Davidson, Philip J. Bones, Richard Green and Richard D. Jones (2010) have studied on a contrivance that capable to perpetually monitor an individual's levels of vigilantness in authentic-time is highly desirable for obviating lethargy and microsleep cognate accidents. The research presents a development of non-intrusive and lighting sensitive video-predicated system that utilizes computer-vision methods to quantify facial metric for identifying visible facial denotements of somnolence and behavioral microsleep. The developed system utilized a remotely placed camera with a near-infrared illumination to acquire the video. The computer-vision methods are then applied to sequentially localize face, ocular perceivers, and ocular lids positions to quantify ratio of ocular perceiver closure. The system is evaluated in frontal images of nine subjects with varying facial structures and exhibiting several ratio of ocular perceiver closure and ocular perceiver gaze under plenarily dark and ambient lighting conditions [8].

Figure 2.2l : An example of a face region of interest (fROI) detection with Haarface detection algorithm implemented in OpenCV.

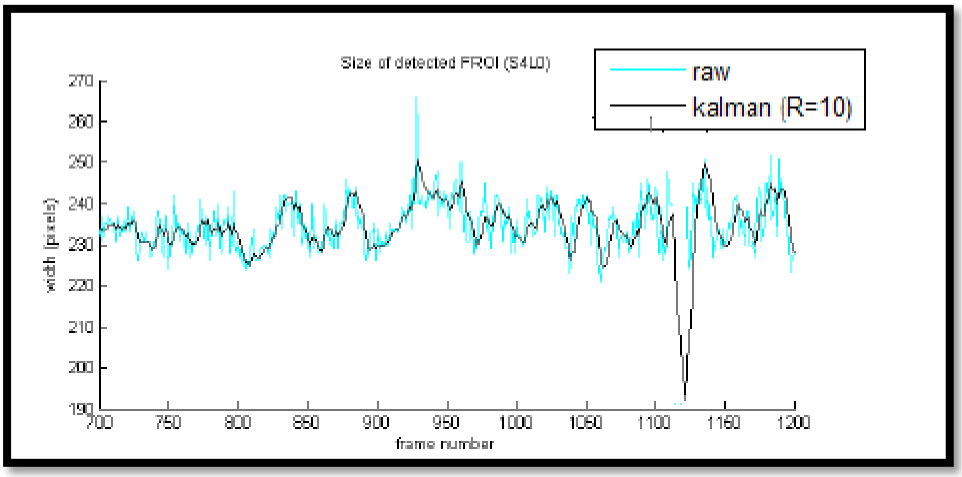


Figure 2.2m :Raw and Kalman filtered size parameter of fROI. Kalman filter stabilizes and also predicts missing fROI.

M. Holub, M. Šrutová, L. Lhotská (2015) developed Adaptable Microsleep Detection Based On EOG Signal for a Feasibility Study. Microsleeps are a very frequently discussed topic, due to their fatal consequences especially in the automotive industry. They can also be the cause of the accidents in air traffic and other control workplaces. These states of extreme drowsiness could be prevented by automatic detection or prediction. With that in mind, the classifier of MS was designed in this study based on the EOG analysis. The algorithm was proposed to be able to adapt independently to each analysed EOG signal. Finally, it was tested on 39 MS episodes and compared with the method based on fix thresholding. The aim of this study was to create a MS classifier which would be able to adapt independently to each of the analysed signals. Only EOG signals are used in the designed detection, which is based on the previous studyIn this manner, there is an extensive inconstancy in BM inclination in typically refreshed subjects which can't be clarified by variety in rest term, quality, or effectiveness. Be that as it may, BMs were not identified with other rest measures. [9].

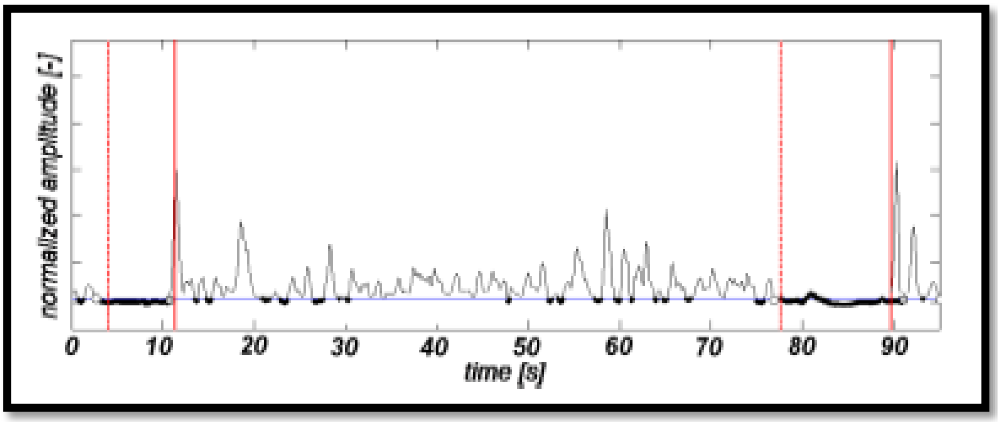


Figure 2.12n : The figure shows the MS detection with using an adaptable threshold

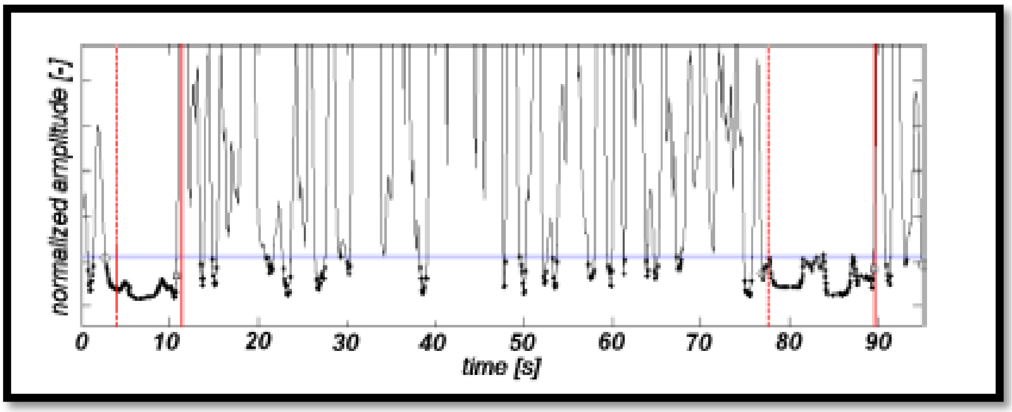


Figure 2.2o : The figure shows the detail of the automated MS detection with using an adaptable threshold

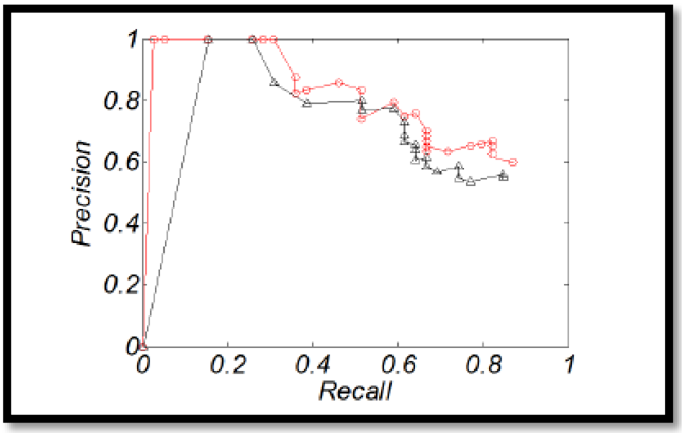


Figure 2.2p : The figure shows the curves Precision-Recall for the adaptable threshold THA (circles) and fix threshold THF (triangles).

Sudhanshu S. D. P. Ayyagari, Richard D. Jones and Stephen J. Weddell (2015) developed a microsleep detector using the echo state network (ESN) with leaky integrator neurons and compare its detection performance to that achieved in our earlier research with other methods. The methods that had been used are collecting data, EEG Analysis, Gold Standard, Feature selection and reduction and a performance evaluation. As for the data, Fifteen healthy male subjects aged 18–36 years (mean =26.5) were recruited. All subjects have visual acuities of 6/9(=20/30) or better in each eye. The EEG Analysis is used 50 Hz notch filter was applied to the EEG to remove power interference. Independent component analysis (ICA) was then applied to remove eye blink artefacts. For Gold Standard, a tracking task performance and video rating were the two independent measured that had been used. Furthermore, as the features selection and reduction, 544 power spectral features were reduced using PCA. Lastly, they did the performance evaluation on performance of the microsleep detection system was calculated [10].

%	Single classifier modules		
	LDA	*Sigmoidal ESN*	*Leaky ESN*
Sensitivity	64.2	66.3	76.6
Specificity	92.0	96.8	95.2
Selectivity	29.3	20.7	45.1
Phi (φ)	0.31	0.20	0.38
%	**Stacked generalization (ensemble modules)**		
	LDA	*Sigmoidal ESN*	*Leaky ESN*
Sensitivity	73.5	61.9	85.8
Specificity	92.0	96.4	94.0
Selectivity	31.3	28.5	53.6
Phi (φ)	0.39	0.27	0.51

Table 2.2b : Microsleep detector performance

2.3 Comparision of Microsleep Detection Development

Author	Methodology	Applications	Advantages	Disadvantages
M. T. R. Peiris, R. D. Jones, P. R. Davidson, P. J. Bones, D.	*EEG *EOG *Facial video	*Drivers	* useful in characterizing EEG *changes in bipolar disorder and	*EEG marks of rest in the transitional (stages 1 and 2) are not all around corresponded with behavioral rest.

J. Myall (2005)			during epileptic seizures. * EEG tends to become less complex during lapses.	*correlation effect might be due to slow variations in alertness level rather than lapses.
P. R. Davidson, R. D. Jones and M. T. R. Peiris (2005)	* EEG * LSTM Recurrent Neural Networks	*Committees	*overcome the "vanishing gradient" issue influencing most other intermittent neural system architectures. *sessions included a greater proportion of deep lapses, making them easier to detect.	*the detector is not yet sufficiently reliable for general use.
Malik T. R. Peiris, Richard D. Jones*, Paul R. Davidson, and Philip J. Bones (2006)	*EEG power spectra.	* Truck drivers, locomotive drivers, pilots, air traffic controllers, and surgeons.	* PCA was utilized to decrease the dimensionality of the element framework and straight discriminant examination used to frame a classifier for every subject.	* the execution of an EEG unearthly power based BM identifier was, by and large, unobtrusive. *notwithstanding, identification of BMs at this level of temporal determination

Wai Yie Leong, Danilo P. Mandic, Martin Golz and David Sommer (2007)	*EEG *EOG *ECG *EMG	* Driving simulation pilot.	*the performance of the algorithm is demonstrated. *deflation stage to remove the artifacts, nonlinearities and noise.	*complex method (cascaded nonlinear estimator to extract the relevant microsleep events). *too many formula in the procedure.
Govinda R. Poudel, Richard D. Jones, Carrie R. H. Innes, Richard Watts, T. Leigh Signal, Philip J. Bones (2009)	* fMRI, *Eyevideo *VEOG	* Truck drivers, locomotive drivers, pilots, air traffic controllers, health professionals.	* facial video and visuomotor responsiveness and found a high rate of BMs during an extended visuomotor task.	*failure to respond can occur frequently during any attention-demanding task. * Slowed responses in the rested-awake state stem primarily.
Carrie R. H. Innes, Govinda R. Poudel, T. Leigh Signal, and Richard D. Jones (2010)	*EEG *fMRI	* Individuals who are sleep-deprived.	*BMs occurred frequently during the task. *sustained-attention lapses can be identified response failures.	*Subjects undertook a continuous 50-min 2-D tracking task. *BMs were identified with high temporal accuracy. *Response failure and delayed

				reaction times during discrete auditory/visual detection tasks.
M. Golz and D. Sommer (2010)	*EOG *EEG	*Drivers.	*EEG characteristics during MS are very specific from individual to individual. *MS detection algorithms works when applied consecutively.	*EEG and EOG signals lead to low error rates of classification between MS and Non-MS. *EEG in detecting MS is a very limited approach.
Amol M. Malla, Paul R. Davidson, Philip J. Bones, Richard Green and Richard D. Jones (2010)	* Video-based system.	*Commercial truck *bus drivers *Pilots *air-traffic controllers, and medical house-staffs can lead to disastrous consequences.	* Videos were collected using QuickCam 4000 webcam whose IR block filter was removed. * The videos with 640 x 480 pixel image resolution were collected at 30 fps.	* the compound errors of each facial detection method resulted in unsatisfactory performance of final eye closure metric for monitoring alertness.
M. Holub, M. Šrutová, L. Lhotská (2015)	*EOG	*Committees	*EOG signals were acquired with pair electrodes placed	*complex formulas and method.

			on temporal bone. *obtain more data for the further verification and extension.	
Sudhanshu S. D. P. Ayyagari, Richard D. Jones and Stephen J. Weddell (2015)	*EEG *ESN	*Driver Motor Vehicle.	*ESN provide an architecture and supervised learning principle for recurrent neural networks.	*ESN exploits long-time dependencies in transient signals and attains higher memory spans. *poor performance.

Table 2.3 : Summary of Previous Microsleep Detection Development

2.4 History of Arduino

Arduino started in 2005 as a project for students at the Interaction Design Institute Ivrea in Ivrea, Italy where at that time, programs student used a basic stamp which considered expensive at $100. Massimo Banzi is one of the founders, taught at Ivrea. Ata that time, the founders usually meet at the bar to discuss about the projects and was named as Arduino. In return, Arduino of Ivrea is name of the bar who was the margrave of Ivrea and King of Italy from 1002 to 1014. At that time also, the wiring development platform is created by Colombian student Hernando Barragán which served as the basis for Arduino. David Cuartielles give the idea to promote the completion of the Wiring platform to the open-source community, associated researchers which is lighter and less expensive versions were created then made available.

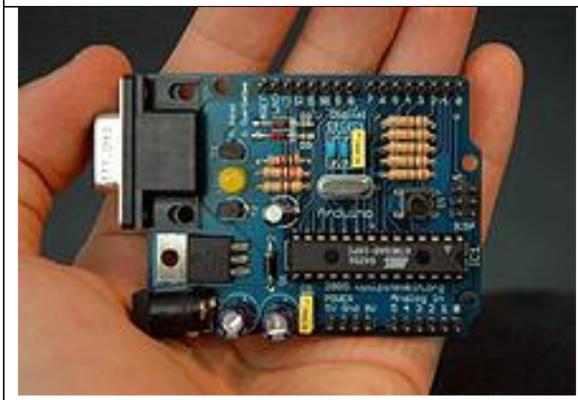
Arduino Boards	Features
 Arduino Atmel ATmega8 microcontroller chip	Consists of an Atmel 8-, 16- or 32-bit AVR microcontroller facilitate programming and incorporation into other circuits.with complementary components
 Arduino Uno	provide 14 digital pin included input and output pins has six analog inputs These inputs can also be used as six digital input and ouput pins. These pins are on the top of the board, via female 0.10-inch (2.5 mm) headers.

Table 2.4 : The types of Arduino Boards

2.5 Pulse Rate System

The first individual to precisely gauge the beat rate was Santorio Santorii who designed the pulsilogium, a form of pendulum, based on the work by Galileo Galilei. Pulse rate is the rate at which your heart beats. Pulse rate is usually called your heart rate, which is the number of times your heart beats each minute (bpm). But the rhythm and strength of the heartbeat can also be noted, as well as whether the blood vessel feels hard or soft. In the other hand, the heart rate is determined by auscultation or audible sounds at the heart apex, in which case it is not the pulse. The heart rate might be more noteworthy or lesser than the pulse rate contingent on physiologic request.

2.5.1 Normal pulse rate

For children over 10 years old, adults 18 and older, a normal resting heart rate is between 60 and100 beats per minute (bpm). However it depends on the person on what they are doing or physical condition, weather, surrundings and age. For children ages 1 to 10, the normal resting heart rate is between 70 and 130 bpm.

newborn (0–3 months old)	infants (3 – 6 months)	infants (6 – 12 months)	children (1 – 10 years)	children over 10 years & adults, including seniors	well-trained adult athletes
100-150	90–120	80-120	70–130	60–100	40–60

Table 2.5 : The pulse rate for different age

2.5.2 Microsleep & Sleepy pulse rate

A person is in state microsleep when his heart beat or pulse rate is in between 40bpm until 50 bpm.

2.6 Radial Basis Function (RBF) Neural Network

A radial basis function (RBF) neural network is a real-valued function that value depends only on the distances from Matlab software This technique is being applied to classify the pulse rate between normal and sleepy pulse rate. By using the test parameter, the data is tried to include all of the parameter to get the data between microsleep and normal pulse rate. The neural network start have a few category that can be used to characterized the data set such as input-output and curve fitting, pattern recognition and classification, clustering and dynamic time series Radial basis functions are usually used to build up function approximations of the form.

$$y(x) = \sum_{i=1}^{N} wi \, \emptyset(||x - xi||)$$

where the approximating function y(x) is represented as a total of N radial basis functions, each associated with a different center(xi), and weighted by an appropriate coefficient (wi). The approximating function is linear in the weights that why the weights can be estimated by using the matrix methods of linear least squares.

CHAPTER 3
METHODOLOGY

3.0 Introduction

This chapter covers the methods that will be used in this project to achieve the objective of analysing the performance of microsleep detection accuracy by using biosensor module and pulse rate. All the steps taken to develop the microsleep detection system by using biosensor module and pulse rate, wireless system, sensor, pulse rate and arduino and the basic coding for the arduino by using integrated development environment(IDE) that support the C and C++ programming languages and others hardware are described.

3.1 Project Implementation

As ECG Sensor became the major part for this project, it used Arduino and ECG shield to detect the pulse rate for microsleep detection. In the other hand, pulse rate that has been detected by the ECG sensor is show the pulse rate of microsleep through the pulse amped sensor on the PC. The initial step is to gather information on the development of microsleep from journal articles, books, newspaper and internet. Then, researches on the arduino and coding for basic arduino software by using integrated development environment (IDE) that support the C and C++ programming language is carried out to understand and create coding to be programmed on the arduino device. Next, design the hardware for the microsleep detection system by using biosensor module and pulse rate. Precaution measure that must be taken during the design of the hardware is identified by performing a survey on equipment, tools and project utility that will be used. Then, the coding of arduino is written by using IDE software that will be embedded into the hardware. Troubleshooting is performed if the coding is not suitable with the hardware. Last, a full report is prepared and submitted to the supervisor. This project shows the methodology overview on Figure 3.1a.

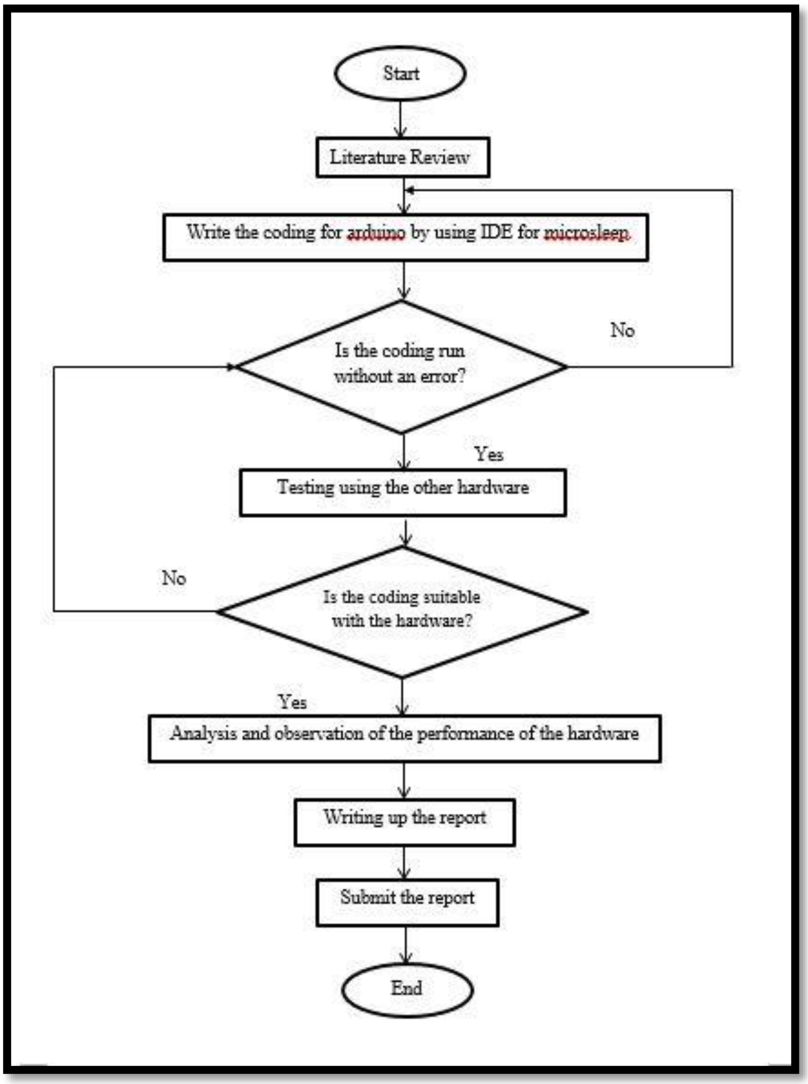


Figure 3.1a : The project flowchart of the methodology.

The title research for a project need a long time and make title comparison and aim unequal title and this project is new innovations in engineering technology industry. Subject searches for the project can go through journals, magazines, books and internet so on. Title chosen is development of microsleep detection system by using biosensor module and pulse rate. This project starts study on the microsleep by using any type of methods, Arduino, ECG sensor and matlab neural network. The sources of the literature review are finding them from internet, university library, journals and from supervisor itself. The hardware using for microsleep detection system has been researched. This project consists of five equipments which are PC, Arduino ECG Sensor and Matlab Radial Basis Function (RBF) neural network.

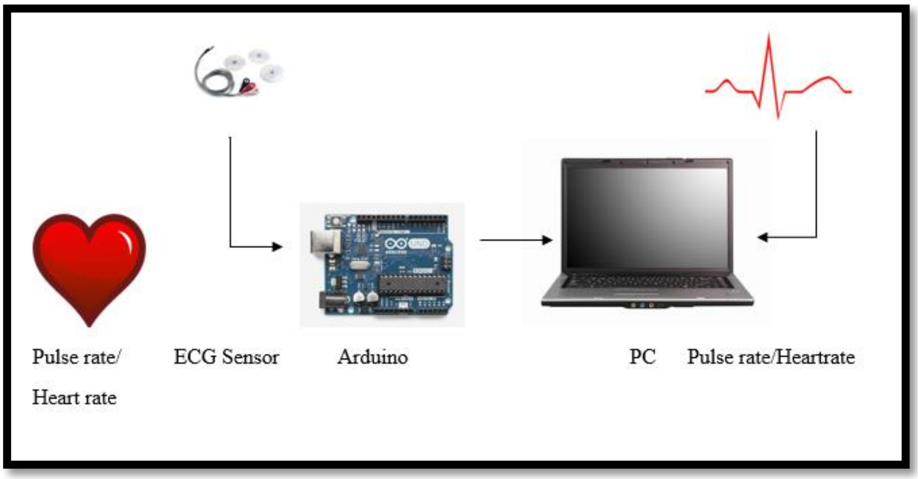


Figure 3.1b : The Project Diagram

This project is beginning with the pulse rate which is detected by using the ECG sensor in three parts of body which are right wrist, left wrist and leg. ECG sensor is the electrocardiogram (ECG) that allow to access the electrical and muscular function of the heart. The pulse that has been detected by the ECG Sensor will go through the arduino to trigger the pulse rate by using the programming in the integrated development environment (IDE). The pulse rate will show the heartbeat waveform with the pulse rate read in the pulse rate amped visualizer. Next, the pulse rate in the PC will be display in two classes which are sleepy and normal pulse rate. The classification process will be done with the use Radial Basis Function (RBF) neural network in the Matlab. The beats per rates (bpm) for the sleepy pulse rate is 40-50 bpm while normal pulse rate is 60-100 bpm.

3.1.1 Arduino

There are several softwares used in this project and they must be downloaded. First, the firmware was downloaded from the website shown in Figure 3.1c.

Figure 3.1c: Arduino main website

The next step was to click download Arduino version 8 as shown in Figure 3.1d where it automatically downloaded the PC.

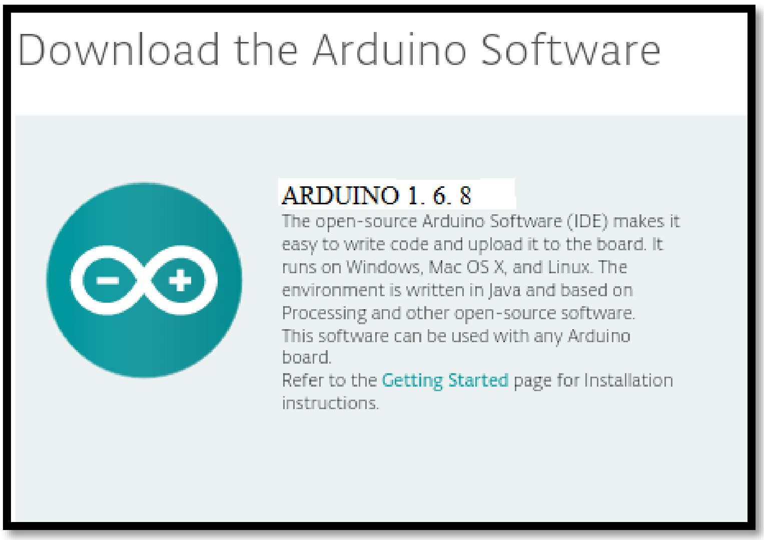


Figure 3.1d: Click download the Arduino Software

Secondly, the Arduino software is being install in the PC by clicking Yes in the Figure 3.1e below.

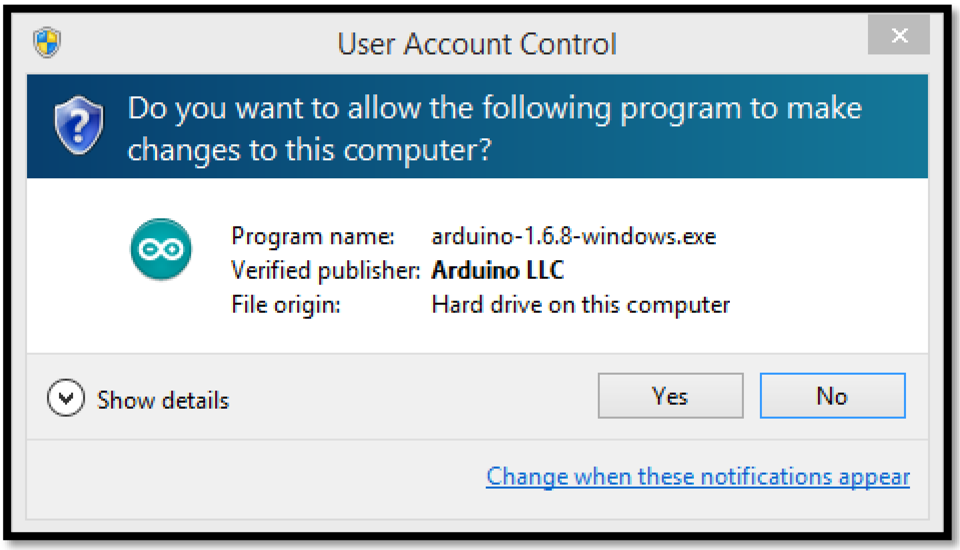


Figure 3.1e: Arduino software ask for permission to being install.

Last but not least, the Arduino is installing in the PC and the Arduino shortcut is save in desktop as shown in Figure 3.1f.

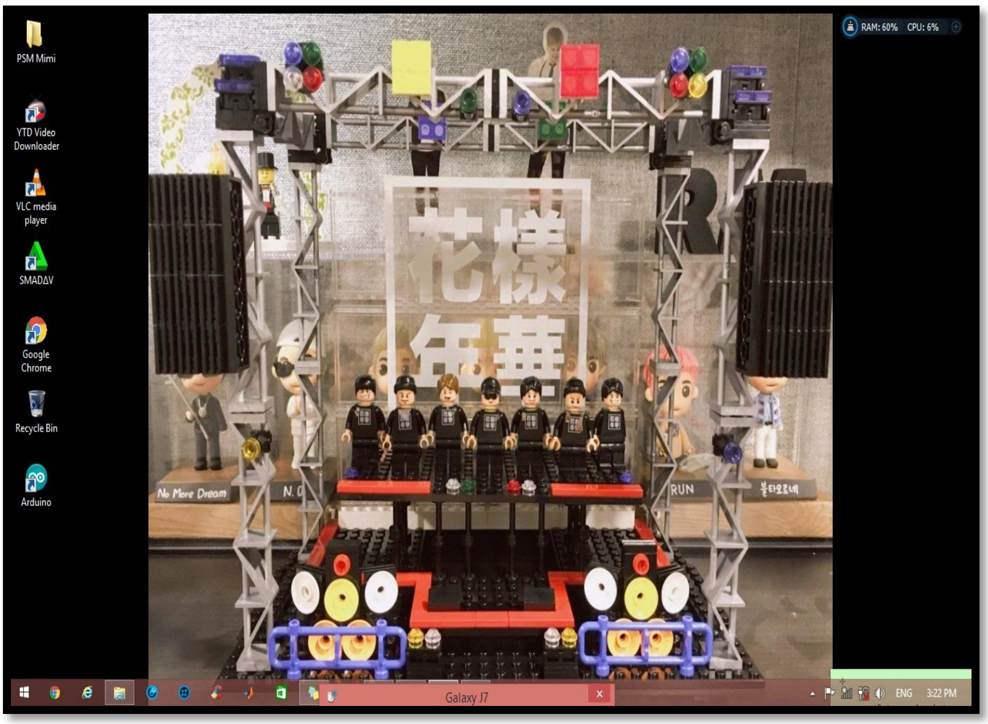


Figure 3.1f: The Arduino Shortcut is on the PC desktop

3.1.2 Arduino Uno Code Development

This project used the Arduino Uno as the main microcontroller that control the functionality of the relay which then control the central locking system for the ECG Sensor.in order to develop this program, an Arduino IDE must be installed into the PC. This Arduino IDE is the platform for writing code that developer want to insert into the Arduino device. Figure 3.1g below shows the code to test the Arduino in the Arduino IDE.

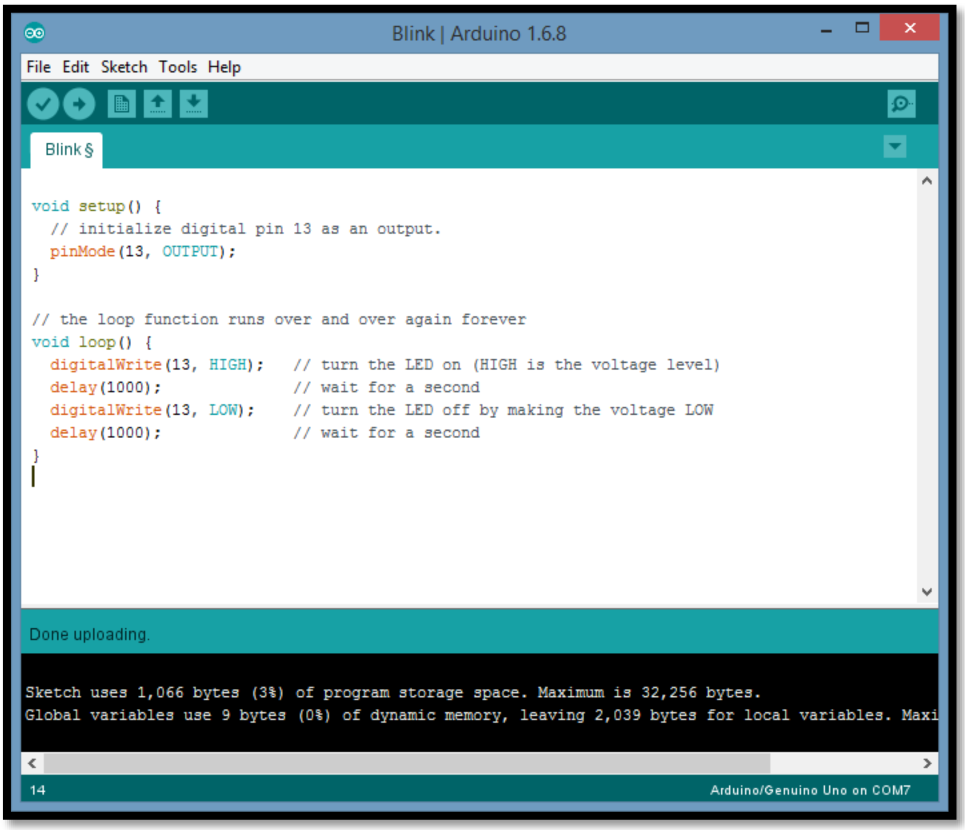


Figure 3.1g : The Arduino Code to test the blink of the LED

The Arduino IDE is basically a platform for developer or Arduino user to program their code and uploading the program code into the Arduino chip. Before able to upload the code into the Arduino, the serial port of the Arduino must be selected first. If the serial port is not detectable, use the update driver to make the Arduino is detectable with the serial port. One precaution step whenever uploading program to Arduino is always verify the code first before upload.

3.2 ECG Sensor

In this project, ECG Sensor has been used to detect the pulse rate in three parts of the body which are the right wrist, left wrist and leg. The pulse rate is taken in beats per minutes (bpm). The person for every person is quite different for each person regardless of whatever they are doing. The person who work at office, the BPM is 60-100 bpm while the athletes will have the bpm is around 40-60 bpm. The people with different ages are also need to pay intention more because different age also have different bpm. The children with age of 1-10 years old have the bpm around 70-130 bpm. The Figure 3.2a below show the ECG Sensor that has been used in this project.

Figure 3.2a : The ECG Sensor

3.2.1 ECG Sensor with Arduino

The ECG Sensor is connected to the Arduino Uno by using EKG-EMG Shield as the Figure 3.2b below. The ECG Sensor code to read the pulse rate detcton is written into Arduino IDE by using the correct coding so that it can detect the pulse rate well.

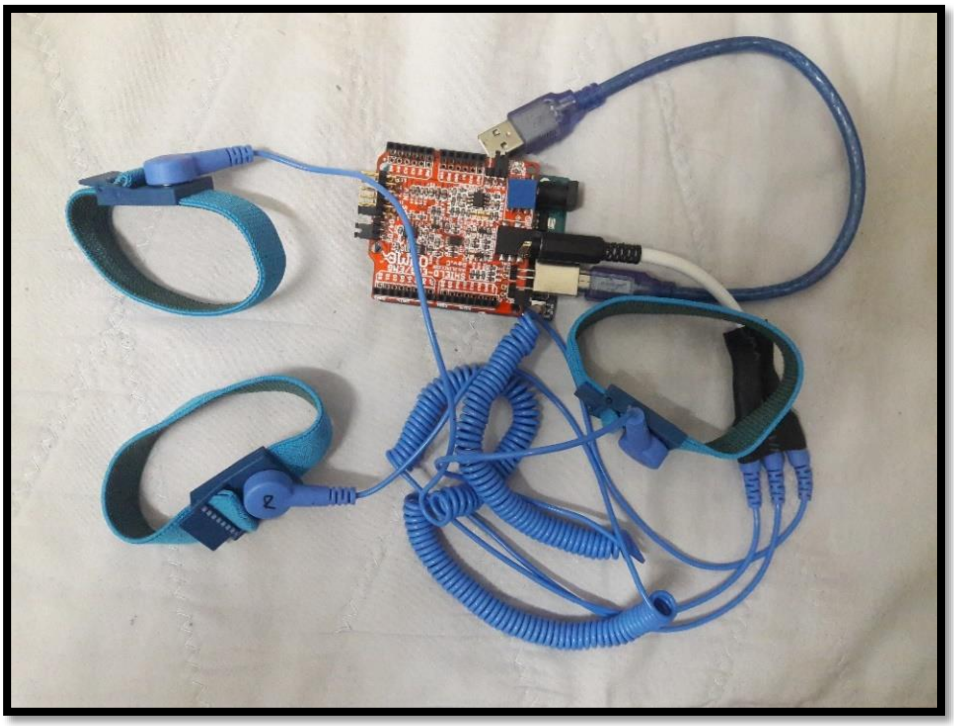


Figure 3.2b ; The ECG is connected to the Arduino

3.2.2 ECG Sensor with Arduino and Buzzer

In order to make sure that the microsleep detection system works, I have added the buzzer to give notification if the sensor detect the microsleep condition. The microsleep beats per minutes (bpm) in between 40bpm until 50 bpm. When the ECG Sensor achieved that bpm, the buzzer will continuously ringing. However, if the ECG Sensor in normal condition, the buzzer will stop ringing.

The connection of the Buzzer to Arduino listed below :

Positive → (Pin 9)

GND → GND

In Arduino IDE, the Pin 9 is set as the variable to declares the pin will use in the process of the arduino. The pin is declared as int ringBuzzer=9; while the pin is also written into void setup to make sure that the process of the pin 9 is valid and the pin will ring when it meet my condition as pinMode (ringBuzzer,OUTPUT).

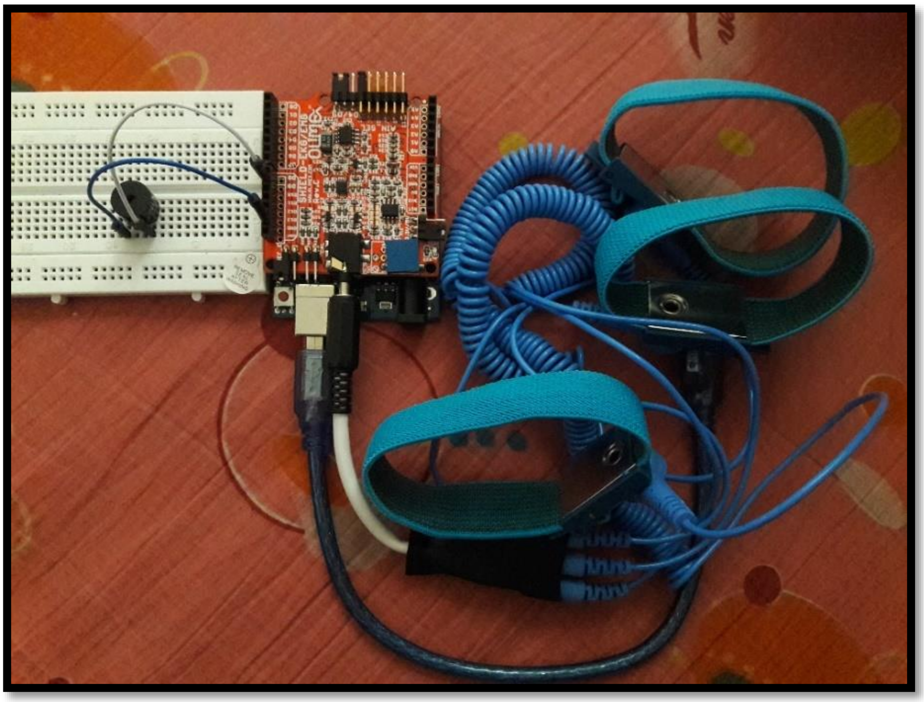


Figure 3.2c : The connection between buzzer and Arduino

3.3 Test Parameter

3.3.1 Matlab Implementation

In this project, the implementation of the Matlab which is Radial Basis Function (RBF) is used in order to insert the data of the parameter of the pulse rate. The parameters that will effects the microsleep which are pulse rate itself, weather, temperature, work, gender, time, body condition, surroundings and age. These parameter is important to be included in the study because these parameter need to be considered in many way to make sure that the data will be get is correct. First and foremost, the parameter that need to be concentrate is the BPM range for each people that have different kind of work in daily life. This kind of parameter should be noted because each of the work is having different of BPM range of pulse rate. For example, the persons who work at office and the person that workout in their daily will have different normal pulse rate. The person who work at office, the BPM is 60-100 bpm while the athletes will have the bpm is around 40-60 bpm. The people with different ages are also need to pay intention more because different age also have different bpm. The children with age of 1- 10 years old have the bpm around 70-130 bpm. These test parameter need to be studied because the microsleep will happen to all person not only working people, but also the athletes. Even the student will experience the microsleep no matter what gender they are. So the study about this parameter is needed to make sure that, the person that have different work, age, gender and different normal pulse rate will having the microsleep.

3.4 Characterization / Testing Technique

The testing technique that will apply in this project is by using Matlab software which is Radial Basis Function (RBF) neural networks. This technique is being applied to classify the pulse rate between normal and sleepy pulse rate. By using the test parameter, we try to include all of the parameter to get the data between microsleep and normal pulse rate.

CHAPTER 4
RESULT & DISCUSSION

4.0 Introduction

In this chapter, the process of testing and analysis are presented including the performance of the system and the accuracy of the performance. The results of this project which are the outcomes from hardware setup will be represented. In this chapter, the analysis will be divided into two parts which are the analysis of software part and the analysis of hardware part.

4.1 Project Hardware Setup

The projects hardware layout consist of the connection of the real component using breadboard. The assembly of the com[onenet are directly according to the schematic that I have made.

4.1.1 Component Layout

Figure 4.1a illustrates the component assembly on breadboard. One buzzer is used to ring when the microsleep condition is detected. The microsleep beats per minutes (bpm) in between 40bpm until 50 bpm.

The connection of the Buzzer to Arduino listed below :

Positive →(Pin 9)

GND→GND

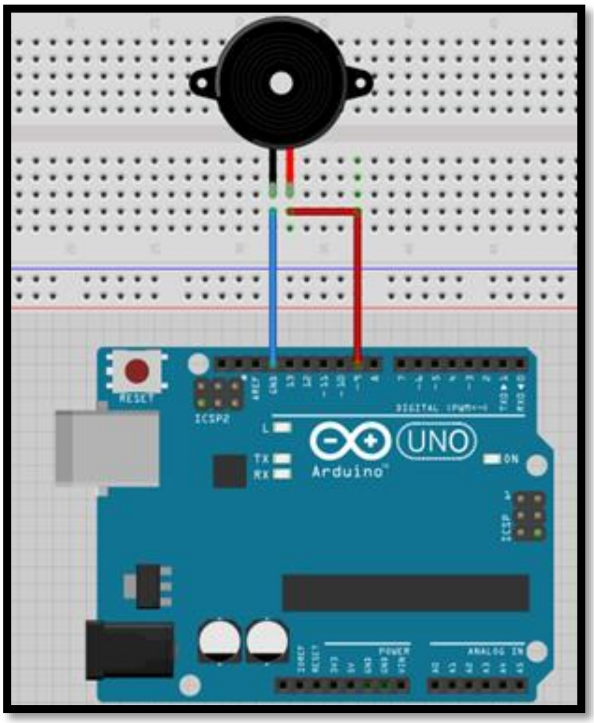


Figure 4.1a : The componenet layout of Buzzer

4.1.2 Hardware Setup

In the Figure 4.1b, the figure shows the connection the overview of the object connection between Arduino, ECG Sensor and PC without prototype. The raw connection just connected to each othe on the breadboard.

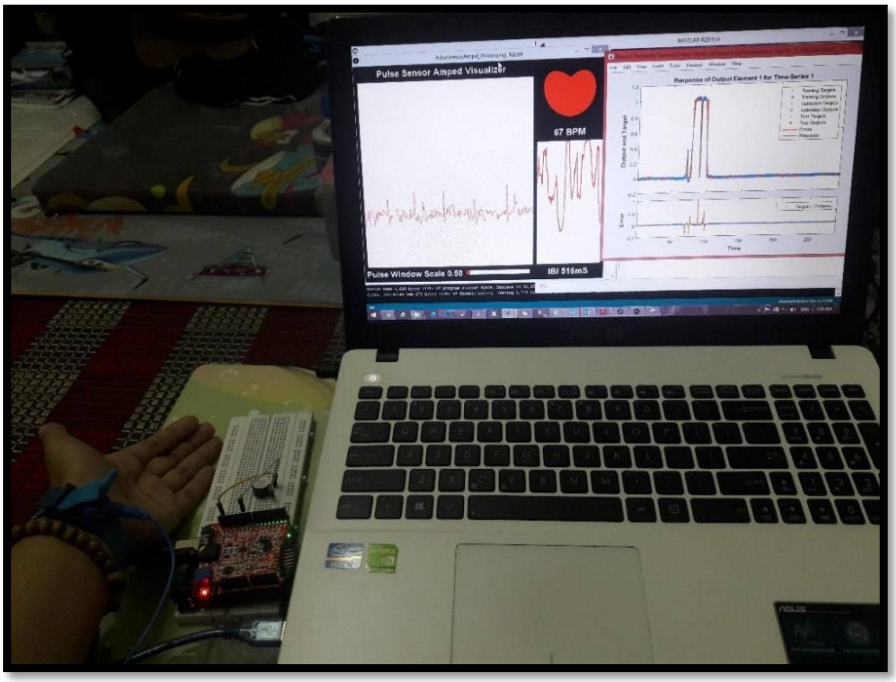


Figure 4.1b : The connection without prototype

Below in the figure 4.1c shows the connection betwwen the equipment hardware in a prototype connected with the PC. In the figure there is no more the breadboard and the wire connection make it more tidy.

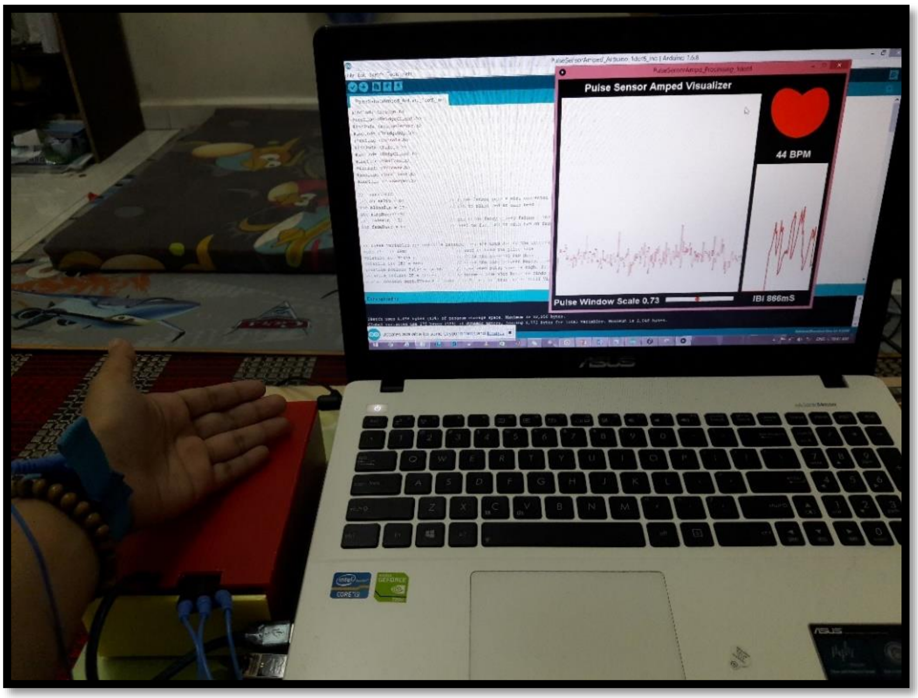


Figure 4.1c : The connection with prototype

4.2 Analysis of Testing Part

4.2.1 Matlab Neural Network Testing

In Matlab Neural Network Testing, first of all, I insert the data of bpm of the pulse rate in the Matlab function. But, before that, I try do the simulation of the neural network start to familiar with the process of neural network before inserting my data. The neural network start have a few category that can be used to characterized the data set such as input-output and curve fitting, pattern recognition and classification, clustering and dynamic time series as Figure 4.2a below.

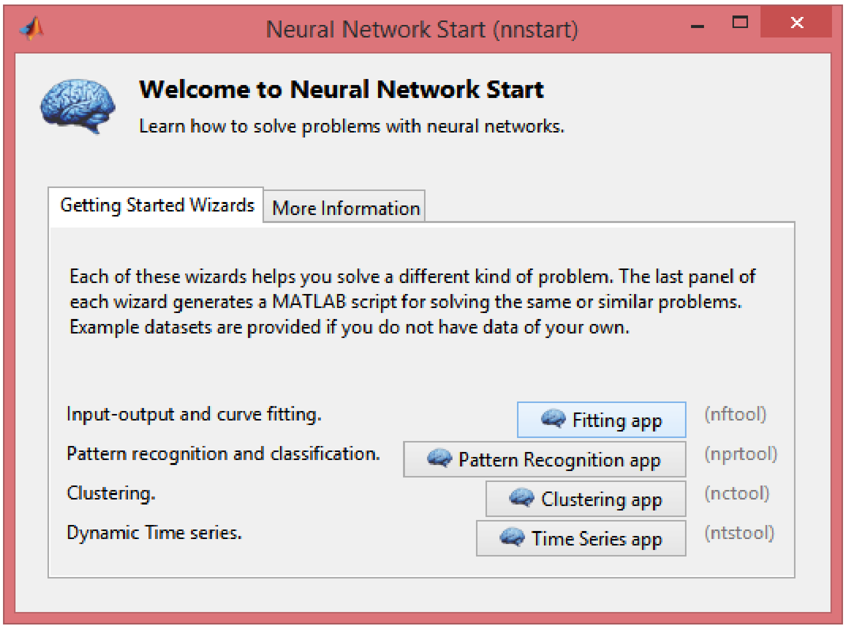


Figure 4.2a :Neural Network Start

After try all of the neural network start, I have choose Time Series application to use my dataset to be characterized and use as my testing technique. Time Series Application has been chosen because it make me easier to make the prediction based on my dataset. As information, prediction are used to predict future values. Generally, it is a kind of dynamic filtering in which past values of one or more time series. Dynamic neural networks are used for nonlinear filtering and prediction which include tapped delay lines.

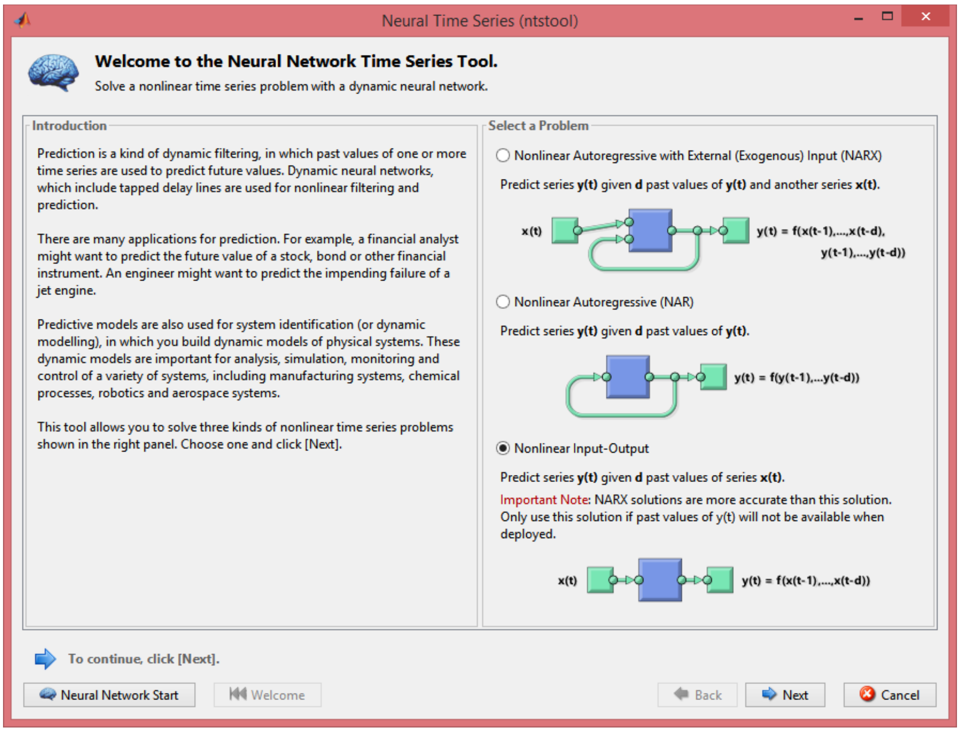


Figure 4.2b : Three kinds of non-linear time series problems

These dynamic models are important for analysis, simulation, monitoring and control of a variety of systems, including manufacturing systems, chemical processes, robotics and aerospace systems. Non-linear Input-Output has been chosen as the type of problem time series because the data set that I had are involved both input and output which are Pulserate inputs and Pulserate Targets.

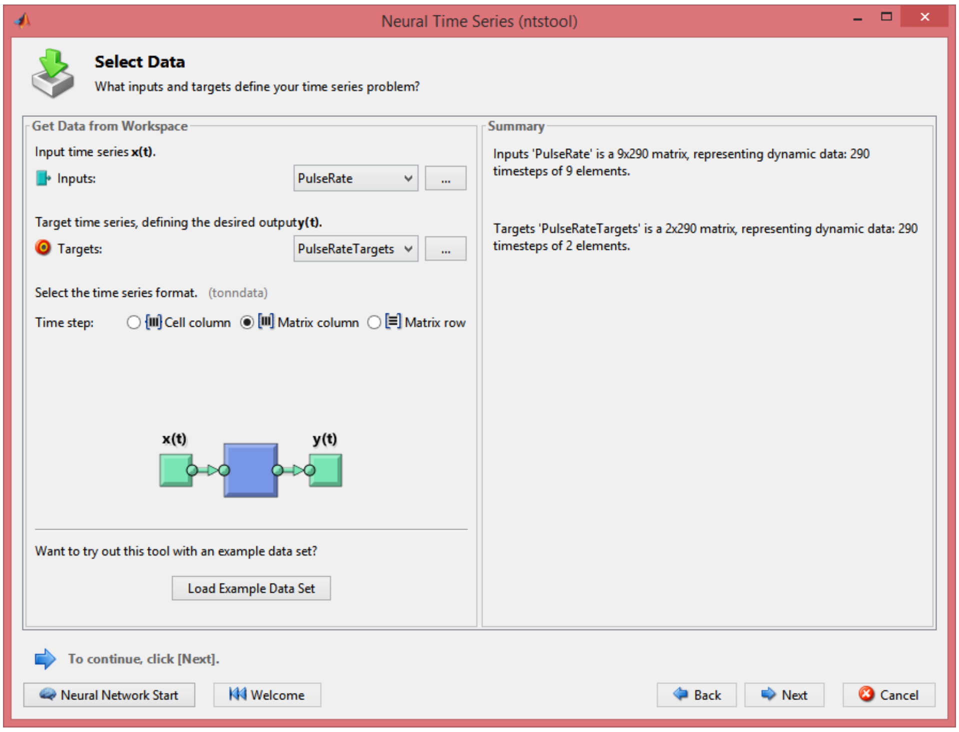


Figure 4.2c : Data set inserted to the Non Linear Input and Outputs

My data must have the parameter that will effects the microsleep which are pulse rate itself, weather, temperature, work, gender, time, body condition, surroundings and age. Inputs 'PulseRate' is a 9x290 matrix, representing dynamic data: 290 timesteps of 9 elements where Targets 'PulseRateTargets' is a 2x290 matrix, representing dynamic data: 290 timesteps of 2 elements. All the data is inserted by only number while for some parameter I just used 0 and 1 to state the value such as work, body condition and gender because it whether the person is working or not. In the data in Matlab also, I stated the target input 0 and 1 whether it yes or not. There are two set of the targets input, which are if one of it 1, the other is 0 to see the difference between them when the pulse rate is read from ECG sensor.

I test my data set of the pulse rate in ECG Sensor too detect the microsleep by using Time Series Application. Time Series application is suitable to use in my testing technique because it can help me to do prediction of my dataset. The testing and validation data also can be set through this application. In my testing, I used 70% of my overall training which is 406 targets time steps from 506 targets for overall dataset. As validation and testing process, I only use 15% of the overall dataset for each set which are 87 targets time steps. The Figure 4.2d below show the completed Validation and Test Data.

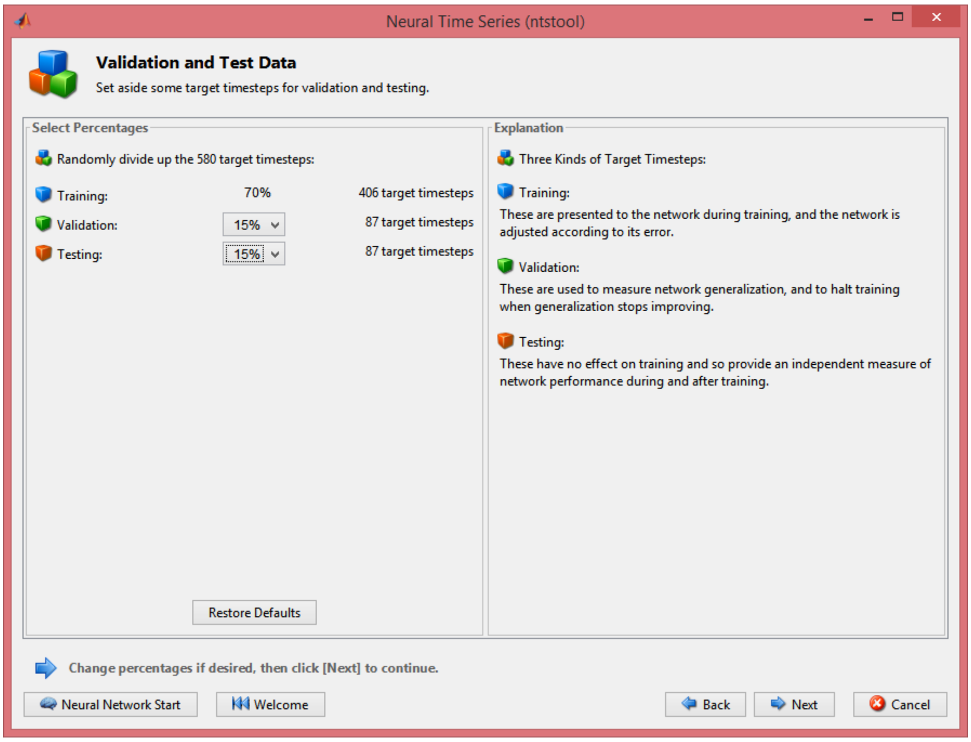


Figure 4.2d : The Validation and Test Data

The Figure 4.2e shows that the step in the network architecture. In the testing technique. The important step is about network architecture which I must choose the number of neurons and inputs delays. The number of hidden neurons that I set is 10 while the number of delay d is 2. The number of hidden neurons and the number delay can be if the network does not perform well after training.

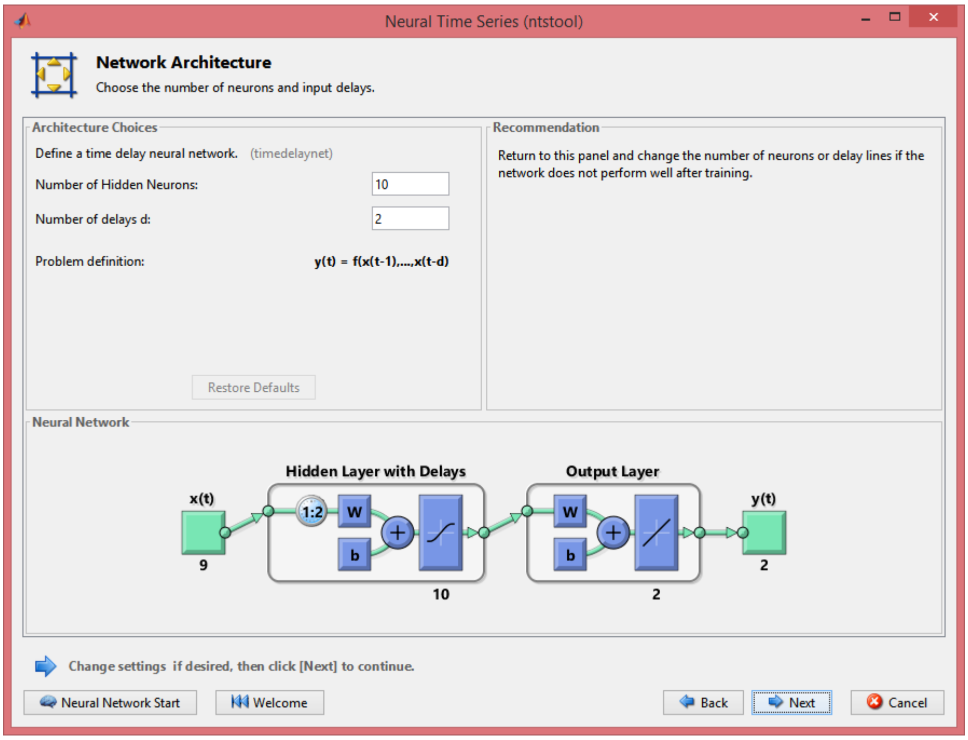


Figure 4.2e : The Neural Network Network Architecture

The training network is used to train the network to fit the inputs and targets. Training algorithm that I chose to use is Levenberg-Marquardt because this training typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples. By using this type of training, the process is faster and there is validation dataset can be detected. Figure 4.2f show the network train details.

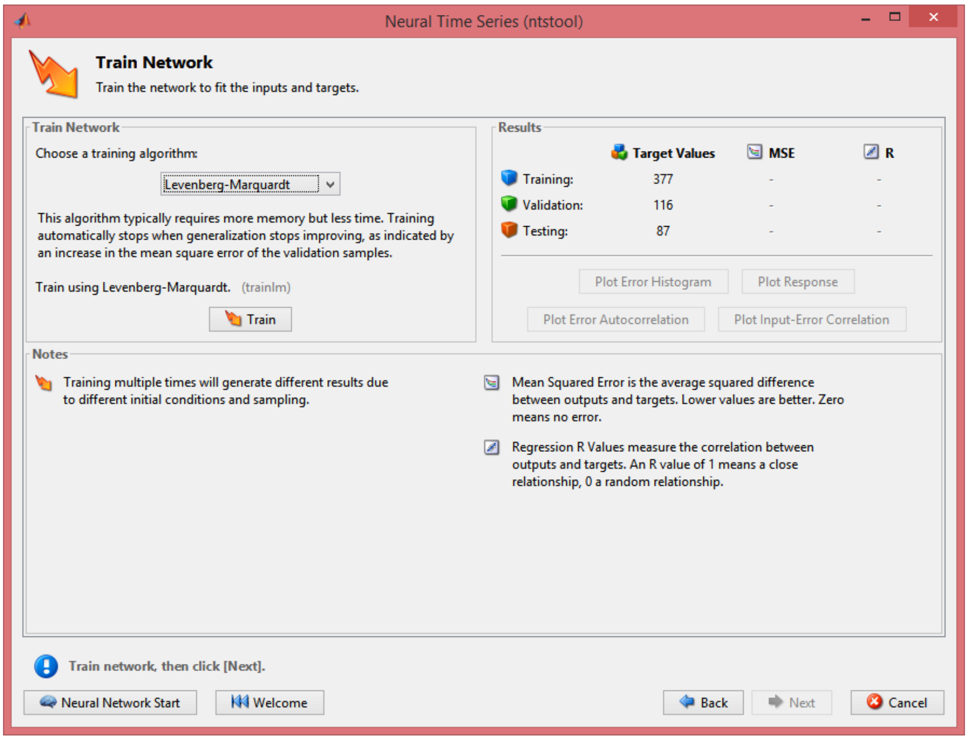


Figure 4.2f : The Training Network for Neural Time Series

By using Levenberg-Marquardt because this training typically requires more memory but less time. As the figure below shows the training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples. The training stops at 72 iterations of 1000 samples because the training already found the validation data for 6 data. The time for the training to finish also are very fast because the training only needs 3 seconds to find the validation data far different between Bayesian Regularization which This algorithm theoretically requires more time, but can result in good generalization for difficult, small or noisy datasets. Furthermore, in the training neural network, there are many analysis type that can be done such as performance, training state, error histogram, regression, time series response or even error auto-correlation. The analysis has different type of result but used the same inserted data before. Figure 4.2g below shows the process of the training neural network.

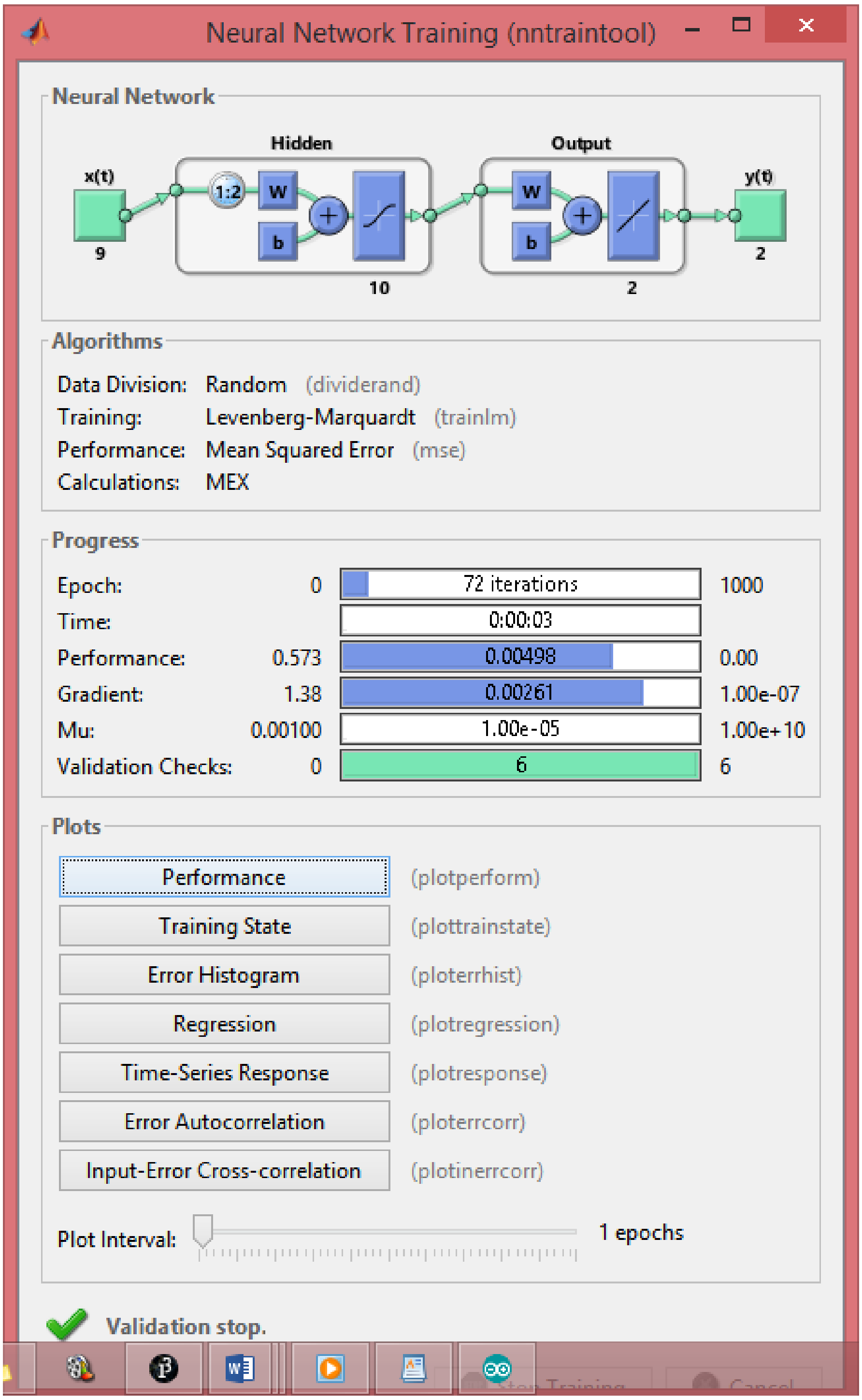


Figure 4.2g : The Neural Network Training

The Figure 4.2h shows the performance of the pulseRate inputs and pulseRate Targets of the data inserted in the matlab training. The best validation performance is at epoch 66. The graph is between epochs and mean squared error. The performance of the data is quite nice because the each of the category process of the data is quite similar and the interception data is determined at the end of the process. The graph below show that the higher the mean squared error, the lower the epochs. Alongside the data training, it performance seem to be decreases as the mean squared error is lower. However, the test data seem quite overlap with the training and validation data. Maybe, some of the data are invalid in the used of the microsleep detection.

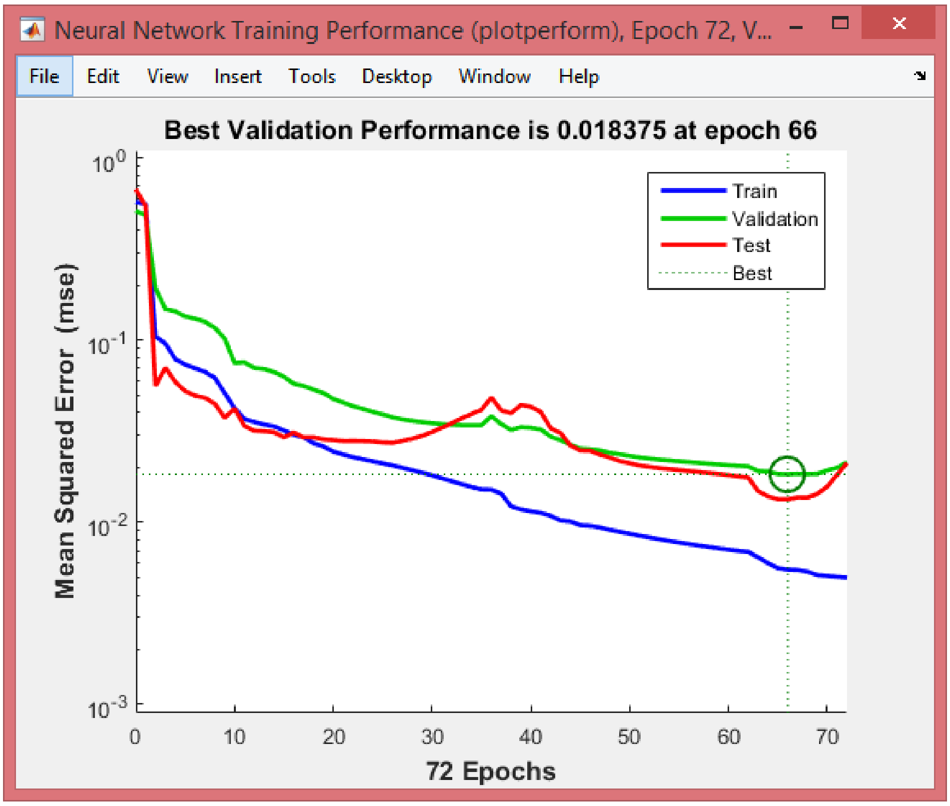


Figure 4.2h : The Performance of the training

The training of Neural Network error histogram has been plotted as the Figure below. Based on the figure, the training of the data of ECG Sensor are about 290 data. The training data from all of the data is 203 data while the testing and validation data are both 87 data simultaneously. Besides, the zero error of the data is at -0.02355 errors. The inserted data are quite small amount of data so that no all of the epoch can read the data. Only certain epoch that can read the data like at epochs -0.1433, -0.08342, -0.02355, 0.036330.0962, 0.1561 and 0.1561. Figure 4.2i show the error histogram of the training.

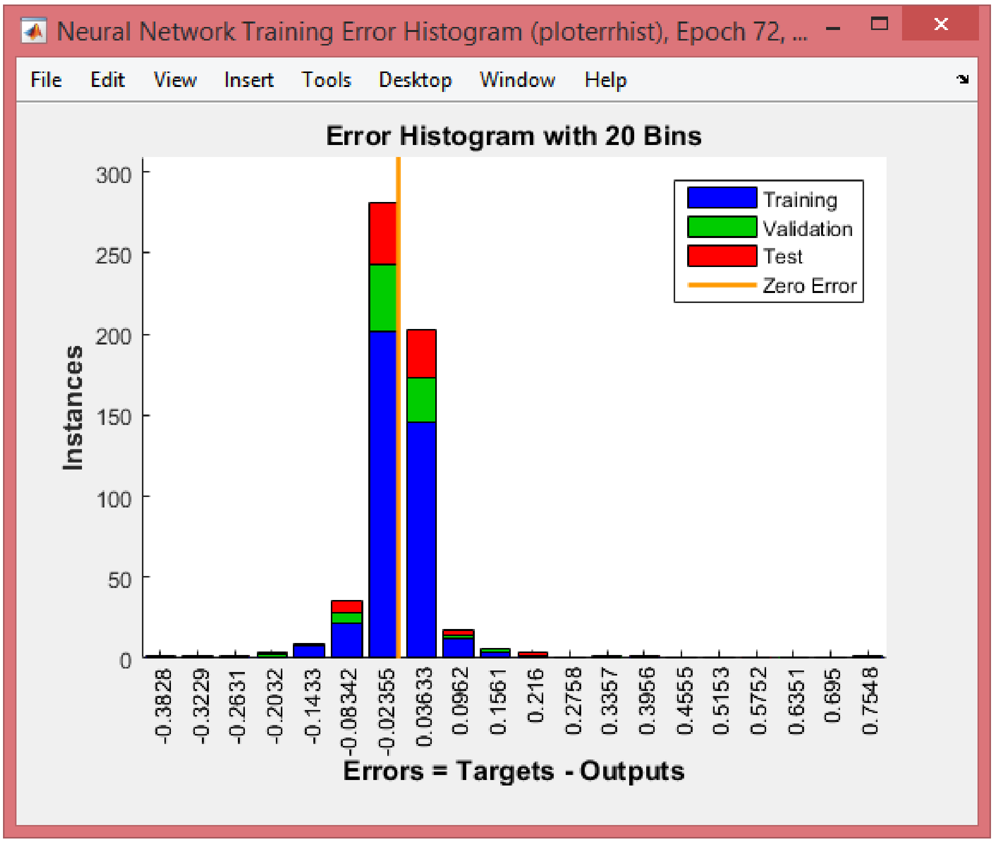


Figure 4.2i : The Neural Network Error Histogram

Figure 4.2j shows the response of the output elements 1 for Time Series 1 where there time from 0 until 300 seconds while output and targets from -0.3 until 1.1 targets. From the figure below, between times 75 seconds until 100 seconds, the response of the output and targets are the most high. However, between those responses, there is also error form from the data because the data are inserted on my own according to the condition of the pulse rate. Within the time response also, there are validation despite of the error data.

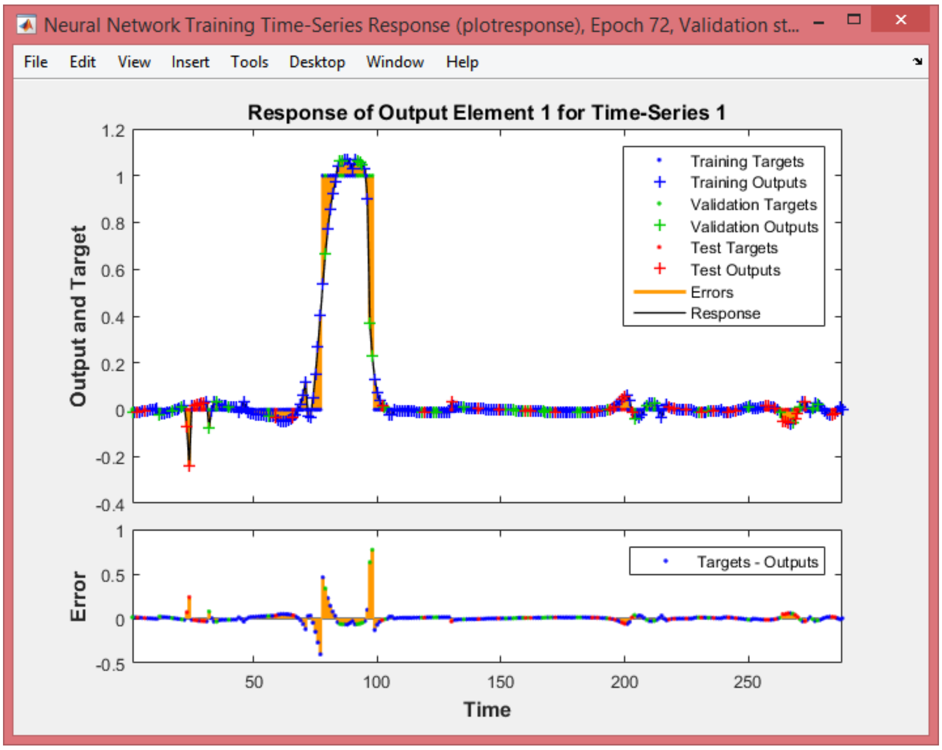


Figure 4.2j : The Neural Network Time Series Response

The regression of the training neural network of my data which is my data at the highest is 145.5 bpm. I used an artificial neural network toolbox to get a regression model to generate simulated data. I trained(70%), validated (15%), and tested(15%) it. I got reasonable result.

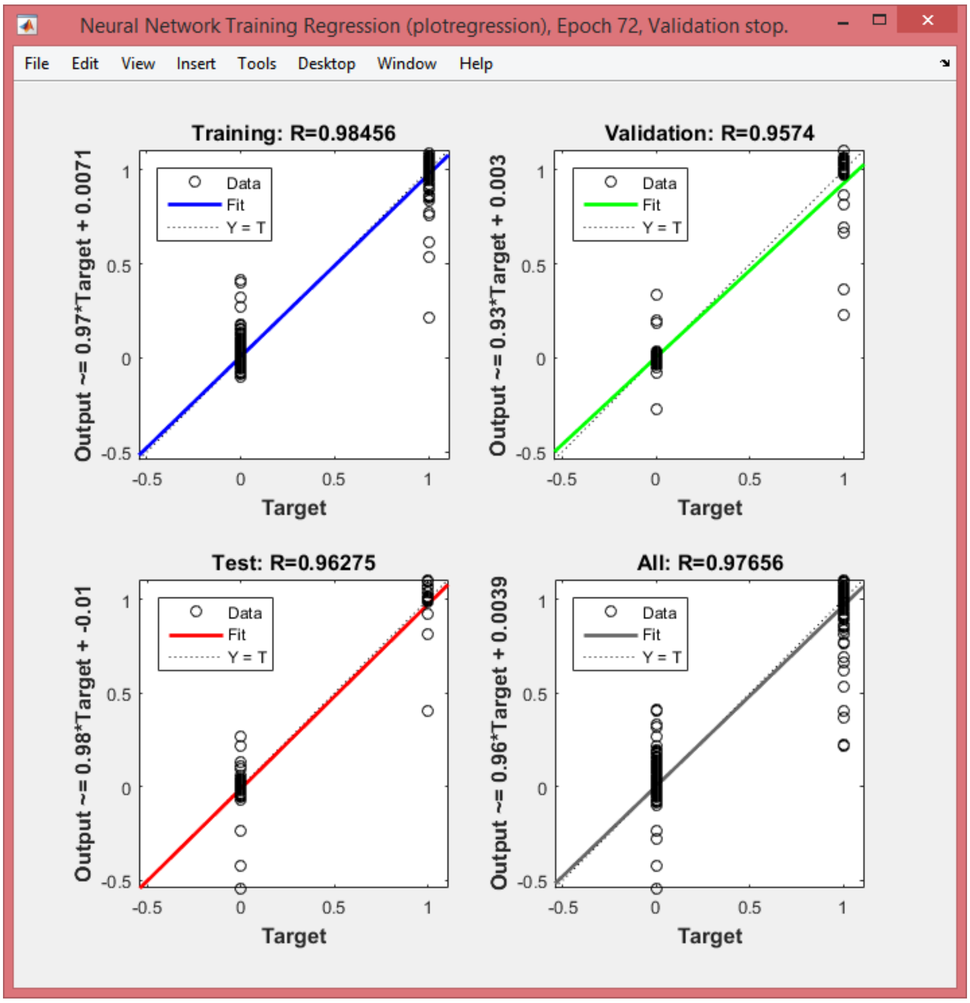


Figure 4.2k : The Neural Network Training Regression

4.2.2 ECG Sensor with Arduino Testing

In this project, the ECG Sensor has been used to detect the pulse rate in beats per minute (bpm) to analyse its rate in the pulse sensor amped visualizer. The figure below show the example of the Pulse Sensor Amped Visualizer. In that window, there are five important elements that must be show when my analysis of the pulse rate is being take place. First of all, heart beats with my pulse. This symbol and value of the heart rate is important in order to read my pulse rate whether it is in normal condition or microsleep condition. If the heart rate show value between 40bpm until 50 bpm, it is in microsleep condition.

Secondly, the pulse window with live heartbeat waveform. This element is also important because it show every heartbeat that work with pulse to display the analysis of the heartbeat waveform. Next, pulse window scale is used to adjust the scale of the heartbeat waveform. The lower the scale, the heartbeat waveform will show more of the waveform.

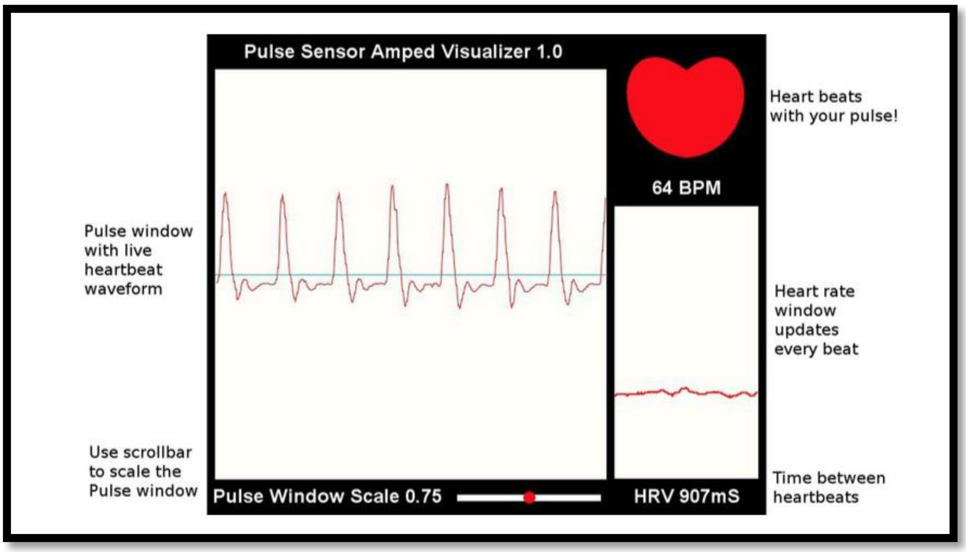


Figure 4.2l : The Pulse Sensor Amped Visualizer

In this project, for the heartbeat that read pulse rate, the maximum pulse rate that I set in Arduino IDE is 200 bpm. The 200 bpm is the state that a person do the hard core activities such as running or even playing. The normal pulse rate must be within 60 bpm until 100 bpm for the normal person both for men and woman. But if the elderly in average 70-80 years old, the pulse rate will reach 130 bpm. The pulse rate also depends to the certain condition such as if a person do a heavy activities, the pulse rate may reach until 160 bpm-200 bpm.

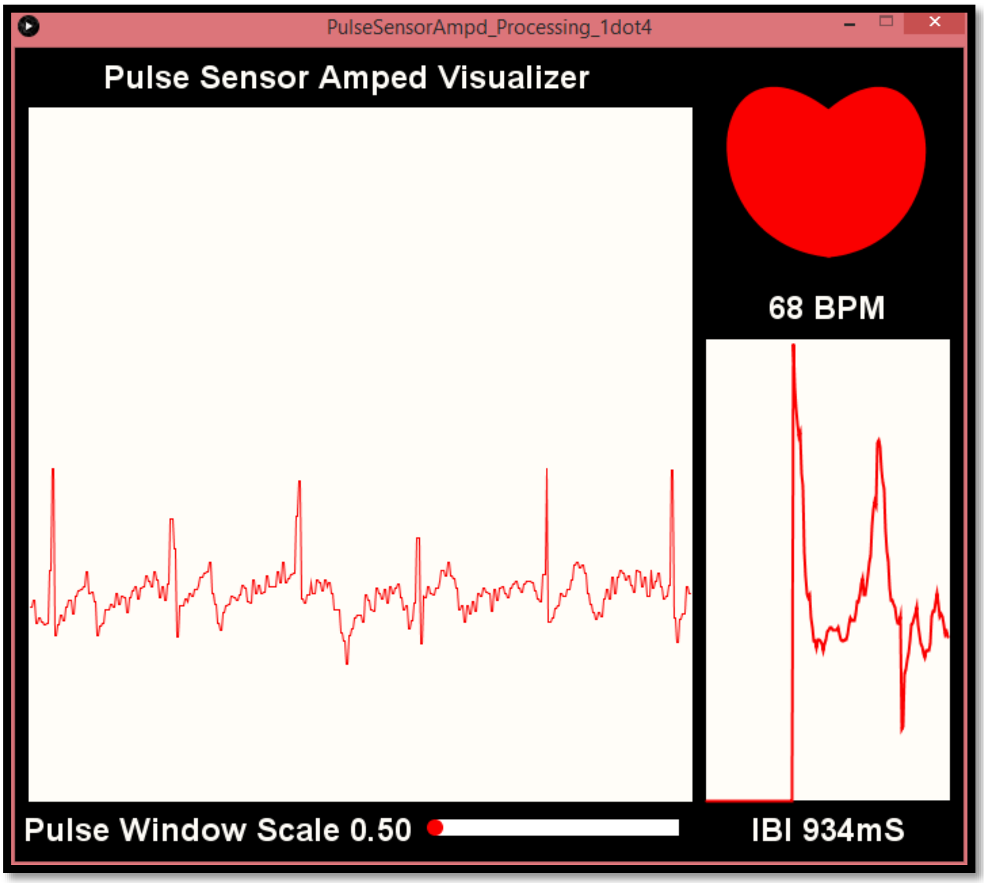


Figure 4.2m : The pulse rate of person in Normal Condition

For the microsleep, the bpm in from 40 until 50 bpm. Besides, microsleeep is depends to the condition of surroundings at that the time. There are a few parameter that will effects of the condition to get microsleep such as weather, temperature, time, age, work, surroundings and the most important things is the pulserate itself. In order to alert if the pulse rate is in the microsleep condition, the buzzer is set through this projects. The use of buzzer is to develop the microsleep system to sure that it ring when the pulse rate reaching 40 bpm until 50 bpm.

The Figure 4.2n below show the addition coding of the buzzer in Arduino IDE. While the Figure 4.2o shows the pulserate that reach 50 bpm which is microsleep condition.

```
                if (BPM >=40&&BPM<=50){
                 digitalWrite(ringBuzzer,HIGH);
                }
                else  {
                 digitalWrite(ringBuzzer,0);
}
```

Figure 4.2n : The coding for the buzzer to ring

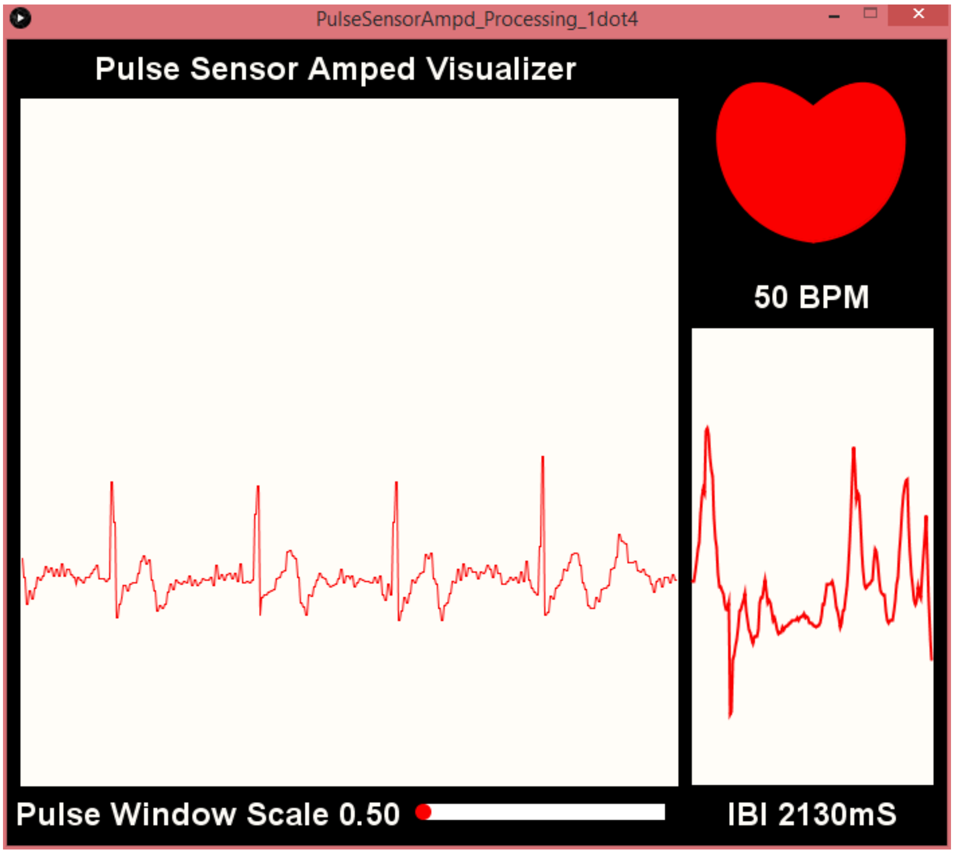


Figure 4.2o : The pulse rate of person in Microsleep Condition

CHAPTER 5
CONCLUSION & FUTURE WORK

5.0 Introduction

This chapter will conclude the overall project and followed by some recommendations on how to improve the performance of the system based on the desired results.

5.1 Conclusion

As a conclusion, the objectives of this project have been achieved. The microsleep detection system by using biosensor module which is ECG Sensor has been developed. The detection of the microsleep can be detected at the exact beats per minutes (bpm) and ifits meets the condition of the microsleep, the buzzer will ring. The buzzer will ring if the BPM of the ECG Sensor achieved the microsleep bpm which is between 40bpm until 50 bpm. Besides that, an interfacing between Arduino and MatLab also has been developed in this project to transmit data of the bpm in Arduino into MatLab. The data in the Arduino also has been trained in the MatLab by using Radial Basis Neural Network. Generally, this project provides some useful developments which have the ability to provide functionality and better performances to users. All the information in this project can be used for future research works and hopefully this can inspire our young generation engineers to design excellent innovation that meets the market needs and public interest.

5.2 Future Work

Many improvements can be done to this project to develop its accuracy, functionality and performance. Below are the suggestions for the future research and development to make this project more excellent:

(a) Add Android system to install into the phone like an alarm to give the warning.

(b) Change the Matlab implementation by using Bluetooth to interpret the data.

(c) Add a buzzer ringer to the songs to detect the microsleep.

(d) Combine the ECG Sensor with the Heart beat sensor to detect both microsleep and the blood pressure.

(f) Add the condition of the ECG Sensor to detect the other condition

REFERENCES

M. T. R. Peiris, R. D. Jones, P. R. Davidson, P. J. Bones and D. J. Myall., 2005. Fractal Dimension of the EEG for Detection of Behavioural Microsleeps. Proceedings of the 2005 IEEE, Engineering in Medicine and Biology, 27th Annual Conference, Shanghai, China, September 1-4.[1]

P. R. Davidson, R. D. Jones and M. T. R. Peiris., 2005. Detecting Behavioral Microsleeps using EEG and LSTM Recurrent Neural Networks. Proceedings of the 2005 IEEE, Engineering in Medicine and Biology 27th Annual Conference Shanghai, China, September 1-4.[2]

Malik T. R. Peiris, Richard D. Jones, Paul R. Davidson, and Philip J. Bones., 2006. Detecting Behavioral Microsleeps from EEG Power Spectra. Proceedings of the 28th IEEE, EMBS Annual International Conference New York City, USA, Aug 30-Sept 3.[3]

Wai Yie Leong, Danilo P. Mandic, Martin Golz and David Sommer., 2007. Blind Extraction Of Microsleep Events. Proc. of the 2007 15th Intl. Conf. on Digital Signal Processing (DSP).[4]

Govinda R. Poudel, Richard D. Jones, Carrie R. H. Innes, Richard Watts, T. Leigh Signal and Philip J. Bones., 2009. fMRI Correlates Of Behavioural Microsleeps During A Continuous Visuomotor Task.[5]

Carrie R. H. Innes, Govinda R. Poudel, T. Leigh Signal, and Richard D. Jones., 2010. Behavioural microsleeps in normally-rested people. 32nd Annual International Conference of the IEEE EMBS, Buenos Aires, Argentina, August 31 - September 4.[6]

M. Golz and D. Sommer., 2010. Monitoring of Drowsiness and Microsleep. 32nd Annual International Conference of the IEEE EMBS, Buenos Aires, Argentina, August 31 - September 4.[7]

Amol M. Malla, Paul R. Davidson, Philip J. Bones, Richard Green and Richard D. Jones., 2010. Automated Video-based Measurement of Eye Closure for Detecting Behavioral Microsleep. 32nd Annual International Conference of the IEEE EMBS, Buenos Aires, Argentina, August 31 - September 4.[8]

M. Holub, M. Šrutová, L. Lhotská., 2015. Adaptable Microsleep Detection Based On Eog Signals: A Feasibility Study. IEEE Dept. of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic.[9]

Sudhanshu S. D. P. Ayyagari, Richard D. Jones and Stephen J. Weddell, 2015. Optimized Echo State Networks with Leaky Integrator Neurons for EEG-Based Microsleep Detection.[10]

Broomhead, David H.; Lowe, David (1988). "Multivariable Functional Interpolation and Adaptive Networks" (PDF). Complex Systems 2: 321–355. Archived from the original (PDF) on 2014-07-14.

Poudel, G. R., Innes, C. R., Bones, P. J., Watts, R., & Jones, R. D. (2012). Losing the struggle to stay awake: Divergent thalamic and cortical activity during microsleeps. Human Brain Mapping: 00:000-000.

Paul, Amit; Linda Ng Boyle; Jon Tippin; Matthew Rizzo (2005). "Variability of driving performance during microsleeps" (PDF). Proceedings of the Third International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design. Retrieved 2008-02-10.

https://en.wikipedia.org/wiki/Microsleep

https://en.wikipedia.org/wiki/Radial_basis_function

https://en.wikipedia.org/wiki/Arduino

APPENDICES

APPENDICES A

ARDUINO UNO CODING

1) Coding For ECG Sensor to Detect Microsleep

```
#include <Bridge.h>

#include <BridgeClient.h>

#include <BridgeServer.h>

#include <BridgeUdp.h>

#include <Console.h>

#include <FileIO.h>

#include <HttpClient.h>

#include <Mailbox.h>

#include <Process.h>

#include <YunClient.h>

#include <YunServer.h>


//  VARIABLES

int pulsePin = 0;                 // Pulse Sensor purple wire connected to analog pin 0

int blinkPin = 13;                // pin to blink led at each beat

int ringBuzzer=9;

int fadePin = 5;                  // pin to do fancy classy fading blink at each beat

int fadeRate = 0;                 // used to fade LED on with PWM on fadePin



// these variables are volatile because they are used during the interrupt service
routine!

volatile int BPM;                   // used to hold the pulse rate
```

```
volatile int Signal;                // holds the incoming raw data

volatile int IBI = 600;             // holds the time between beats, must be seeded!

volatile boolean Pulse = false;     // true when pulse wave is high, false when it's
low

volatile boolean QS = false;        // becomes true when Arduoino finds a beat.

static boolean serialVisual = true;   // Re-set to 'true' to do ASCII Visual Pulse  : )



void setup(){

  pinMode(blinkPin,OUTPUT);         // pin that will blink to your heartbeat!

  pinMode(fadePin,OUTPUT);          // pin that will fade to your heartbeat!

  pinMode(ringBuzzer,OUTPUT);

  Serial.begin(115200);             // we agree to talk fast!

  interruptSetup();                 // sets up to read Pulse Sensor signal every 2mS

  // UN-COMMENT THE NEXT LINE IF YOU ARE POWERING The Pulse
Sensor AT LOW VOLTAGE,

  // AND APPLY THAT VOLTAGE TO THE A-REF PIN

  //analogReference(EXTERNAL);


}




void loop(){

  sendDataToProcessing('S', Signal);     // send Processing the raw Pulse Sensor
data

  if (QS == true){                       // Quantified Self flag is true when arduino finds a
heartbeat

    fadeRate = 255;                      // Set 'fadeRate' Variable to 255 to fade LED with
pulse

    sendDataToProcessing('B',BPM);   // send heart rate with a 'B' prefix
```

```
    sendDataToProcessing('Q',IBI);   // send time between beats with a 'Q' prefix
    QS = false;                      // reset the Quantified Self flag for next time
  }

  ledFadeToBeat();

  delay(20);                             //  take a break
}



void ledFadeToBeat(){
  fadeRate -= 15;                         //  set LED fade value
  fadeRate = constrain(fadeRate,0,255);   //  keep LED fade value from going into
negative numbers!
  analogWrite(fadePin,fadeRate);          //  fade LED
}



void sendDataToProcessing(char symbol, int data ){
  Serial.print(symbol);                // symbol prefix tells Processing what type of data
is coming
  Serial.println(data);                // the data to send culminating in a carriage return
}
volatile int rate[10];                    // array to hold last ten IBI values
volatile unsigned long sampleCounter = 0;          // used to determine pulse timing
volatile unsigned long lastBeatTime = 0;           // used to find IBI
volatile int P =512;                      // used to find peak in pulse wave, seeded
volatile int T = 512;                     // used to find trough in pulse wave, seeded
volatile int thresh = 525;                // used to find instant moment of heart beat,
seeded
```

```
volatile int amp = 100;                   // used to hold amplitude of pulse waveform, seeded
volatile boolean firstBeat = true;        // used to seed rate array so we startup with reasonable BPM
volatile boolean secondBeat = false;      // used to seed rate array so we startup with reasonable BPM



void interruptSetup(){
  // Initializes Timer2 to throw an interrupt every 2mS.
  TCCR2A = 0x02;     // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND GO INTO CTC MODE
  TCCR2B = 0x06;     // DON'T FORCE COMPARE, 256 PRESCALER
  OCR2A = 0X7C;      // SET THE TOP OF THE COUNT TO 124 FOR 500Hz SAMPLE RATE
  TIMSK2 = 0x02;     // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2 AND OCR2A
  sei();             // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED
}



// THIS IS THE TIMER 2 INTERRUPT SERVICE ROUTINE.
// Timer 2 makes sure that we take a reading every 2 miliseconds
ISR(TIMER2_COMPA_vect){                         // triggered when Timer2 counts to 124
  cli();                                      // disable interrupts while we do this
  Signal = analogRead(pulsePin);              // read the Pulse Sensor
  sampleCounter += 2;                         // keep track of the time in mS with this variable
  int N = sampleCounter - lastBeatTime;       // monitor the time since the last beat to avoid noise
```

```
   //  find the peak and trough of the pulse wave
  if(Signal < thresh && N > (IBI/5)*3){       // avoid dichrotic noise by waiting 3/5
of last IBI
    if (Signal < T){                        // T is the trough
      T = Signal;                         // keep track of lowest point in pulse wave
    }
  }

  if(Signal > thresh && Signal > P){          // thresh condition helps avoid noise
    P = Signal;                             // P is the peak
  }                                        // keep track of highest point in pulse wave

  //  NOW IT'S TIME TO LOOK FOR THE HEART BEAT
  // signal surges up in value every time there is a pulse
  if (N > 250){                                   // avoid high frequency noise
    if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3) ){
      Pulse = true;                               // set the Pulse flag when we think there is a
pulse
      digitalWrite(blinkPin,HIGH);                // turn on pin 13 LED
      IBI = sampleCounter - lastBeatTime;         // measure time between beats in mS
      lastBeatTime = sampleCounter;               // keep track of time for next pulse

      if(secondBeat){                        // if this is the second beat, if secondBeat ==
TRUE
        secondBeat = false;                  // clear secondBeat flag
        for(int i=0; i<=9; i++){             // seed the running total to get a realisitic BPM
at startup
          rate[i] = IBI;
        }
      }
```

```
    if(firstBeat){                      // if it's the first time we found a beat, if firstBeat
== TRUE
      firstBeat = false;                  // clear firstBeat flag
      secondBeat = true;                  // set the second beat flag
      sei();                              // enable interrupts again
      return;                             // IBI value is unreliable so discard it
    }



    // keep a running total of the last 10 IBI values
    word runningTotal = 0;                  // clear the runningTotal variable

    for(int i=0; i<=8; i++){                // shift data in the rate array
      rate[i] = rate[i+1];                  // and drop the oldest IBI value
      runningTotal += rate[i];              // add up the 9 oldest IBI values
    }

    rate[9] = IBI;                          // add the latest IBI to the rate array
    runningTotal += rate[9];                // add the latest IBI to runningTotal
    runningTotal /= 10;                     // average the last 10 IBI values
    BPM = 60000/runningTotal;               // how many beats can fit into a minute?
that's BPM!
    QS = true;                              // set Quantified Self flag
    // QS FLAG IS NOT CLEARED INSIDE THIS ISR
   }
  }

  if (Signal < thresh && Pulse == true){   // when the values are going down, the
beat is over
   digitalWrite(blinkPin,LOW);            // turn off pin 13 LED
```

```
    Pulse = false;                        // reset the Pulse flag so we can do it again
    amp = P - T;                          // get amplitude of the pulse wave
    thresh = amp/2 + T;                   // set thresh at 50% of the amplitude
    P = thresh;                           // reset these for next time
    T = thresh;
  }

  if (N > 2500){                          // if 2.5 seconds go by without a beat
    thresh = 512;                         // set thresh default
    P = 512;                              // set P default
    T = 512;                              // set T default
    lastBeatTime = sampleCounter;         // bring the lastBeatTime up to date
    firstBeat = true;                     // set these to avoid noise
    secondBeat = false;                   // when we get the heartbeat back
  }
if (BPM >=40&&BPM<=50){
  digitalWrite(ringBuzzer,HIGH);
}
else  {
 digitalWrite(ringBuzzer,0);
}
  sei();                                  // enable interrupts when youre done!
}// end isr
```

APPENDICES B

PULSE SENSOR AMPD PROCESSING

1) Pulse Sensor Ampd Coding

a) Pulse Sensor Ampd Processing

```
/*

THIS PROGRAM WORKS WITH PulseSensorAmped_Arduino-xx ARDUINO
CODE

THE PULSE DATA WINDOW IS SCALEABLE WITH SCROLLBAR AT
BOTTOM OF SCREEN

PRESS 'S' OR 's' KEY TO SAVE A PICTURE OF THE SCREEN IN SKETCH
FOLDER (.jpg)

MADE BY JOEL MURPHY AUGUST, 2012

UPDATED BY JOEL MURPHY JULY 2016 WITH SERIAL PORT LOCATOR
TOOL


THIS CODE PROVIDED AS IS, WITH NO CLAIMS OF FUNCTIONALITY
OR EVEN IF IT WILL WORK

    WYSIWYG

*/



import processing.serial.*;

PFont font;

PFont portsFont;

Scrollbar scaleBar;


Serial port;


int Sensor;      // HOLDS PULSE SENSOR DATA FROM ARDUINO

int IBI;         // HOLDS TIME BETWEN HEARTBEATS FROM ARDUINO

int BPM;         // HOLDS HEART RATE VALUE FROM ARDUINO
```

```
int[] RawY;      // HOLDS HEARTBEAT WAVEFORM DATA BEFORE SCALING

int[] ScaledY;   // USED TO POSITION SCALED HEARTBEAT WAVEFORM

int[] rate;      // USED TO POSITION BPM DATA WAVEFORM

float zoom;      // USED WHEN SCALING PULSE WAVEFORM TO PULSE WINDOW

float offset;    // USED WHEN SCALING PULSE WAVEFORM TO PULSE WINDOW

color eggshell = color(255, 253, 248);

int heart = 0;   // This variable times the heart image 'pulse' on screen

//  THESE VARIABLES DETERMINE THE SIZE OF THE DATA WINDOWS

int PulseWindowWidth = 490;

int PulseWindowHeight = 512;

int BPMWindowWidth = 180;

int BPMWindowHeight = 340;

boolean beat = false;    // set when a heart beat is detected, then cleared when the BPM graph is advanced


// SERIAL PORT STUFF TO HELP YOU FIND THE CORRECT SERIAL PORT

String serialPort;

String[] serialPorts = new String[Serial.list().length];

boolean serialPortFound = false;

Radio[] button = new Radio[Serial.list().length];



void setup() {

  size(700, 600);  // Stage size

  frameRate(100);

  font = loadFont("Arial-BoldMT-24.vlw");

  textFont(font);

  textAlign(CENTER);
```

```
  rectMode(CENTER);
  ellipseMode(CENTER);
// Scrollbar constructor inputs: x,y,width,height,minVal,maxVal
  scaleBar = new Scrollbar (400, 575, 180, 12, 0.5, 1.0);  // set parameters for the scale bar
  RawY = new int[PulseWindowWidth];          // initialize raw pulse waveform array
  ScaledY = new int[PulseWindowWidth];       // initialize scaled pulse waveform array
  rate = new int [BPMWindowWidth];           // initialize BPM waveform array
  zoom = 0.75;                               // initialize scale of heartbeat window


// set the visualizer lines to 0
 for (int i=0; i<rate.length; i++){
    rate[i] = 555;      // Place BPM graph line at bottom of BPM Window
   }
 for (int i=0; i<RawY.length; i++){
    RawY[i] = height/2; // initialize the pulse window data line to V/2
 }


 background(0);
 noStroke();
 // DRAW OUT THE PULSE WINDOW AND BPM WINDOW RECTANGLES
 drawDataWindows();
 drawHeart();


// GO FIND THE ARDUINO
  fill(eggshell);
  text("Select Your Serial Port",245,30);
  listAvailablePorts();
```

```
}

void draw() {
if(serialPortFound){
  // ONLY RUN THE VISUALIZER AFTER THE PORT IS CONNECTED
  background(0);
  noStroke();
  drawDataWindows();
  drawPulseWaveform();
  drawBPMwaveform();
  drawHeart();
// PRINT THE DATA AND VARIABLE VALUES
  fill(eggshell);                                  // get ready to print text
  text("Pulse Sensor Amped Visualizer",245,30);     // tell them what you are
  text("IBI " + IBI + "mS",600,585);                 // print the time between
heartbeats in mS
  text(BPM + " BPM",600,200);                        // print the Beats Per Minute
  text("Pulse Window Scale " + nf(zoom,1,2), 150, 585); // show the current scale
of Pulse Window

//  DO THE SCROLLBAR THINGS
  scaleBar.update (mouseX, mouseY);
  scaleBar.display();

} else { // SCAN BUTTONS TO FIND THE SERIAL PORT

  for(int i=0; i<button.length; i++){
    button[i].overRadio(mouseX,mouseY);
    button[i].displayRadio();
```

```
  }


}


}  //end of draw loop



void drawDataWindows(){
    // DRAW OUT THE PULSE WINDOW AND BPM WINDOW
RECTANGLES
    fill(eggshell);  // color for the window background
    rect(255,height/2,PulseWindowWidth,PulseWindowHeight);
    rect(600,385,BPMWindowWidth,BPMWindowHeight);
}


void drawPulseWaveform(){
  // DRAW THE PULSE WAVEFORM
  // prepare pulse data points
  RawY[RawY.length-1] = (1023 - Sensor) - 212;   // place the new raw datapoint
at the end of the array
  zoom = scaleBar.getPos();                   // get current waveform scale value
  offset = map(zoom,0.5,1,150,0);               // calculate the offset needed at this
scale
  for (int i = 0; i < RawY.length-1; i++) {      // move the pulse waveform by
    RawY[i] = RawY[i+1];                        // shifting all raw datapoints one pixel
left
    float dummy = RawY[i] * zoom + offset;      // adjust the raw data to the
selected scale
    ScaledY[i] = constrain(int(dummy),44,556);   // transfer the raw data array to
the scaled array
  }
```

```
 stroke(250,0,0);                          // red is a good color for the pulse waveform
 noFill();
 beginShape();                             // using beginShape() renders fast
 for (int x = 1; x < ScaledY.length-1; x++) {
   vertex(x+10, ScaledY[x]);                    //draw a line connecting the data points
 }
 endShape();
}


void drawBPMwaveform(){
// DRAW THE BPM WAVE FORM
// first, shift the BPM waveform over to fit then next data point only when a beat is found
 if (beat == true){   // move the heart rate line over one pixel every time the heart beats
   beat = false;      // clear beat flag (beat flag waset in serialEvent tab)
   for (int i=0; i<rate.length-1; i++){
     rate[i] = rate[i+1];                  // shift the bpm Y coordinates over one pixel to the left
   }
// then limit and scale the BPM value
   BPM = min(BPM,200);                     // limit the highest BPM value to 200
   float dummy = map(BPM,0,200,555,215);   // map it to the heart rate window Y
   rate[rate.length-1] = int(dummy);       // set the rightmost pixel to the new data point value
 }
 // GRAPH THE HEART RATE WAVEFORM
 stroke(250,0,0);                          // color of heart rate graph
 strokeWeight(2);                          // thicker line is easier to read
 noFill();
 beginShape();
```

```
 for (int i=0; i < rate.length-1; i++){    // variable 'i' will take the place of pixel x
position
   vertex(i+510, rate[i]);                 // display history of heart rate datapoints
 }
 endShape();
}

void drawHeart(){
  // DRAW THE HEART AND MAYBE MAKE IT BEAT
    fill(250,0,0);
    stroke(250,0,0);
    // the 'heart' variable is set in serialEvent when arduino sees a beat happen
    heart--;                    // heart is used to time how long the heart graphic swells
when your heart beats
    heart = max(heart,0);       // don't let the heart variable go into negative numbers
    if (heart > 0){             // if a beat happened recently,
      strokeWeight(8);          // make the heart big
    }
    smooth();   // draw the heart with two bezier curves
    bezier(width-100,50, width-20,-20, width,140, width-100,150);
    bezier(width-100,50, width-190,-20, width-200,140, width-100,150);
    strokeWeight(1);          // reset the strokeWeight for next time
}

void listAvailablePorts(){
  println(Serial.list());    // print a list of available serial ports to the console
  serialPorts = Serial.list();
  fill(0);
  textFont(font,16);
  textAlign(LEFT);
```

```
  // set a counter to list the ports backwards
  int yPos = 0;
  for(int i=serialPorts.length-1; i>=0; i--){
    button[i] = new Radio(35,
95+(yPos*20),12,color(180),color(80),color(255),i,button);
    text(serialPorts[i],50, 100+(yPos*20));
    yPos++;
  }
  textFont(font);
  textAlign(CENTER);
}
```

b) Keyboard Mouse

```
void mousePressed(){
  scaleBar.press(mouseX, mouseY);
  if(!serialPortFound){
    for(int i=0; i<button.length; i++){
      if(button[i].pressRadio(mouseX,mouseY)){
        try{
          port = new Serial(this, Serial.list()[i], 115200);  // make sure Arduino is
talking serial at this baud rate
          delay(1000);
          println(port.read());
          port.clear();            // flush buffer
          port.bufferUntil('\n');  // set buffer full flag on receipt of carriage return
          serialPortFound = true;
        }
        catch(Exception e){
          println("Couldn't open port " + Serial.list()[i]);
```

```
      }
    }
  }
 }
}

void mouseReleased(){
  scaleBar.release();
}

void keyPressed(){

 switch(key){
   case 's':    // pressing 's' or 'S' will take a jpg of the processing window
   case 'S':
     saveFrame("heartLight-####.jpg");    // take a shot of that!
     break;

   default:
     break;
 }
}
```

c) Radio Buttons

```
class Radio {
  int _x,_y;
  int size, dotSize;
  color baseColor, overColor, pressedColor;
  boolean over, pressed;
  int me;
```

```
Radio[] radios;

Radio(int xp, int yp, int s, color b, color o, color p, int m, Radio[] r) {
  _x = xp;
  _y = yp;
  size = s;
  dotSize = size - size/3;
  baseColor = b;
  overColor = o;
  pressedColor = p;
  radios = r;
  me = m;
}

boolean pressRadio(float mx, float my){
  if (dist(_x, _y, mx, my) < size/2){
    pressed = true;
    for(int i=0; i<radios.length; i++){
      if(i != me){ radios[i].pressed = false; }
    }
    return true;
  } else {
    return false;
  }
}

boolean overRadio(float mx, float my){
  if (dist(_x, _y, mx, my) < size/2){
    over = true;
    for(int i=0; i<radios.length; i++){
      if(i != me){ radios[i].over = false; }
    }
    return true;
  } else {
    return false;
  }
}

void displayRadio(){
  noStroke();
  fill(baseColor);
  ellipse(_x,_y,size,size);
```

```
    if(over){
      fill(overColor);
      ellipse(_x,_y,dotSize,dotSize);
    }
    if(pressed){
      fill(pressedColor);
      ellipse(_x,_y,dotSize,dotSize);
    }
  }
}
```

APPENDICES C

TABLE OF DATA IN MATLAB AND SCHEMATIC DIAGRAM

1) Microsleep Data in Matlab

Pulse rate	Weather	Tempe-rature	Surround-ings	Gender	Body Condition	Time	Age	Work
0.5	0.5	-20	-20	0	1	6	-20	0
1	1	-19.5	-19.5	0	1	6.05	-19.5	0
1.5	1.5	-19	-19	0	1	6.1	-19	0
2	2	-18.5	-18.5	0	1	6.15	-18.5	0
2.5	2.5	-18	-18	0	1	6.2	-18	0
3	3	-17.5	-17.5	0	1	6.25	-17.5	0
3.5	3.5	-17	-17	0	1	6.3	-17	0
4	4	-16.5	-16.5	0	1	6.35	-16.5	0
4.5	4.5	-16	-16	0	1	6.4	-16	0
5	5	-15.5	-15.5	0	1	6.45	-15.5	0
5.5	5.5	-15	-15	0	1	6.5	-15	0
6	6	-14.5	-14.5	0	1	6.55	-14.5	0
6.5	6.5	-14	-14	0	1	7	-14	0
7	7	-13.5	-13.5	0	1	7.05	-13.5	0
7.5	7.5	-13	-13	0	1	7.1	-13	0
8	8	-12.5	-12.5	0	1	7.15	-12.5	0
8.5	8.5	-12	-12	0	1	7.2	-12	0
9	9	-11.5	-11.5	0	1	7.25	-11.5	0
9.5	9.5	-11	-11	0	1	7.3	-11	0
10	10	-10.5	-10.5	0	1	7.35	-10.5	0
10.5	10.5	-10	-10	0	1	7.4	-10	0
11	11	-9.5	-9.5	0	1	7.45	-9.5	0
11.5	11.5	-9	-9	0	1	7.5	-9	0
12	12	-8.5	-8.5	0	1	7.55	-8.5	0
12.5	12.5	-8	-8	0	1	8	-8	0
13	13	-7.5	-7.5	0	1	8.05	-7.5	0
13.5	13.5	-7	-7	0	1	8.1	-7	0
14	14	-6.5	-6.5	0	1	8.15	-6.5	0
14.5	14.5	-6	-6	0	1	8.2	-6	0
15	15	-5.5	-5.5	0	1	8.25	-5.5	0
15.5	15.5	-5	-5	0	1	8.3	-5	0
16	16	-4.5	-4.5	0	1	8.35	-4.5	1
16.5	16.5	-4	-4	0	1	8.4	-4	1
17	17	-3.5	-3.5	0	1	8.45	-3.5	1
17.5	17.5	-3	-3	0	1	8.5	-3	1
18	18	-2.5	-2.5	0	1	8.55	-2.5	1

18.5	18.5	-2	-2	0	1	9	-2	1
19	19	-1.5	-1.5	0	1	9.05	-1.5	1
19.5	19.5	-1	-1	0	1	9.1	-1	1
20	20	-0.5	-0.5	0	1	9.15	-0.5	1
20.5	20.5	0	0	0	1	9.2	0	1
21	21	0.5	0.5	0	1	9.25	0.5	1
21.5	21.5	1	1	0	1	9.3	1	1
22	22	1.5	1.5	0	1	9.35	1.5	1
22.5	22.5	2	2	0	1	9.4	2	1
23	23	2.5	2.5	0	1	9.45	2.5	1
23.5	23.5	3	3	0	1	9.5	3	1
24	24	3.5	3.5	0	1	9.55	3.5	1
24.5	24.5	4	4	0	1	10	4	1
25	25	4.5	4.5	0	1	10.05	4.5	1
25.5	25.5	5	5	0	1	10.1	5	1
26	26	5.5	5.5	0	1	10.15	5.5	1
26.5	26.5	6	6	0	1	10.2	6	1
27	27	6.5	6.5	0	1	10.25	6.5	1
27.5	27.5	7	7	0	1	10.3	7	1
28	28	7.5	7.5	0	1	10.35	7.5	1
28.5	28.5	8	8	0	1	10.4	8	1
29	29	8.5	8.5	0	1	10.45	8.5	1
29.5	29.5	9	9	0	1	10.5	9	1
30	30	9.5	9.5	0	1	10.55	9.5	1
30.5	30.5	10	10	0	1	11	10	1
31	31	10.5	10.5	0	1	11.05	10.5	1
31.5	31.5	11	11	0	1	11.1	11	1
32	32	11.5	11.5	0	1	11.15	11.5	1
32.5	32.5	12	12	0	1	11.2	12	1
33	33	12.5	12.5	0	1	11.25	12.5	1
33.5	33.5	13	13	0	1	11.3	13	1
34	34	13.5	13.5	0	1	11.35	13.5	1
34.5	34.5	14	14	0	1	11.4	14	1
35	35	14.5	14.5	0	1	11.45	14.5	1
35.5	35.5	15	15	0	1	11.5	15	1
36	36	15.5	15.5	0	1	11.55	15.5	1
36.5	36.5	16	16	1	1	12	16	1
37	37	16.5	16.5	1	1	12.05	16.5	1
37.5	37.5	17	17	1	1	12.1	17	1
38	38	17.5	17.5	1	1	12.15	17.5	1
38.5	38.5	18	18	1	1	12.2	18	1
39	39	18.5	18.5	1	1	12.25	18.5	1
39.5	39.5	19	19	1	1	12.3	19	1
40	40	19.5	19.5	1	1	12.35	19.5	1

40.5	40.5	20	20	1	1	12.4	20	1
41	41	20.5	20.5	1	1	12.45	20.5	1
41.5	41.5	21	21	1	1	12.5	21	1
42	42	21.5	21.5	1	1	12.55	21.5	1
42.5	42.5	22	22	1	1	13	22	1
43	43	22.5	22.5	1	1	13.05	22.5	1
43.5	43.5	23	23	1	1	13.1	23	1
44	44	23.5	23.5	1	1	13.15	23.5	1
44.5	44.5	24	24	1	1	13.2	24	1
45	45	24.5	24.5	1	1	13.25	24.5	0
45.5	45.5	25	25	1	1	13.3	25	0
46	46	25.5	25.5	1	1	13.35	25.5	0
46.5	46.5	26	26	1	1	13.4	26	0
47	47	26.5	26.5	1	1	13.45	26.5	0
47.5	47.5	27	27	1	1	13.5	27	0
48	48	27.5	27.5	1	1	13.55	27.5	0
48.5	48.5	28	28	1	1	14	28	0
49	49	28.5	28.5	1	1	14.05	28.5	0
49.5	49.5	29	29	1	1	14.1	29	0
50	50	29.5	29.5	1	1	14.15	29.5	0
50.5	50.5	30	30	1	1	14.2	30	0
51	51	30.5	30.5	1	1	14.25	30.5	0
51.5	51.5	31	31	1	1	14.3	31	0
52	52	31.5	31.5	1	1	14.35	31.5	0
52.5	52.5	32	32	1	1	14.4	32	0
53	53	32.5	32.5	1	1	14.45	32.5	0
53.5	53.5	33	33	1	1	14.5	33	0
54	54	33.5	33.5	1	1	14.55	33.5	0
54.5	54.5	34	34	1	1	15	34	0
55	55	34.5	34.5	1	1	15.05	34.5	0
55.5	55.5	35	35	1	1	15.1	35	0
56	56	35.5	35.5	1	1	15.15	35.5	0
56.5	56.5	36	36	1	1	15.2	36	0
57	57	36.5	36.5	1	1	15.25	36.5	0
57.5	57.5	37	37	1	1	15.3	37	0
58	58	37.5	37.5	1	1	15.35	37.5	0
58.5	58.5	38	38	1	1	15.4	38	0
59	59	38.5	38.5	1	1	15.45	38.5	0
59.5	59.5	39	39	1	1	15.5	39	0
60	60	39.5	39.5	1	1	15.55	39.5	0
60.5	60.5	40	40	1	1	16	40	0
61	61	40.5	40.5	1	1	16.05	40.5	0
61.5	61.5	41	41	1	1	16.1	41	0
62	62	41.5	41.5	1	1	16.15	41.5	0

62.5	62.5	42	42	1	1	16.2	42	0
63	63	42.5	42.5	1	1	16.25	42.5	0
63.5	63.5	43	43	1	1	16.3	43	0
64	64	43.5	43.5	1	1	16.35	43.5	0
64.5	64.5	44	44	1	1	16.4	44	0
65	65	44.5	44.5	1	1	16.45	44.5	0
65.5	65.5	45	45	0	1	16.5	45	0
66	66	45.5	45.5	0	1	16.55	45.5	0
66.5	66.5	46	46	0	1	17	46	0
67	67	46.5	46.5	0	1	17.05	46.5	0
67.5	67.5	47	47	0	1	17.1	47	0
68	68	47.5	47.5	0	1	17.15	47.5	0
68.5	68.5	48	48	0	1	17.2	48	0
69	69	48.5	48.5	0	1	17.25	48.5	0
69.5	69.5	49	49	0	1	17.3	49	0
70	70	49.5	49.5	0	1	17.35	49.5	0
70.5	70.5	50	50	0	1	17.4	50	0
71	71	50.5	50.5	0	1	17.45	50.5	0
71.5	71.5	51	51	0	1	17.5	51	0
72	72	51.5	51.5	0	1	17.55	51.5	0
72.5	72.5	52	52	0	1	18	52	0
73	73	52.5	52.5	0	1	18.05	52.5	0
73.5	73.5	53	53	0	1	18.1	53	0
74	74	53.5	53.5	0	1	18.15	53.5	0
74.5	74.5	54	54	0	1	18.2	54	0
75	75	54.5	54.5	0	1	18.25	54.5	0
75.5	75.5	55	55	0	1	18.3	55	0
76	76	55.5	55.5	0	1	18.35	55.5	0
76.5	76.5	56	56	0	1	18.4	56	0
77	77	56.5	56.5	0	1	18.45	56.5	0
77.5	77.5	57	57	0	1	18.5	57	0
78	78	57.5	57.5	0	1	18.55	57.5	0
78.5	78.5	58	58	0	1	19	58	0
79	79	58.5	58.5	0	1	19.05	58.5	0
79.5	79.5	59	59	0	1	19.1	59	0
80	80	59.5	59.5	0	1	19.15	59.5	0
80.5	80.5	60	60	0	1	19.2	60	0
81	81	60.5	60.5	0	1	19.25	60.5	0
81.5	81.5	61	61	0	1	19.3	61	0
82	82	61.5	61.5	0	1	19.35	61.5	0
82.5	82.5	62	62	0	1	19.4	62	0
83	83	62.5	62.5	0	1	19.45	62.5	0
83.5	83.5	63	63	0	1	19.5	63	0
84	84	63.5	63.5	0	1	19.55	63.5	0

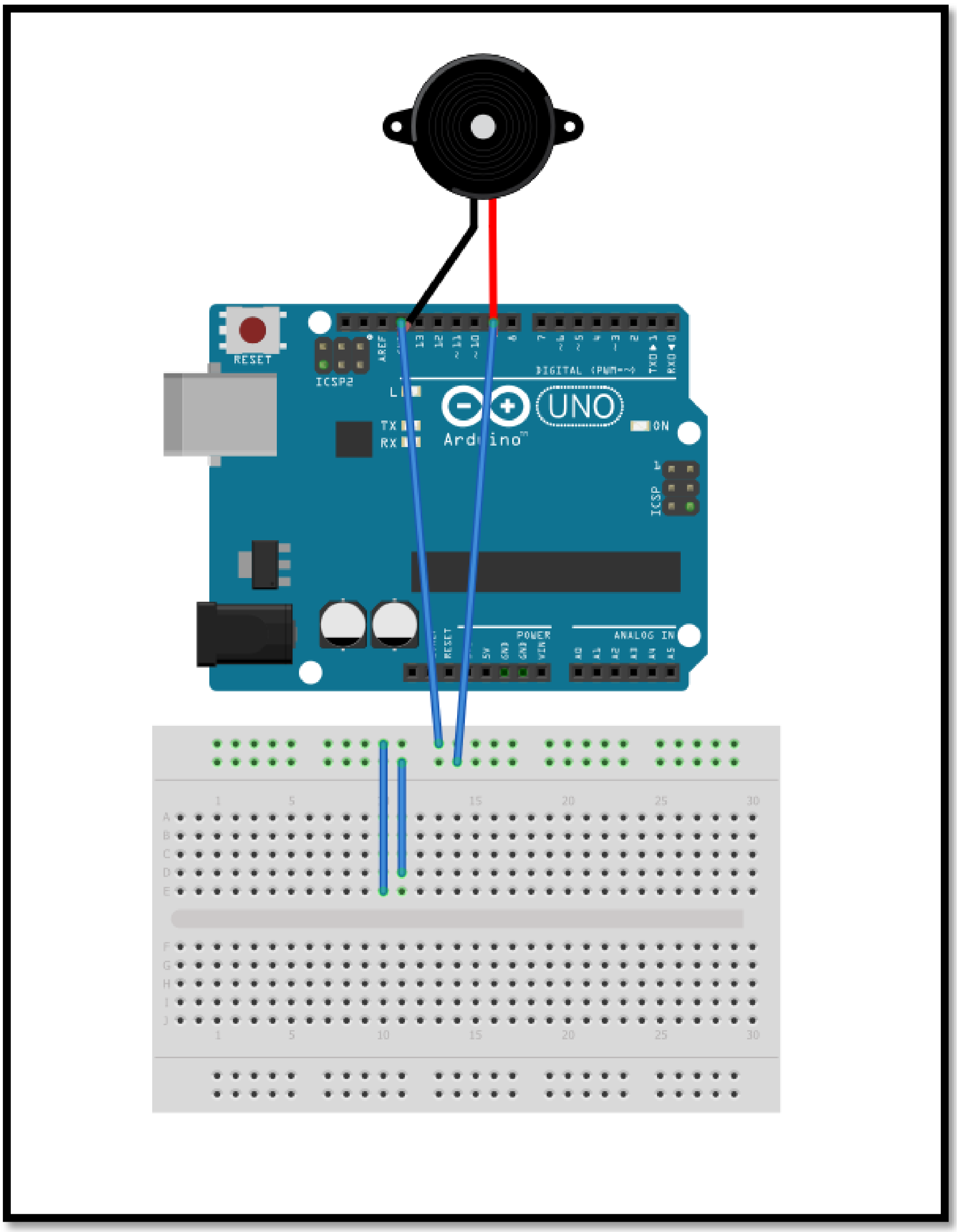
84.5	84.5	64	64	0	1	20	64	0
85	85	64.5	64.5	0	1	20.05	64.5	0
85.5	85.5	65	65	0	1	20.1	65	0
86	86	65.5	65.5	0	1	20.15	65.5	0
86.5	86.5	66	66	0	1	20.2	66	0
87	87	66.5	66.5	0	1	20.25	66.5	0
87.5	87.5	67	67	0	1	20.3	67	0
88	88	67.5	67.5	0	1	20.35	67.5	0
88.5	88.5	68	68	0	1	20.4	68	0
89	89	68.5	68.5	0	1	20.45	68.5	0
89.5	89.5	69	69	0	1	20.5	69	0
90	90	69.5	69.5	0	1	20.55	69.5	0
90.5	90.5	70	70	0	1	21	70	0
91	91	70.5	70.5	0	1	21.05	70.5	0
91.5	91.5	71	71	0	1	21.1	71	0
92	92	71.5	71.5	0	1	21.15	71.5	0
92.5	92.5	72	72	0	1	21.2	72	0
93	93	72.5	72.5	0	1	21.25	72.5	0
93.5	93.5	73	73	0	1	21.3	73	0
94	94	73.5	73.5	0	1	21.35	73.5	0
94.5	94.5	74	74	0	1	21.4	74	0
95	95	74.5	74.5	0	1	21.45	74.5	0
95.5	95.5	75	75	0	1	21.5	75	0
96	96	75.5	75.5	0	1	21.55	75.5	0
96.5	96.5	76	76	0	1	22	76	0
97	97	76.5	76.5	0	1	22.05	76.5	0
97.5	97.5	77	77	0	1	22.1	77	0
98	98	77.5	77.5	0	1	22.15	77.5	0
98.5	98.5	78	78	0	1	22.2	78	0
99	99	78.5	78.5	0	1	22.25	78.5	0
99.5	99.5	79	79	0	1	22.3	79	0
100	100	79.5	79.5	0	1	22.35	79.5	0
100.5	100.5	80	80	0	1	22.4	80	0
101	101	80.5	80.5	0	1	22.45	80.5	0
101.5	101.5	81	81	0	1	22.5	81	0
102	102	81.5	81.5	0	1	22.55	81.5	0
102.5	102.5	82	82	0	1	23	82	0
103	103	82.5	82.5	0	1	23.05	82.5	0
103.5	103.5	83	83	0	1	23.1	83	0
104	104	83.5	83.5	0	1	23.15	83.5	0
104.5	104.5	84	84	0	1	23.2	84	0
105	105	84.5	84.5	0	1	23.25	84.5	0
105.5	105.5	85	85	0	1	23.3	85	0
106	106	85.5	85.5	0	1	23.35	85.5	0

106.5	106.5	86	86	0	1	23.4	86	0
107	107	86.5	86.5	0	1	23.45	86.5	0
107.5	107.5	87	87	0	1	23.5	87	0
108	108	87.5	87.5	0	1	23.55	87.5	0
108.5	108.5	88	88	0	1	0	88	0
109	109	88.5	88.5	0	1	0.05	88.5	0
109.5	109.5	89	89	0	1	0.1	89	0
110	110	89.5	89.5	0	1	0.15	89.5	0
110.5	110.5	90	90	0	1	0.2	90	0
111	111	90.5	90.5	0	1	0.25	90.5	0
111.5	111.5	91	91	0	1	0.3	91	0
112	112	91.5	91.5	0	1	0.35	91.5	0
112.5	112.5	92	92	0	1	0.4	92	0
113	113	92.5	92.5	0	1	0.45	92.5	0
113.5	113.5	93	93	0	1	0.5	93	0
114	114	93.5	93.5	0	1	0.55	93.5	0
114.5	114.5	94	94	0	1	1	94	0
115	115	94.5	94.5	0	1	1.05	94.5	0
115.5	115.5	95	95	0	1	1.1	95	0
116	116	95.5	95.5	0	1	1.15	95.5	0
116.5	116.5	96	96	0	1	1.2	96	0
117	117	96.5	96.5	0	1	1.25	96.5	0
117.5	117.5	97	97	0	1	1.3	97	0
118	118	97.5	97.5	0	1	1.35	97.5	0
118.5	118.5	98	98	0	1	1.4	98	0
119	119	98.5	98.5	0	1	1.45	98.5	0
119.5	119.5	99	99	0	1	1.5	99	0
120	120	99.5	99.5	0	1	1.55	99.5	0
120.5	120.5	100	100	0	1	2	100	0
121	121	100.5	100.5	0	1	2.05	100.5	0
121.5	121.5	101	101	0	1	2.1	101	0
122	122	101.5	101.5	0	1	2.15	101.5	0
122.5	122.5	102	102	0	1	2.2	102	0
123	123	102.5	102.5	0	1	2.25	102.5	0
123.5	123.5	103	103	0	1	2.3	103	0
124	124	103.5	103.5	0	1	2.35	103.5	0
124.5	124.5	104	104	0	1	2.4	104	0
125	125	104.5	104.5	0	1	2.45	104.5	0
125.5	125.5	105	105	0	1	2.5	105	0
126	126	105.5	105.5	0	1	2.55	105.5	0
126.5	126.5	106	106	0	1	3	106	0
127	127	106.5	106.5	0	1	3.05	106.5	0
127.5	127.5	107	107	0	1	3.1	107	0
128	128	107.5	107.5	0	1	3.15	107.5	0

128.5	128.5	108	108	0	1	3.2	108	0
129	129	108.5	108.5	0	1	3.25	108.5	0
129.5	129.5	109	109	0	1	3.3	109	0
130	130	109.5	109.5	0	1	3.35	109.5	0
130.5	130.5	110	110	0	1	3.4	110	0
131	131	110.5	110.5	0	1	3.45	110.5	0
131.5	131.5	111	111	0	1	3.5	111	0
132	132	111.5	111.5	0	1	3.55	111.5	0
132.5	132.5	112	112	0	1	4	112	0
133	133	112.5	112.5	0	1	4.05	112.5	0
133.5	133.5	113	113	0	1	4.1	113	0
134	134	113.5	113.5	0	1	4.15	113.5	0
134.5	134.5	114	114	0	1	4.2	114	0
135	135	114.5	114.5	0	1	4.25	114.5	0
135.5	135.5	115	115	0	1	4.3	115	0
136	136	115.5	115.5	0	1	4.35	115.5	0
136.5	136.5	116	116	0	1	4.4	116	0
137	137	116.5	116.5	0	1	4.45	116.5	0
137.5	137.5	117	117	0	1	4.5	117	0
138	138	117.5	117.5	0	1	4.55	117.5	0
138.5	138.5	118	118	0	1	5	118	0
139	139	118.5	118.5	0	1	5.05	118.5	0
139.5	139.5	119	119	0	1	5.1	119	0
140	140	119.5	119.5	0	1	5.15	119.5	0
140.5	140.5	120	120	0	1	5.2	120	0
141	141	120.5	120.5	0	1	5.25	120.5	0
141.5	141.5	121	121	0	1	5.3	121	0
142	142	121.5	121.5	0	1	5.35	121.5	0
142.5	142.5	122	122	0	1	5.4	122	0
143	143	122.5	122.5	0	1	5.45	122.5	0
143.5	143.5	123	123	0	1	5.5	123	0
144	144	123.5	123.5	0	1	5.55	123.5	0
144.5	144.5	124	124	0	1	6	124	0
145	145	124.5	124.5	0	1	6.05	124.5	0

2) Schematic Diagram

a) Schematic Diagram connection on breadboard



b) Schematic Diagram connection the real circuit