Sheffield Hallam University

Faculty of Arts, Computing, Engineering & Sciences

Department of Computing

Final Year Individual Project (SEGM)
[55-6727] 2015/2016

Author:	Shaun Webb
Date Submitted:	11^{th} April 2016
Supervisor:	Dr Peter C. Collingwood
Degree Course:	Software Engineering BSc
Title of Project:	Real-time robotics control using Electroencephalography

Confidentially Required?

NO ☒

YES ☐

Abstract

This project report explores the research and development of a prototype application that utilises Electroencephalography to control a robot. Electrical fluctuations from the neurons in the brain are measured using a brain-computer interface and advanced event-related potential techniques are employed to understand specific cognitive actions. The application developed interprets cognitive actions and sends commands to manipulate robot movements. A test-driven development approach allowed the project to progress incrementally, producing quality concurrent software with minimal cyclomatic complexities. The application developed proves the feasibility of the objectives.

This project was motivated by a visit to Miraikan (The National Museum of Emerging Science and Innovation) in Tokyo, Japan. The potential for robots such as ASIMO to be controlled in more natural and intuitive ways was evident.

The final prototype including demonstration video can be found at:
http://shaunwebb.co.uk/eeg.html

Acknowledgements
I would like to take this opportunity to offer my sincerest gratitude to my dissertation supervisor Dr. Peter C. Collingwood for providing professional support and guidance throughout the development of this project. Furthermore, I would like to thank Citrix Systems for providing financial support which made this project possible and Dr. Peter O'Neill for providing an opportunity to establish a working partnership with Barnsley hospital. Finally, thank you to my father, Leonard Webb who has suffered from, a traumatic brain injury, dementia, and other problems but has provided me with a lifetime worth of support.

Contents

Abstract 1

Acknowledgements 2

Chapter 1 – Introduction 7

1.1 Introduction 7

1.2 Project aim 7

Chapter 2 – Background Investigation 8

2.1 Electroencephalography (EEG) 8

2.1.1 Introduction to Electroencephalography 8

2.1.2 A Brief Historical look at EEG 8

2.1.3 How it relates today 8

2.1.4 Usage of EEG 8

2.2 Brain-Computer Interfaces (BCI) 9

2.2.1 Introduction to Brain-Computer Interfaces 9

2.2.2 Advantages of BCI 9

2.2.3 Disadvantages of BCI 9

2.2.4 Real World Usage of BCI 9

2.2.5 Alternative BCI techniques 10

2.3 Robotics 11

2.3.1 Introduction to Robotics 11

2.3.2 Real World Usage of Robotics 11

2.4 Summary 12

Chapter 3 – Requirements 13

3.1 Introduction to Requirements 13

3.2 Requirements elicitation 13

3.3 Limitations 13

3.4 Summary 13

3.4.1 Client Requirements 14

3.4.2 Device Requirements 14

3.4.3 Application requirements 14

Chapter 4 – Design 14

4.1 Introduction to Design 14

4.2 Hardware 14

4.2.1 EPOC Emotiv System 14
4.2.2 Zumo Arduino Based Robot 16
4.3 Implementation Plan 16
4.3.1 SDK - Programming Language 16
4.3.2 Integrated development environment 17
4.3.3 Version Control 17
4.4 Conceptual Architecture 18
4.5 User Interface 19
4.6 Test Plan 19
4.6.1 Test-Driven Development 19
4.6.2 User Testing 20
4.7 Summary 20
Chapter 5 – Development 21
5.1 Introduction to Development 21
5.2 User Interface interactivity and functionality 21
5.2.1 Connecting to the EEG EPOC 21
5.2.2 Profile Management with Cognitive Training 22
5.2.3 EEG Data Retrieval 22
5.2.4 Data Visualization Chart 23
5.2.5 EEG Data Logging 23
5.3 Real-time Robotics control 24
5.3.1 Robot Simulator 24
5.3.2 Real Robot 25
5.4 Supporting tools 26
5.4.1 Documentation 26
5.4.2 Environment 27
5.4.3 Version Control 27
5.5 Summary 28
Chapter 6 – Testing and Logging 28
6.1 Introduction to Testing 28
6.2 Unit tests and UI Automation 28
6.3 Error Logging 30
6.4 Summary 30

Chapter 7 – Evaluation 30
7.1 Introduction to Evaluation 30
7.2 Justification of features and problems encountered 30
7.2.1 Cognitive Recognition 30
7.2.2 User Application 31
7.2.3 Real-time Robotics Control 32
7.3 Summary 33
Chapter 8 – Critical Reflection 33
8.1 Introduction to Critical Reflection 33
8.2 Future considerations 34
8.2.1 User interface 34
8.2.2 Robot control 36
8.2.3 Process changes 36
8.2.4 Post-WIMP 36
8.2.5 Techniques 37
8.3 Ethical issues 37
8.4 Summary 38
Chapter 9 – Personal and Professional development 38
Chapter 10 – Conclusion 39
References 40
Glossary 45
Appendix A – Project Specification 46
Appendix B – Ethics Checklist 49
Appendix C 50
Appendix D 51
Appendix E 52
Appendix F 53
Appendix G 54
Appendix H 55
Appendix I 56
Appendix J 57
Appendix K 58
Appendix L 59

Appendix M 60
Appendix N 61
Appendix O 62
Appendix P 64
Appendix Q 65

Chapter 1 – Introduction

1.1 Introduction

Every 10-15 years a new computing platform emerges, from mainframes to PCs, to the web and then mobile. The potential surrounding the next evolutionary step in computing where immersive devices are used in our everyday lives is explored. Devices connected to our vision will likely thrive initially, with products like Oculus Rift and Microsoft HoloLens driving innovation. Then after the indoctrination period, people will be more accustomed to these immersive devices and brain-computer interface (BCI) augmentation will eventually be used in most aspects of life.

Electroencephalography (EEG), Robotics and brain-computer interface (BCI) technology continues to improve at an unprecedented rate (Rebolledo-Mendez et al., 2009). This is evident in the availability and affordability of research devices and the increased interest in these areas.

Brain-computer interfaces (BCI) are fascinating, offering a completely unique way of interacting with technology that isn't ordinarily available. Most people use their hands to interact with a desktop computer, mobile phone or gaming console, but using just the brain as the input controller provides near limitless possibilities. Devices which are incomprehensible by today's standards will take advantage of these technological advances, especially with the rise of ubiquitous computing, where computing can stem from any type of device, in any location and format (Weiser, 2002). The potential uses encapsulating this technology will have a profound impact on the world. With the emergence of Bionics, people with prosthetic body parts can move their artificial limbs using their brain (Herr, 2014). Being able to control the lighting in a room, change the TV channel, make a phone call to a family member, will all one day be controlled using only thoughts.

1.2 Project aim

This project demonstrates that by measuring the electrical fluctuations of the neurons in the brain using electroencephalography in conjunction with a brain–computer interface, *real-time* control over real world entities such as a robot can be achieved. The project will use sophisticated APIs via a BCI to interoperate together and produce a deliverable, demonstrated in the form of a robot that is controlled through EEG data derived from thoughts.

A background investigation will be conducted in the relevant subject areas. The insight gained will be applied to establishing an understanding of the hardware and techniques required, respective of the resources and time available. A set of requirements defines what the application is required to encompass. Design decisions are discussed which justifies the hardware, software, and processes that the development should follow. The progress of development is monitored with justifications of the implementation methods. In-depth testing of the application is done and an exploration is conducted into why a test-driven iterative approach improved the quality of the application. An evaluation of the project and developed application are made. The entire project is then critically reflected upon, expanding into future considerations and ethical issues with brain-computer interfaces. Followed by an insight into how this project improved personal and professional development skills. A final conclusion summarises the findings.

Chapter 2 – Background Investigation

2.1 Electroencephalography (EEG)

2.1.1 Introduction to Electroencephalography

Electroencephalography (EEG) measures voltage fluctuations from the neurons in the brain. (Niedermeyer, Schomer and Lopes da Silva, 2011). A typical non-invasive method of using EEG is to place electrodes along the scalp to produce an electroencephalogram. Having previously been used primarily for medical reasons, scientists are finding new experimental ways to take advantage of the ability to read brain waves (Blankertz et al., 2010).

2.1.2 A Brief Historical look at EEG

Over the last 100 years, extensive progress has been made on Electroencephalography. In 1875 Richard Caton an English physician from Liverpool was interested in electrophysiology, he went on to study the cerebral hemispheres of rabbits and monkeys (Coenen and Zayachkivska, 2013). Caton discovered the existence of electrical currents in the brain and laid the groundwork for future researchers (Jacks, 2003).

It wasn't until 1924 that significant progress would be made, Hans Berger a neuropsychiatrist discovered the first human electroencephalogram (EEG) (Kaku, 2014). He found that without opening the skull, weak electric currents can be detected and recorded on paper. Berger is also known for coining the term electroencephalogram (Chapin and Russell-Chapin, 2013).

2.1.3 How it relates today

Electroencephalography is now used in myriad ways, from diagnosing medical conditions such as epilepsy (EpilepsySociety.org.uk, 2015), to being used with prosthetic limbs (Loudin et al., 2007), and even in computer games (Nijholt, Bos and Reuderink, 2009).

Scientists are measuring EEG signals using new non-invasive innovative technologies to lead research. Emotiv, NeuroSky, OCZ, Psychlab are just some of the companies that are getting involved in EEG.

2.1.4 Usage of EEG

Within the neurons of the brain, electrical fluctuations can be measured using an Electroencephalogram with what is generally a non-invasive device placed along the scalp (Niedermeyer, Schomer and Lopes da Silva, 2011). Using EEG data it is possible to diagnose sleep disorders, comas, epilepsy and many other health problems (NHS.uk, 2015). It is also possible to map specific thoughts and even muscle movements to EEG signals, this makes it possible to tell when a person is thinking or moving in a certain way.

Event-related potentials (ERP) are a type of EEG data that is the result of a specific cognitive action or movement (Teplan, 2002). ERPs can be used to detect different cognitive actions that are being performed by a user wearing a brain-computer interface.

EEG reading offers a completely non-invasive way of reading information from the brain, so it holds great potential use in medical and software applications with no risk of harming the user.

2.2 Brain-Computer Interfaces (BCI)

2.2.1 Introduction to Brain-Computer Interfaces

A Brain-Computer interface (BCI) also known as a "Neural interface" is a method of directly communicating with an external device via the brain (McFarland and Wolpaw, 2011). There are many different kinds of BCI devices, with EEG-based being the most common.

Non-invasive EEG-based BCI devices are popular due to the ease of using the device, and providing worthwhile results without the potentially damaging repercussions of invasive devices.

Invasive BCI devices read cranial nerve information by embedding electrodes directly into the brain. A more accurate signal is recorded from invasive devices however due to the practicalities of such devices the non-invasive form has a wider applicable range (Yanagisawa et al., 2010).

BCI technology has received significant attention from respected experts. Microsoft has established a BCI research team (Research.Microsoft.com, no date) and Elon Musk publically stated interest in this area (Invest Hong Kong, 2016).

2.2.2 Advantages of BCI

Many people with severe physical impairments require assistive technology to aid them in their day to day activities. Those who suffer from total paralysis or have locked-in syndrome are unable to use conventional methods of assistive technology. EEG-based BCI has a unique advantage, if a user has cognitive capabilities then it could be possible for them to use a BCI as an alternative technological solution (Schalk et al., 2004), (Mohammed et al., 2015), (Shi, 2015), (Guneysu and Akin, 2013).

2.2.3 Disadvantages of BCI

As explained by (Allison and Neuper, 2010) about 20% of people are unable to proficiently use a typical BCI system. This phenomenon is known as "BCI illiteracy" (Vidaurre and Blankertz, 2009). Extensive efforts have been made to overcome these problems through various means such as extensive training of the user with the device, improved signal processing, and error correction, all of which have only been partly successful. Random artefacts from eye or muscle movements can create significant attenuation distorting waveforms in the EEG (Ekanayake, 2010). The signal to noise ratio in devices like the Emotiv EPOC + is significantly worse than conventional medical devices (Wójcik, Wierzgała and Gajos, 2015). It is important to consider the implications these difficulties could have with the development of the project.

2.2.4 Real World Usage of BCI

The most popular EEG-based BCI devices are made by Emotiv, NeuroSky and Muse, each targeting different audiences. The main devices researched are the Emotiv EPOC and the NeuroSky MindWave. Both devices offer a Software Development Kit (SDK) with the potential to be used in this project. The research found that the MindWave is cheaper and less obtrusive (Grierson and Kiefer, 2011) than the EPOC, however, the EPOC has greater precision in detecting users' mental states (Folgieri et al., 2014).

BCI today is mostly used for research purposes. The BrainDriver project uses the Emotiv EPOC + BCI and allows cars to be drove using thought alone (Waibel, 2011). The Wadsworth Center, a public health laboratory in New York, is dedicated to science in the pursuit of health, they focus primarily on non-invasive EEG-based BCI methods and their developments have resulted in prototype systems for everyday use in people's homes (Vaughan et al., 2006). Honda's ASIMO has been controlled using EEG and near-infrared spectroscopy (NIRS) sensors. They allow the user to imagine moving their hand in a specific direction, then after some compute time the robot can move in the direction commanded. The commands issued are far from perfect, however, when combining EEG with NIRS up to 90% accuracy was achieved (Blain, 2009). With further refinement this huge leap in technology has a profound impact on the future of BCI.

Honda's ASIMO being controlled using electroencephalography (EEG) and near-infrared spectroscopy (NIRS). ***APPENDIX C***

BCI is a newly emerging area of research, and the impact it will have on the world is exciting to see unfold.

2.2.5 Alternative BCI techniques

This project focuses on using a non-invasive EEG-based device, however, it is crucial to study and understand the importance of other types of BCI.

Electromyography (EMG) and Electrooculography (EOG) track muscle and eye movements respectively. The data produced using these technologies can also be used with BCI devices.

(Yanagisawa et al., 2010) suggest using NIRS as an alternative to EEG, stating "*EEG has low spatial resolution, and measurements focused on particular parts are difficult. Furthermore, this method is sensitive to electric noise. NIRS measures variations of oxygenated hemoglobin and deoxygenated hemoglobin using near-infrared rays; thus, it enables experiments on the subject in a natural state, with few restrictions on body movement. This system has higher spatial resolution than systems that use EEG and uses near-infrared rays; thus it is robust against electric noise.*"

Other more intrusive methods can connect directly into the brain, offering higher quality readings. Due to the practicalities and potential health risks of this, this area has not been as thoroughly studied, as it is not viable option for a first-time prototype application.

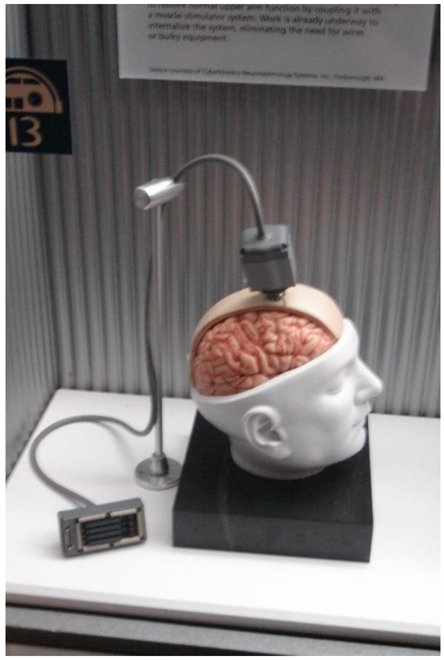


Illustration of the design of a BrainGate interface developed by Cyberkinetics. ***APPENDIX D***

2.3 Robotics

2.3.1 Introduction to Robotics

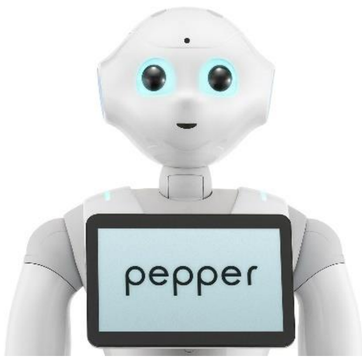
Robotics have changed the 21st century, they transformed the manufacturing industry from CAM designs to full on manufacturing automation (Groover, 1987). Currently, there are around 1.5 million robots hard at work, with about 230,000 in the US alone (PwC.com, no date). The existence of robots in everyday life has become much more apparent than ever before so looking at how they can be used in further innovative areas is important for the robotics industry.

Robotics suffers some of the same problems that arise from Artificial Intelligence. Currently "Weak" or "Narrow" AI currently performs much better in a specific problem domain than a generalised "Strong" AI (Nath and Levinson, 2014). Most robots have specific tasks which they were purposely created to excel in. Some smart architectural design decisions allow for some robots to complete a wider variety of tasks but are still limited by supervised learning (Zhou et al., 2012).

Neurobotics is one way which could overcome this problem, by intuitively controlling robots naturally by using the brain (Kandel, Schwartz and Jessell, 2000).

2.3.2 Real World Usage of Robotics

The range of robotics in the modern era has grown astronomically.



Alderbaran's Pepper Robot. ***APPENDIX E***

In France, Aldebaran a robotics company that has been popular in the public eye created their first humanoid robot Nao and has been the driving force in robotics for many educational institutes. Pepper a later project also crafted by Aldebaran is the first humanoid robot designed to live with humans, with over 140 of them even being used in SoftBanks Mobile retail stores in Japan, using the robot to welcome and inform customers (AldebaranRobotics.com, no date).

Even in Cornwall in the UK, Engineered Arts creates robots that are designed to interact directly with people, their "RoboThespian" robot can tell jokes, stories or even sing (EngineeredArts.co.uk, no date). Tobit Software a German company have created entertaining pole dancing robots (Taha, 2012). A robot restaurant is situated in the busy districts of Tokyo where giant robots are used to parade with performers (Shinjuku-robot.com, no date).

Robot competitions are also thriving, the Lego Robot Challenge allows teams from all around the world to compete for prizes (FirstLegoLeague.org, no date). The Darpa Robotics Challenge is designed for an experienced audience and advanced robotics, gathering some of the best talents from around the world to compete for top cash prizes (TheRoboticsChallenge.org, no date).

The recently Google acquired Nest is a good example of a home automation device that acts just like robot serving to solve a specific problem domain (Nest.com, 2016). Humanoid looking robots are generally easier to define, their appearance matches the general model of a robot. (Thobbi et al., 2010) explain about achieving remote presence using a humanoid robot controlled by a non-invasive BCI device. The numerous sensors available on humanoid robots, such as vision, aided by the further development of virtual reality allow for a more human-like experience mimicking the senses of humans. This level of precision controlling a robot remotely can have many practical applications such as being able to perform surgery in remote places (McCloy and Stone, 2001), working in hostile environments that are hazardous to humans (Bellingham and Rajan, 2007) or simply working in a remote environment.

2.4 Summary

The research carried out highlights the fact that the topics involved, are very broad and span many different subject areas. Further proving how instrumental a narrow focus is when developing the application. New areas and techniques have been identified and explored. A non-invasive EEG-based BCI event-related potential technique is highlighted as the most practical method to correctly understand cognitive actions produced by the brain. In many published articles worked on by a team of academics, they raised the issue of detection accuracy being the most challenging and the existence of "BCI illiteracy", highlighting the fact that 20% of people may be unable to BCI devices due to recognition issues.

The research carried out also showed that robotics is a very comprehensive field. This in-depth research has further proved the importance of using a robot in the simplest form to complete only the most basic tasks. The robot only needs to represent a physical entity, following the commands interpreted by the application.

With the research complete it was now possible to determine what is possible within the timeframe and resources at hand.

Chapter 3 – Requirements

3.1 Introduction to Requirements

The research was conducted in the related subject areas and consideration of the technologies and architectures to be used in the project was completed. The project then progressed towards the requirements stage. The requirements stage focuses on what the application (project deliverable) will set out to accomplish.

3.2 Requirements elicitation

Introducing a brain-computer interface and robotics increases the complexity of the project, for this reason a throwaway prototype was chosen as the most appropriate method of supporting requirements gathering (Sommerville, 2000).

The application for the project is focused on a specific target audience. It is not intended to be a system for wide adoption after completion. The software will only operate with the specifically intended software and hardware devices. Implementing a cross-platform solution is not the point nor focus of this application. A narrow focus on frameworks and architecture support is necessary in order for the project to be successfully completed on time. It should attempt to take advantage of the hardware that the system is targeted for.

The system should support, or be easily extendable to support multiple users using the application. It should have a feature which enables EEG data from different people to be read and interpret those actions as required. If the system does require other users to test the application an appropriate consent form will have to be distributed to those involved.

The application will need to be user tested in order to ensure the data is being interpreted as desired. Conventional testing methodologies should be followed to improve the quality of the application. The prototype application requires that the main features supported are robust and built upon strong foundations.

3.3 Limitations

The requirements importantly dictate the limitations of the application.

Due to potential difficulties in reading and interpreting data, a limited number of directions will be supported. A basic 4-way system (UP, DOWN, LEFT, RIGHT) is the most sensible approach to start with. This would if accurately interpreted, allow the robot to move in those directions. Once the application has a foundation and framework to work with, it will be possible to establish greater control of movement in the future.

The system will only work with a specific robot and cross-platform usage is not supported due to time restrictions.

3.4 Summary

The requirements have been identified as follows:

3.4.1 Client Requirements

- Calibrate and create a unique profile with the associated EEG data.
- Can choose the direction of cognitive action to be recorded.
- Is able to move a robot using just the brain.

3.4.2 Device Requirements

- Be non-intrusive and not cause harm to the user.
- Reasonably easy to connect and setup.
- The BCI device must expose data via an API.
- The robot must be able to receive and follow out commands in *real-time*.

3.4.3 Application requirements

- Can correctly interpret thoughts as specific cognitive actions using the APIs.
- Must correctly send commands to the robot.
- Can be easily expanded to work with additional robots/devices.
- Support multiple users.
- A set architecture is followed to keep software consistent and maintainable.
- Handles communication with all devices and appropriately handles disconnects.
- Enterprise architectural principles and coding conventions are followed.

Chapter 4 – Design

4.1 Introduction to Design

The design phase of a software development project is designed to satisfy the requirements of the project while taking into consideration the project timeline and laying the foundations for future stages. Software development has come a long way since the 80s, from the scale and prevalence of software companies to the architecture which software is required to support. That being said many old conventional techniques such as COCOMO (Boehm, 1984) and MVC (Freeman and Sanderson, 2012) are still used in many software projects. The modern ideas from the “agile manifesto” (Beck et al., 2001) ideally suit throwaway prototypes.

Since this project was undertaken independently an agile methodology was chosen. This would allow slight changes in the requirements while retaining the focus of the project. A test-driven development (TDD) methodology will be used throughout the development of this project. The application to be developed requires high reliability from a limited amount of features, supporting an ideal case for TDD. For this reason along with maintainability, it is important to keep work modular.

4.2 Hardware

4.2.1 EPOC Emotiv System

Although there are quite a few choices for EEG devices, the Emotiv EPOC + headset was chosen. The reasonable price of $500 (US Dollars) and ease of buying online, was favorable in this decision process. The device has been featured in TED talks, gained attention in the news and used in various research projects (Le, 2010). The Emotiv forums and library documentation for SDK support look promising. With no prior experience in developing BCI applications, the research found that the application could benefit from the greater precision offered by the EPOC, proving instrumental when attempting to manipulate robot control.

The Emotiv EPOC + headset offers full cognitive control capabilities, allowing thoughts to be programmatically interpreted from EEG into a cognitive action. The pure raw graph data which is usually seen on an electroencephalogram is not needed, it is beyond the scope of this project. Programmatically the EEG data can still be accessed but it is in a completely different format. The EEG data provided by the EPOC for cognitive recognition also includes Electromyography (EMG) data from muscle movement (Kamen and Graham, 2004) and Electrooculography (EOG) data from eye movement (Bulling et al., 2011). Other EEG devices attempt to filter out the other non-brain activity data (Emotiv.com, no date). When physical muscle movement is done, EMG and EEG data is received due to the brain working in conjunction with the muscles.

4.2.1.1 Emotiv EPOC headset

The Emotiv EPOC is a non-intrusive, Brain-Computer interface headset made by Emotiv Systems. It offers scientific contextual EEG data using 14 EEG channels and 2 references as sensors. The headset also has 8 sensors around the frontal and prefrontal lobes that help in detecting signals from facial muscles and eyes. Facial muscles are monitored in the EEG signals a supports blinking, winking, eyebrow movement, smiling and teeth clenching.

The device can help recognize patterns in electroencephalograms and provides APIs to understand cognitive actions through detection libraries. Created for research purposes the headset gives scientists and developers alike a large toolset to work with, from a self-sufficient Software Development Kit (SDK) to wireless Bluetooth capabilities. The Emotiv EPOC headset is intended to give a more pleasant user experience over conventional systems.

Conventional EEG system (left) and an Emotiv EEG system (right). ***APPENDIX F***

4.2.1.2 Emotiv EPOC SDK API

Initially a user creates a profile, and are then required to complete a set of training tasks. The user performs a thought for different cognitive actions when prompted, this data is then recorded in their profile. The recorded profile is used when reading live EEG data. If the live data received is the same or similar to the recorded data in the user profile, it is possible to see which cognitive action is being performed. A custom sensitivity algorithm can be used to help filter out noise in the data.

The complete list of available actions supported by the EPOC SDK is as follows:

- Neutral
- Push
- Pull
- Lift
- Drop
- Left
- Right
- Rotate Left
- Rotate Right
- Rotate Clockwise
- Rotate Counter Clockwise
- Rotate Forwards
- Rotate Reverse
- Disappear

Using the SDK and provided algorithms, a user will complete training for 1-4 of the actions, not including the neutral action. The application in development will only focus on Push, Pull, Left and Right cognitive actions. The headset limits the total number of actions to 4 per user profile, due to the increasing difficulty in differentiating the data between many different potential actions. It is technically possible to override this limitation by modifying the profile class and performing other modifications to read the data correctly. The 4 actions per profile limit are kept, and consideration is made on lowering this to 2, as the main purpose of this project is not to attempt to differentiate dozens of EEG signals but to interpret EEG data and send specific commands to a robot or similar end-device.

There is also "Affective" data that can be interpreted, these are emotional states such as frustration or excitement. This data should be displayed to the user in a meaningful way.

4.2.2 Zumo Arduino Based Robot

A Zumo Arduino based robot was chosen due to the compatibility of the language and framework. It can integrate easily with the main system under development. The Zumo is limited on physical movements, it can move in 4 basic directions, a perfect match for the project application.

Originally a Nao robot was considered, but to keep the project focused on the main objectives, the decision to not use the Nao was made. By selecting to use the Zumo, some unnecessary and unrelated complexity was removed. The application should be easily extendable, allowing support for additional robots to be implemented. It is important to note that under the time constraints Hofstadter's law has to be respected, which states "things always take longer than expected".

4.3 Implementation Plan

4.3.1 SDK - Programming Language

The Emotiv EPOC SDK has support for the following programming languages

- C#
- C++

- Java
- Objective-C
- Python

The programming languages Python, C#, C++, and Java were examined initially, to gain a better understanding of how the SDK actually works. Even at an early stage, before the development begun simple experiments were done to connect to the headset.

Previous knowledge gained from working in the software industry made it known that the features of SDKs and API wrappers can sometimes vary depending on the language. Experimenting early without entrusting the SDK completely could highlight potential issues. A basic Python script that connected to the headset was created but the decision was made to go with C# as the primary development language. C# has matured over its 16 years of existence and has been well supported by Microsoft, who provide many code tutorials and examples via MSDN. As a 'C' based language it offers compatibility with Arduino while also providing a well-established environment for simple GUI development.

4.3.2 Integrated development environment

The following integrated development environments (IDE) were considered initially.

- Visual Studio
- Python IDLE
- NetBeans
- Eclipse

NetBeans and Eclipse both offer a developer-friendly solution. Using Java with these IDEs provide JavaDocs support, a useful code documentation generator. JUnit is an extendable, feature-rich testing framework, boasting automation and unit test functionality. If Java was chosen as the primary language, one of these IDEs would have been used.

The decision to use C# as the primary programming language for the SDK and application influenced the decision in choosing Visual Studio as the main IDE. Visual Studio boasts many inbuilt features and powerful debugging capabilities, such as memory profiling, and CPU monitoring.

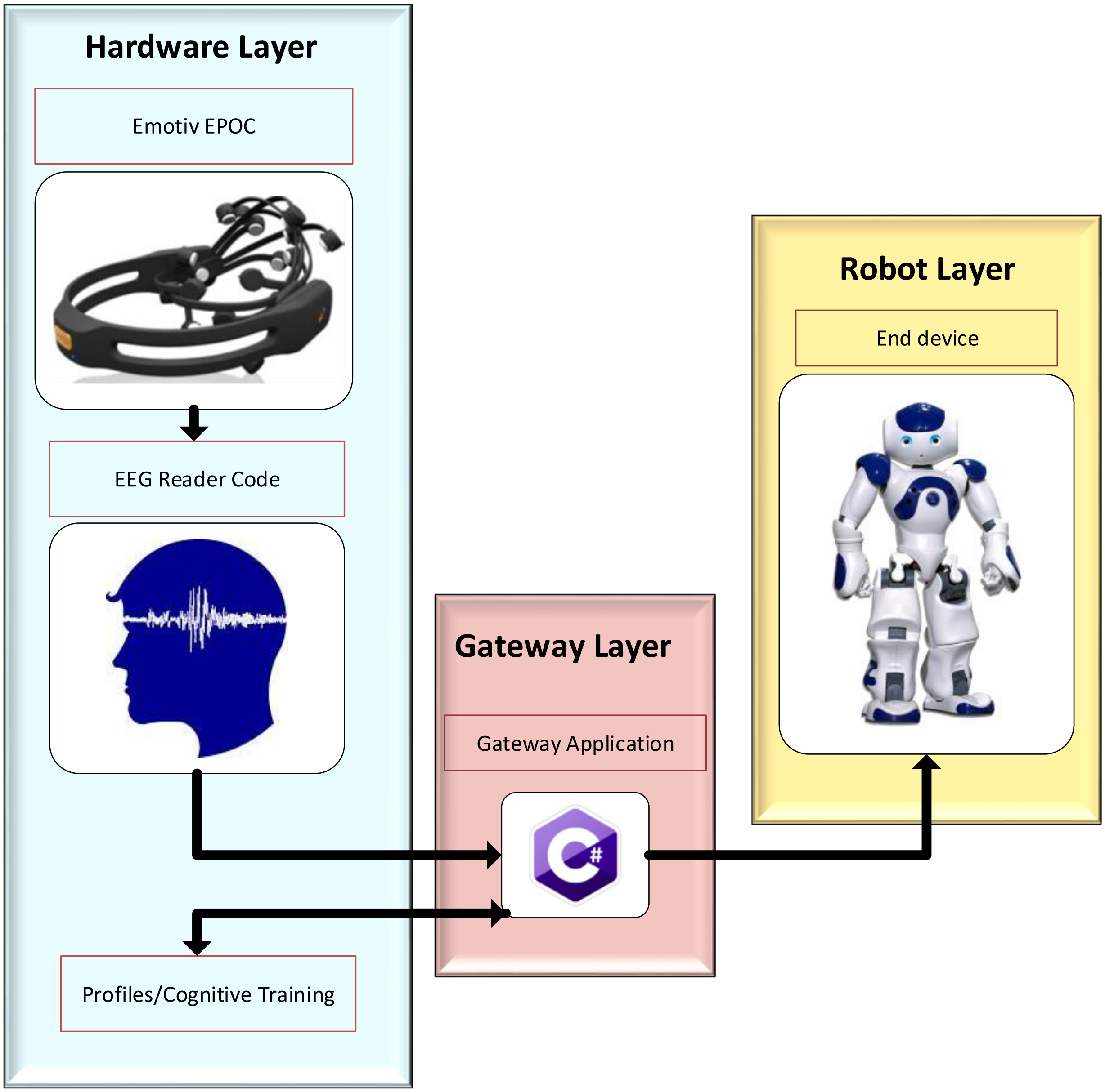
4.3.3 Version Control

For version control, GitHub will be used, it can track software throughout the development stages. Different "revisions" and "branches" can be used, allowing for changes to be implemented, easily reverted or discarded at a later date (Git-Scm.com, no date). The latest versions of Visual Studio integrate with GitHub for even greater control of the code.

Apache SVN, BitBucket, and even Perforce were briefly investigated but by using GitHub the project will contribute to the professional open-source profile being established by the developer.

4.4 Conceptual Architecture

In order to fully read EEG data from the brain, interpret the data and then issue commands to a robot there are multiple layers of the software architecture.



Conceptual Architecture. ***APPENDIX G***

Based on the conceptual architecture, the following can be discerned:

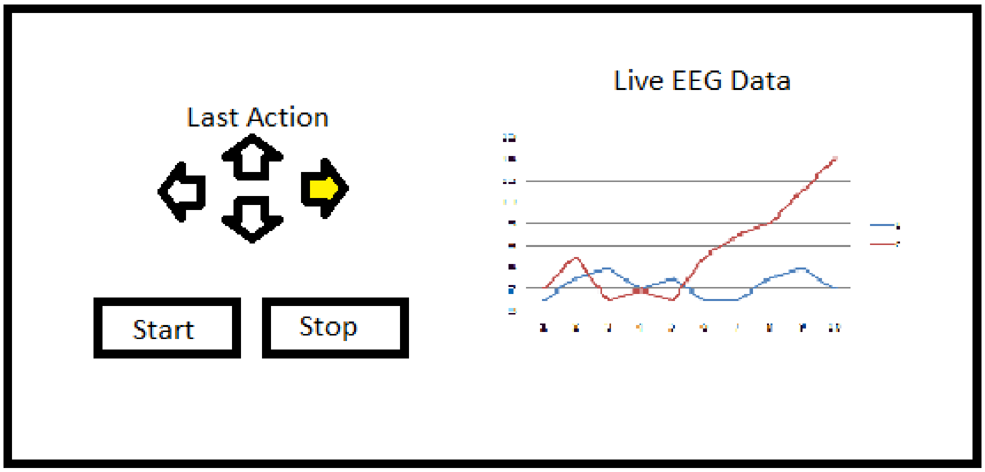
- Hardware Layer: Users wear the Emotiv EPOC headset and the EEG reader code is responsible for collecting that data. To create unique user profiles and perform cognitive training the headset must communicate with the Gateway Layer.

- Gateway Layer: The backbone of the project. It is in charge of interpreting the data received from the hardware layer and displaying its data in a meaningful way via a UI. The gateway application also sends commands to the Robot Layer.
- Robot Layer: Receives commands from the Gateway Layer and performs the appropriate movement/action.

4.5 User Interface

Based on the requirements for the project, the User Interface will likely be fairly basic. The UI needs to display information about the EEG data that is being received and notify the user on the commands that have been sent to the robot.

The following is a basic UI design that has been devised.



Basic UI Design. ***APPENDIX H***

4.6 Test Plan

4.6.1 Test-Driven Development

Unit tests will be written early on and the code would have to be written to pass those tests. In some cases, a general unit test would have to be modified to fit with the style at a later time.

Backend unit tests will be used to ensure the core classes behave as expected.

UI Automation with acceptance testing can ensure that the correct messages are being displayed to the user.

The following tests have been selected due to their necessity in developing a quality prototype.

- Gateway Layer can receive data from the Hardware Layer.
- Gateway application can interpret EEG data and display it on a UI.
- Gateway application can send commands to the Robot Layer.

Additional smaller unit tests will be created but these 3 tests, test communication between every architecture layer and are at the core of what the project application is trying to do.

Unit tests should input data randomly between the accepted range and sometimes outside the expected range to ensure error handling is done. Using a randomized and specific input technique it is possible to run thousands of unique tests in either a specific or randomized order.

By keeping individual assertions separate from test methods a chained randomized method can be used, allowing for a very high percentage of code coverage in these areas. Running nightly builds and outputting anomalies found is very useful. Considerations were made to configure a Jenkins test environment for continuous integration testing (RCP-Vision.com, no date), however, due to time constraints the decision was made not to include this.

4.6.2 User Testing

The Emotiv EPOC EEG headset is completely non-invasive, however, potential ethical issues could arise from other users using the device. The device will only be tested on one main computer due to the installation of Emotiv EPOC drivers required on the machine beforehand. There are also risks with only the developer testing the software. Developers know how to use their application and their view of how to use the software could be different from other user's expectations. Having a user available for testing would be challenging as they would be required to be available during most of the development phase. The decision was made for just the developer to use the headset.

If time allows then perhaps further UI evaluation testing could be done to gain feedback on how the software works. This would involve, the developer being the only one to use the device but showing the UI to a set of users and acquiring feedback. Using this feedback the UI could be updated to meet user expectations.

4.7 Summary

The design satisfies the requirements for the application, it sets out a plan of how to interpret cognitive actions and then send commands to a robot. A conceptual architecture has been created that lays out important parts of the system. The hardware and software to be used in developing the application have been selected.

The software has currently just been designed to work on Windows platforms, due to familiarity with the development environment. Further development of the UI for assistive technology purposes has not been researched thoroughly but has been considered. The software follows a standard desktop application style design, requiring a mouse and keyboard.

Challenges will arise from the lack of experience in the domain areas covered by this project. Research can be done beforehand to prevent problems but it is important that the project is developed understanding that change is probable. It may not be possible to know all the areas which could change but by being flexible, small changes won't require a complete redesign of the system, this is where following an agile methodology comes into play.

Keeping the scope of the project focused is one of the main reasons for not developing other features into the application.

Chapter 5 – Development

5.1 Introduction to Development

This section explores how the development of the application progressed.

The application developed represents the conceptual architecture designs that were devised. An object-oriented programming paradigm is followed and separate classes implement an interface to establish a connection with a robot. The main part of the architecture which users interact with is the gateway layer. This abstraction allows for a narrower focused layer of software to complete specific tasks.

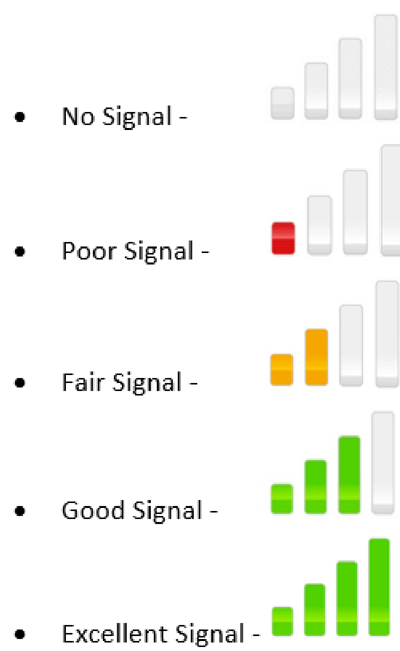
The main gateway layer of the application has been developed using C#. Its roles include communicating with the Emotiv EEG headset, interpreting the data received and then issuing commands to an end target device, such as a robot.

5.2 User Interface interactivity and functionality

The front-end UI for the gateway application has been developed using .NET WinForms. WinForms provide a simple way to create an event-driven UI that is fully supported by the .NET framework, ideal when working with the C# EPOC SDK.

5.2.1 Connecting to the EEG EPOC

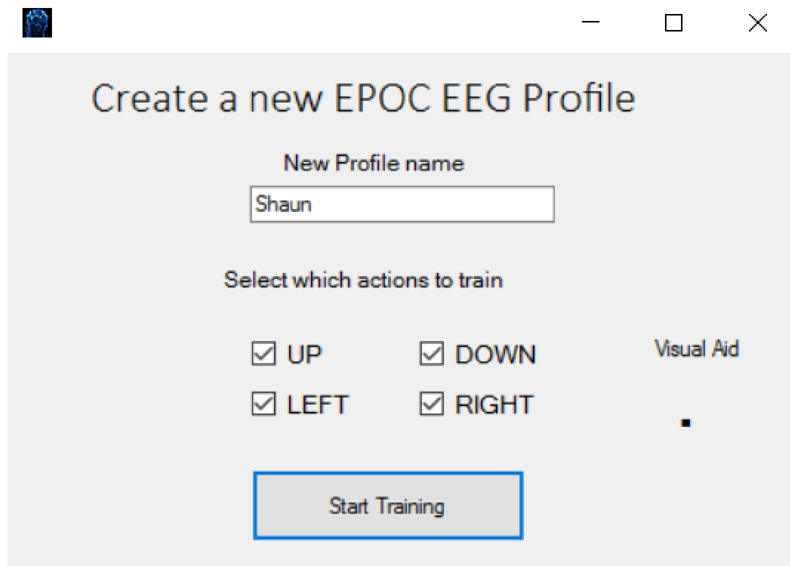
The application developed automatically detects when it can connect to the EPOC headset and at runtime, instantiates the EmoEngine class. This instance is then used to collect all types of data from the headset. A signal image is displayed in the top corner of the application, which notifies the user on the connection strength. Instead of simply providing the user with a basic number from 0-4, a graphical image is used to improve usability.



Application Signal Display. ***APPENDIX I***

5.2.2 Profile Management with Cognitive Training

A user profile is used to load and save information about a user's specific EEG cognitive actions. Initially, a user will create a profile and choose the actions they wish to train. The visual aid prompts the user to think about specific actions a recording will take place. After the training has been completed for each action selected, the user is then able to use the application. A user profile can be saved and loaded, preventing the need to complete the training each time the user wishes to use the application.



Application New Profile Interface. ***APPENDIX J***

The gateway application knows what cognitive data is in the user profile so when the latest live data is matched, it recognizes the cognitive action as a particular movement.

5.2.3 EEG Data Retrieval

The engine class is instantiated at runtime, using this instance a ProcessEvents method can be called.

```
engine.ProcessEvents();
```

Technically there are 2 constructors for the above method with the other being

```
public void ProcessEvents(int maxTimeMilliseconds)
```

When using the default constructor a 0 is passed as the default parameter. A custom EventHandler is then used to see when an EmoState or CognitivEvent update has occurred. By using an EmoState and EE_CognitivAction_t object it is possible to detect the cognitive action and related power. The power can be used to calculate how likely the user did actually think of this action.

```
EmoState es = e.emoState;
EdkDll.EE_CognitivAction_t currentAction = es.CognitivGetCurrentAction();
```

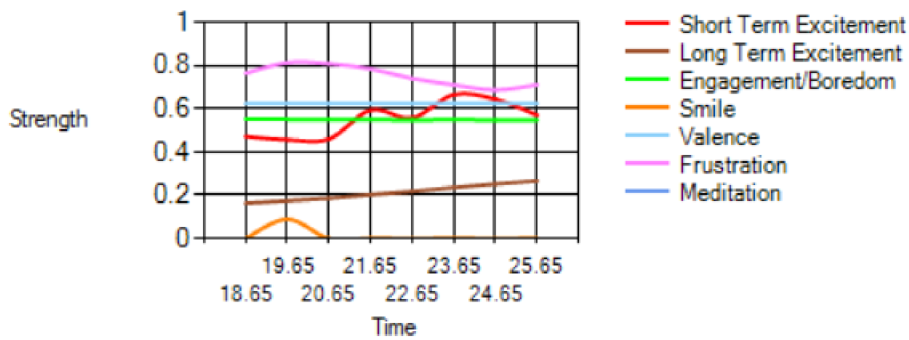
Once the computation for the detection algorithm has completed an EE_CognitivAction_t object is returned containing the action details. There are different sensitivities that can be configured, which determine how closely the live reading has to match the recorded data set. A too high sensitivity will likely incur false positives to be read, although if the sensitivity is too low it can be hard to get the correct cognitive reading. After a successful detection, the application will know which action was performed, correctly update the UI and then send the appropriate commands to the robot.

A limit of 4 cognitive actions is set by the hardware but it is possible to override this. During the development and constant testing of the application it was found that using 4 cognitive actions at once was significantly more difficult to consciously control. An assist mode was created which allows a user to only use 2 cognitive actions but control the robot with 4 movements. The assist mode is a simple check box that the user needs to activate on the UI, this “inverts” the actions, so a “forward” action will now command the robot “left”. If the users skill or device recognition at that time is not good enough to control 4 actions the assign mode can be used. Some potential problems were presumed in the design stages, but were obviously unclear until testing of the EPOC + had commenced.

5.2.4 Data Visualization Chart

A data visualization chart plots data received from the EPOC headset using Cartesian coordinates. The chart shows the latest 8 seconds of affective data and its corresponding strength received from the Performance Metrics API. Performance Metrics is a measurement of the user's emotional state that is detected using EEG.

The GUI displays the following emotional states:



Affective Data being displayed on the UI. ***APPENDIX K***

A new timer instance collects affective data and applies a similar technique previously used to collect emotive data. A custom event handler is activated and an Affective class allows only the required data to be saved, discarding unneeded information. The object is then used as a list to plot X and Y points for each series of emotional states. This approach simplifies adding the affective data to the chart.

5.2.5 EEG Data Logging

The Gateway application includes a logging feature, this allows data to be collected over long periods of time. It can be enabled or disabled at any time. A simple folder called “Logging” and a text file named “data.log” will be created.

The following is a typical single entry of logging data for emotive and affective information, along with the respective timestamp.

```
Log Entry : 7:27:52 PM Wednesday, March 23, 2016
Name: ShortExcitementLevel, Value: 0.0007831208
Name: LongExcitementLevel, Value: 0.0004284323
Name: FrustrationLevel, Value: 0.3647476
Name: MeditationLevel, Value: 0.3316737
Name: ValenceLevel, Value: 0.624995
Name: EngagementBoredomLevel, Value: 0.5530405
Name: SmileLevel, Value: 0
Name: Busy, Value: False
Name: SignalStrength, Value: GOOD_SIGNAL
Name: Timestamp, Value: 20.32997

-------------------------------

Log Entry : 7:27:52 PM Wednesday, March 23, 2016
COG_PULL - 0.4728535
```

Log Data Example. **APPENDIX L**

The name of the action or emotion is listed along with its related power value. With 0 being the lowest (no action/emotion) and 1 being the highest. A power of 1 is generally never achieved, from experimentation a power of 0.3+ is likely a correct detection. However, each person's brain is unique, adding a layer of complexity to the accuracy level algorithm used for detection. The accuracy level will need to be changed dynamically to be optimized for a wide range of users.

5.3 Real-time Robotics control

5.3.1 Robot Simulator

Initially, the UI designs lacked a way to start a Robot simulator, so additional buttons were added where required.

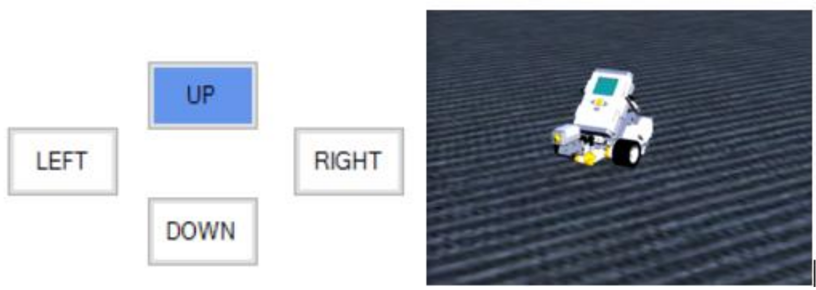
A MessagingInterfaces project was added to the Visual Studio solution, it contains 2 classes that act as an interface between a client and server using the Windows Communication Foundation (WCF) framework. The Gateway application acts as a server, issuing commands to the client. This method was chosen as it is built around the .NET framework and requires no 3rd party libraries.

The application utilises separate task parallelisation programming techniques (MSDN.com, no date) to take advantage of multi-core and hyper-threaded systems. A ServiceHost instance is created and initializes a server which will connect to 1 or many clients. To prevent process termination a background thread is used to launch a modified application based on the Microsoft Robotics Developer Studio. The simulator is in the same project, and has been modified to use a custom manifest layout with the client-side WCF interface that was previously created. Using the client-side ServiceHost, a connection to the server can be achieved, simplified by taking advantage of the WCF framework. A globally unique identifier (GUID) is created and registers with the server. This GUID represents the unique client that is connected to it, allowing for multiple devices to use the interface as a bridge for improved interoperability. Any command that is now sent from the gateway layer is received by the robot simulator. The robot simulator code, uses the UpdateJoystickAxes method to programmatically control the robots movements based on the commands received, in *real-time*. The simulator is configured to use a Lego Mindstorm NXT Robot but can be customized to use many different kinds of robots by using a different manifest XML file. The robot receives the command and completes the related movement in *real-time*.

At this stage, it is important to define the meaning of *real-time* and *near real-time*, as defined in (Telecommunications: Glossary of Telecommunication Terms, 1998) *"The distinction between the terms "near real time" and "real time" is somewhat nebulous and must be defined for the situation at hand. The term implies that there are no significant delays. In many cases, processing described as "real-time" would be more accurately described as "near real-time"*."

This distinction is important because although the robot is controlled in perceived *real-time* there are other processing and communication delays occurring. There is also an artificial pause after every command is sent. Based on testing of the system, this was implemented to help with improving control over robot movements. Without it, the robot would occasionally move slightly more than anticipated.

When a cognitive action is detected the application layer updates the UI and simultaneously commands the robot to move in the specified direction.



UI direction with robot simulator. ***APPENDIX M***

5.3.2 Real Robot

Zumo Arduino Based Robot. ***APPENDIX N***

A Zumo robot for Arduino has been used to demonstrate the software developed for this project. The physical capabilities of the Zumo are ideally suited to work in conjunction with directional based cognitive actions. The robot uses a Zumo Shield to connect to an Arduino Uno which in turn receives commands sent via a serial connection.

The Arduino offers an ideal environment for sending commands over a serial connection. Before the robot layer was even linked to the main application it was possible to emulate commands received using the Serial Monitor software built into the Arduino IDE. This allowed focus to be spent on ensuring the Zumo Robot can interpret each command and algorithmically calculate the direction it should move in.

The programming language Arduino uses is based on C/C++ (Arduino.com, no date) and as the Arduino is lacking in terms of raw performance, it was important to take advantage of the some advanced features of the language to optimize the efficiency of the code.

The Arduino program listens for a serial command, and upon receiving it, the validity of the command is checked and finally the motor speeds are set as required. Pointers with references to the speed variable are used, and are being modified only when required, this minimizes code usage while sustaining maintainability. A single speed loop is used which is manipulated where needed, this allows a single algorithm to control the movement of the robot for all directions. A basic timer that acts as a benchmark was used to further improve code optimization through an iterative process, improving code execution time helped gain an improved understanding of the Arduino architecture.

The setMotors function has a direction parameter which is used to set the motor speeds accordingly.

```
void setMotors(int dir){
  //use pointer to ref speed as we can keep the existing speed if we don't change it
  int *lSpeed = &speed;
  int *rSpeed = &speed;
  if (dir == 2)//Reverse
    setReverseSpeed();
  else if (dir == 3)//Left
    lSpeed = 0;
  else if (dir == 4)//Right
    rSpeed = 0;

  //loop from 0 to 400 unless reverse is active then speed=-400,limit=0
  for (speed; speed <= limit; speed++){
    motors.setSpeeds(*lSpeed, *rSpeed);
    delay(2);
  }
  //after the movement has completed, stop the motors (set speed to 0)
  stopMotors();
}
```

*The complete Zumo Arduino code can be seen in **APPENDIX O***

When the motors need to be reversed, the speed becomes the additive inverse of the limit and the limit is reset to 0. This allows the same loop to work in all situations.

```
void setReverseSpeed(){
  //set the speed to negative if command is reverse (2)
  speed = limit*-1;
  limit = 0;
}
```

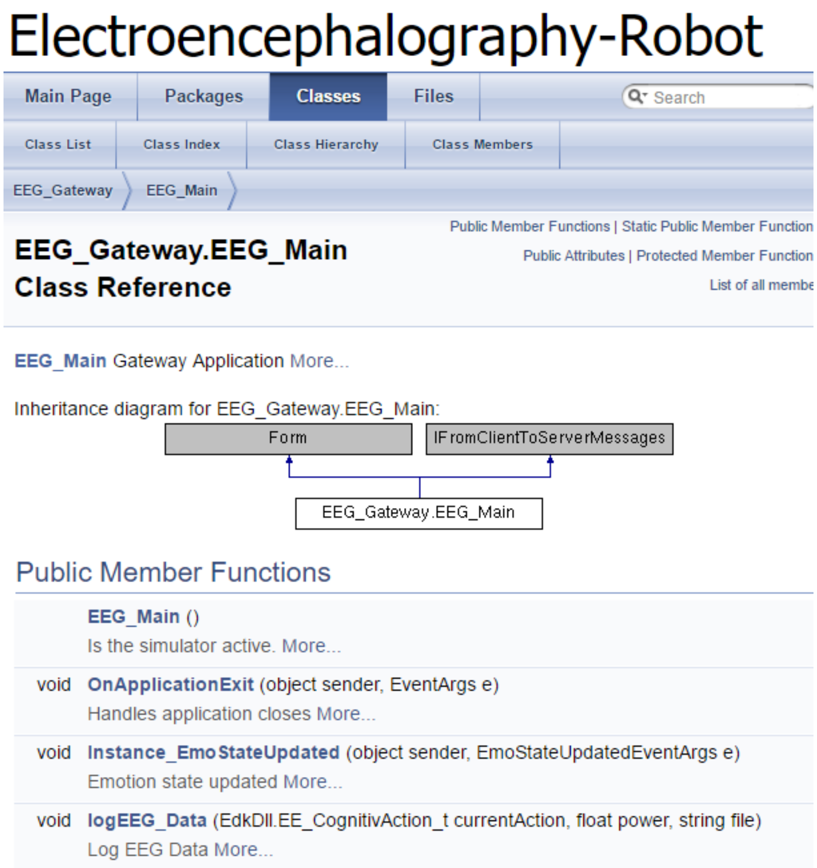
The gateway application takes advantage of the .NET Task Parallel Library to allow multi-core systems to be utilised when sending commands over a serial port to the robot. This allows for communication to be fluid, asynchronous and non-blocking.

5.4 Supporting tools

This section lists some of the tools that were used in the development of the project.

5.4.1 Documentation

Doxygen was used to facilitate code documentation, it offers HTML generated documents reminiscent to that of JavaDocs.



Doxygen code documentation. ***APPENDIX P***

5.4.2 Environment

Visual Studio 2015 and Arduino Genuino were the main IDEs used. Each offer unique tools that made development easier. The latest versions of Visual Studio have advanced resource diagnostic monitoring and memory profiling features, important for dramatically improving code performance.

Genuinos serial monitor feature simplified tasks, this feature wasn't discovered in the early design stages. Serial monitor can simulate the sending and receiving of serial messages, allowing focus to be applied on developing code that would appropriately handle the received message.

5.4.3 Version Control

SourceTree by Atlassian has GitHub integration and greatly improves the usability of Git repositories. A simple but affective GUI allows code changes to be easily tracked. The software shows additions and removals of all code commits, this feature is extremely useful to spot mistakes in code changes.

Using a single repository ensured easy tracking of code changes. Each layer of the architecture, including the robot simulator and Arduino code is controlled.

5.5 Summary

Following the architectural design that was devised, the information gathered from the headset can interact with another layer of software via a bridge. The application can also create or retrieve an existing user profile that has completed cognitive training.

Once a cognitive action has been recognized, a command is sent to a robot. The robot receives the command in *real-time* and executes a specific movement. The application developed in this project allows for both a real robot and for a software simulated robot to be controlled simultaneously using measured EEG data collected from the brain.

Code execution optimization ensured the flow of data being received from the EEG headset is stable and the chart which displays the latest affective information can remain unaffected by demanding tasks. Code naming and layout conventions were used throughout and adhered to, while remaining consistent in style and appearance.

Chapter 6 – Testing and Logging

6.1 Introduction to Testing

Software testing can be time-consuming, resource-hungry, labour-intensive and is even prone to human error (Ciortea et al., 2010). Despite these issues, testing can find problems early in development and can reduce costs associated with software updates/hotfixes and in rare cases prevent catastrophes (Amelan, 1996).

In an ideal world, every permutation of a piece of software would be tested. However, full test coverage is usually not possible, due to high cyclomatic complexities that apply to large software codebases (Sarwar et al., 2013). As (Dijkstra, 1970) points out "testing can be used very effectively to show the presence of bugs but never to show their absence". Even a relatively simple program could have thousands of possible input and output combinations (Myers, Sandler and Badgett, 2012). When faced with this issue a priority system can aid the creation of unit tests. A priority system can look at which parts of the system are more critical and which are more likely to occur, test coverage can then focus more in these areas. This is still problematic as issues can be caused by unknown code paths, anomalies can arise in unexpected ways or areas that required more focus were left without attention.

The importance of testing in the software industry should be higher than it currently is.

6.2 Unit tests and UI Automation

Unit tests can dramatically reduce bugs, however, unit tests are susceptible to problems. This could lead to detrimental effects, especially if a developer believes a specific piece of code is being tested. If an output is required to be a specific type but the test doesn't handle this properly, the developer could overlook this issue as the unit test is passing.

Following on from the test plan devised in the design stages, the following features were found to be of the highest importance requiring extensive testing.

- Gateway Layer can receive data from the Hardware Layer.
- Gateway application can interpret EEG data and display meaningful data on a UI.
- Gateway application can send commands to the Robot Layer.

To further take advantage of the features that Visual Studio has to offer, the built-in C# Unit Test Project was used throughout. In order for the unit tests to have public access to the desired classes, the project is linked to the main solution with references to each project.

All the tests use a single class but have individual test methods that are separated based on their purpose. A simple initialization method is always called first to instantiate the form and setup the test environment. Tests are then executed to ensure the form has been corrected loaded. UI automation is combined with standard unit tests to further increase test coverage.

The robot is given 30 random positions to move, followed by a test to check that the correct UI button has been highlighted on the UI based on the position of the artificial command given. The testLimit has an arbitrary value of 30 but could easily be set by the person executing the unit tests. It is important to test more than 5 possible actions even though the UI is limited to only 4 directions, as it is unknown if a particular sequence of commands causes problems.

```
public void UI_Updates(){
    Random r = new Random();
    int x = 0;
    for (int i = 0; i < testLimit; i++){
       x = r.Next(1, 5);
       moveRobot(x);
       isUIMoveBtnSet(x);
    }
}
```

A similar technique is used to test that data is being received from the headset and that it is being displayed in the chart on the UI. Other unit tests include ensuring the application layer can connect to the physical robot with the required baud rate, and is using the correct COM port. The robots movements and UI are tested simultaneously. It is important that the Zumo robot is placed in a large enough area, as it will move sporadically to test random commands. Turning the robot upside down and monitoring its movements proved useful. The same testing principle is applied to the robot simulator. The simulator application is launched programmatically by the unit tests. The server and client-side WCF connection are then tested and a randomized sequence of commands are used to test the robots movements.

All of the tests can be ran multiple times for wider coverage using the randomized input method. A high test coverage percentage is important, but for this prototype application establishing a framework in which the tests could be expanded upon is critical. The advantages of unit testing and UI automation were instrumental in exposing the prevalence of bugs and influenced the development of the main application. An issue highlighted by unit tests was when logging was enabled, it activated an incorrectly placed “if statement”, but only when the robot moved “right” (which in code is command number 4) multiple times. This was due to a simple logic mistake where the application was only looking for commands less than 4. It is possible that during the user testing stage, when logging was enabled, “forward” and “reverse” commands were used more than others and perhaps the user subconsciously wasn’t aware that they didn’t turn right multiple times in sequence.

One of the main justifications for using UI automation is when a feature or event changes, it is difficult to track or even be aware that the particular change has affected other components. With test-driven development, the tests are executed thought out and usually a problem can be found immediately after changes are made.

Testing a simple sequence of commands or parts of the UI in extremely fast succession is much more efficient than manual testing. Running thousands of separate tests which look at almost every possible input is useful, however, if code coverage is missing, the tests themselves have bugs, or the application behaves in a way which was never expected, issues with the software are still being overlooked. Manual user testing was still done alongside development but the use of unit tests greatly improved the quality of the project application.

6.3 Error Logging

As previously mentioned the application also features a logging system which can be used for monitoring a full range of activities. The logging system reports every type of exception or error, any problems with the software, or with the communication to the headset, can be found in a log file with a relevant timestamp. Allowing a dynamic logging system like this allows for the prototype to be used and then optimized based on the data that is being produced.

One example where data logging came in useful was when the UI didn't display EEG information but data was being received. This was a result of a problem with how part of the class parsed the data so the unit test also passed its checks. The logging did successfully show the data being received. From this both the application and unit test was updated. Without the data logging feature, extensive debugging would have been required to understand the problem.

6.4 Summary

Front and back-end unit tests with UI automation have been created, which satisfies the test criteria set out in the design test plan. Randomized inputs heavily test important parts of the system. The tests expanded alongside development, with basic tests being created initially, which improved as the development of the feature continued. Employing a test-driven development agile methodology was important, it allowed the development phase to be dynamically improved based on test results. Minimizing cyclomatic complexities for improved maintainability while effectively nullifying the presence of bugs.

Chapter 7 – Evaluation

7.1 Introduction to Evaluation

The aim of this project was to develop a prototype system, which allowed a robot to be controlled in *real-time* using electroencephalography. The project spanned multiple disciplines requiring constant focus on the main priorities in order to be successful.

7.2 Justification of features and problems encountered

7.2.1 Cognitive Recognition

Cognitive recognition using an event-related potential (ERP) technique allowed for thoughts to be recorded and mapped to specific cognitive actions. Event-related potentials were easily measured as they are the brains direct response after completing a specific cognitive or sensory activity. From the research conducted, EEG with an ERP technique is much cheaper than other alternatives such using an MRI machine, and with the current resources is the only viable option (Zanzotto et al., 2010).

A conscious thought is referred to as a "cognitive and/or affective task-relevant processes one is consciously aware of while attending to a task". A study by (Thorsteinson and Withrow, 2009) found that conscious thoughts are significantly slower than subconscious ones. The time of a measured conscious thought can vary, but can be as slow as 350ms (Bock and Marsh, 1993).

The main difficulties came in the reading accuracies of the EEG data. In order for the headset to work as intended without noise, each sensor needed to be lubricated and fitted tightly on the surface of the head. This caused problems as the sensors sometimes suffered from oxidation, resulting in malformed readings. Human skin and hair also absorbed the solution from the sensors, normally requiring a second fitting after lubrication in order to retrieve reliable EEG data. Headset fitting problems, due to physical skull size caused some issues as each sensor needs to be quite tightly fitted and throughout development at least 1 sensor was unable to connect and collect any data. The single sensor did not impede the usage of the system but probably contributed to the difficulty of development. EEG noise caused by a variety of factors, such as slight movements of the head or the sensors not being able to data at a given time sometimes caused anomalies. Restricting head movement helped keep the data consistent. Due to these problems occasionally incorrect or even no data was received. Noise was generated by moving the headset or from another nearby electronic device. The headset may not have been correctly fitted on the head or a sensor could have been missing contact with the scalp, a sensor could have also dried up and was unable to receive a strong EEG signal. The signal issues motivated the implementation of the signal status in the main user application.

When attempting to use multiple cognitive actions at once, the EEG data grew increasingly more difficult to differentiate. To skew the EEG data to be more easily read, usually, some extra facial expressions were performed along with the particular thought. Creating a stronger spike in the sine wave of the EEG which was easier to read when comparing with the original recordings in the user profile. Using a bi-directional method with only 2 cognitive actions made the EEG data easier to differentiate and improved the accuracy of detection. This discovered led to the development of allowing the user to choose how many cognitive actions they wish to train when creating a profile. An additional assist feature allows 2 cognitive actions to issue up to 4 movement commands.

Some information from the EPOC headset is not always accurate, mostly concerning facial expressions. This caused a small problem with the information that should be displayed in the chart, occasionally it detected that the user is smiling when that wasn’t the case.

As previously found in the background investigation (Allison and Neuper, 2010) explains that around 20% of users are unable to even use a BCI device, with the problems mentioned being reminiscent of the coding and design challenges that were found. All EEG-based BCI devices have a fairly high chance of error rates, good detection algorithms, smoothing, hysteresis and artefact filtering will help improve the data. This is not helped by the fact that the Emotiv headset is not shielded well further complicating matters (Duvinage et al., 2013).

Despite these issues, the cognitive recognition works quite well and is fairly consistent. There are still some problems, such as it may take an extra thought or 2 for the system to detect a particular action, or the power for the action wasn’t high enough to be recognized. Despite these shortcoming the system matches the requirements for the project.

7.2.2 User Application

The user application and UI changed slightly thought-out development as progression was made. Unit tests highlighted issues that the user should have been notified about, so the application was made to handle these prompts. Simple buttons or labels were added to display additional information to the user. The original design did not include a method to launch the robot simulator so the components for this were added.

A basic WinForms framework was used but further consideration into other, newer and more attractive frameworks could have been completed. A universal web framework could have been easily implemented while retaining the current features. Due to the UI design not being at the forefront of this project the decision was made to stick with WinForms.

The logging feature came later in the project and was not originally envisioned in the design. This feature was initially created as a simple debugging tool but changed into a user related feature. It allows users to monitor full cognitive actions over a long period of time, something that the UI currently doesn't support.

The addition of a bridge pattern allowed for an interface to be easily used, improving communication between separate parts of the system. The conceptual architecture shows a very good example of what the UI is intended to do and its purpose for this project.

From the software design it is evident that the project attempted to support flexibility and scalability. Advanced tools for were used to support the project and best practices from enterprise software were followed. A multi-tiered software conceptual architecture modularises components, providing abstraction layers for improved maintainability. A syntactically similar structure that respects coding conventions was used throughout development. Utilising concurrent tasks prevented delays and allowed for multiple components to work in parallel.

7.2.3 Real-time Robotics Control

The robotics control works intuitively and the additional robot simulation improved work efficiency. Using a Zumo Arduino robot instead of a Nao was a wise decision and necessary to keep work on target to be completed in time. Using a USB connection, instead of wireless communication reduced some extra unnecessary complications that have could have arisen from problems outside the project's scope. The robot simulator was a useful addition and is rewarding when working in synchronization with a real robot.

After implementation was complete, slight changes were made to the robot controls. A slight pause was created after each issued command, improving robot control. It was included because if a thought was held over 3 seconds the system may detect 2 actions for that direction, so the decision was made that this should be just a single action. This does slow down the *real-time* effect of the robots movements, however, the initial action is still fast and a pause only occurs after a movement is complete.

Significant improvements could be done in expanding the range of movements the robot could achieve. In the developed application the real and simulated robot are limited to 4 directions, however, when only 2 directions are used a higher level of precision is gained. Due to limitations with the SDK and EEG recognition complexities, the total amount of actions remained at 4, with room for future improvements in this area.

The end result matches the design and passes the requirements set out for this project, the robot can move in directions based on commands issued by the brain. The movements are not as fast as originally intended but research into EEG found that this would never likely be the case due to limitations in response time with the technology.

7.3 Summary

Overall the project proved the feasibility of the objectives that were originally devised, further optimizations, and feature additions inevitably could still be added. The completion of the project demonstrates the idea's viability in future technological use.

Despite this, problems were encountered throughout and decisions were made to decrease complexities. Development in the cognitive differentiator and the limited move set could have been extended, although the research conducted does prove it is possible, albeit it problematic. Each layer of the architecture worked well enough, but a single streamed communication method could have been better instead of opting for WCF for simulation control and serial connections for real robots, the original intention for these choices was to avoid conflicts.

The large technology focus and domain knowledge required increased the ambiguity of the project. Knowledge gained from the research and development stages of this project allowed for a complete working prototype to be completed. Justifications were given when changes in direction arose, of which later proved indispensable, allowing the focus to remain on the completion of the deliverable.

Chapter 8 – Critical Reflection

8.1 Introduction to Critical Reflection

In order to critically reflect upon the project, the objectives set out in the project aim need to be quantified. The project aimed to demonstrate that using an EEG-based BCI device, real-time control over real world entities such as a robot can be achieved. The end deliverable does demonstrate that this is possible, although research into EEG showed that the speed and accuracy of this technology have limitations.

Multiple sources of research have been used in an effective manner. Research ranges from the history of EEG, to highly detailed benchmarks of BCI devices comparing their precision. Many decisions made in this project have been influenced by reputable academic papers.

It became evident as to why companies spend tens of millions of dollars on researching these areas (Bogue, 2010). Requiring mass resources and unparalleled expertise in a multi-disciplined project proves to be highly challenging. The broad scope of knowledge required to fully understand all areas of Neuroscience, Computer Science and Robotics is inconceivable. To overcome this, solid Software Engineering principles were applied, which allowed for a successful and complete prototype application to be developed. Following a test-driven agile approach to development, small parts of the system were developed to a high professional standard by reducing cyclomatic complexities. This minimized bugs, prevented code revisits, and improved maintainability.

Improving the cognitive detection accuracy from ~90% to 99.9% proved to increase the difficulty by an order of magnitude. Working with these technologies improved fundamental understanding of Computer Science. Computers or at least, software is generally deterministic. Adding a human element added a whole new layer of potential non-deterministic problems. EEG signals come from the brain, and sometimes humans can make mistakes even if they are unaware of doing so, causing difficulties in high-level design choices. In rare cases, there were anomalies in readings that never occurred again and were impossible to reproduce, some were possibly related to signal or noise as determined by the logging system. These types of problems didn't occur often but with the range of hardware being used, the potential problems were at times overwhelming. Even with the mentioned complexities the applications detection remained accurate enough to not cause any major issues.

Other issues surrounding how to help users become accustomed to BCI have not been approached. As defined in ISO 9126, "Learnability is the amount of effort that is required to learn how to use the application" (ISO, 2001). There are limitations and issues with how a user can use the application. It is difficult to just start using the application, conscious thoughts are difficult to purposely produce, and trying to explain to a user how to do this is problematic. Using conscious thoughts in this manner is not a natural thing that people are accustomed to doing, especially when it's acting as a controller for a robot. This problem was also highlighted in the research stages of this project. Improvements in this area have not been explored due to time constraints. It was considered that using a game-like tutorial could improve user's abilities in this area.

The Emotiv EPOC SDK had less support than originally hoped. The documentation for the Java and C++ SDK was used when information was missing for C#. This required modification in order for certain libraries to work as intended, wasting precious time. Time management could have been improved in this area, as time was wasted getting insignificant additions implemented.

In order for the application to remain focused, certain deviations were made. Specific application feature changes can be read in the "7.2 Justification and problems encountered" section of the project.

The end application matched the conceptual architecture and UI designs quite well, while small additional features were added in the development stages which improved the user experience. More planning in the UI design stages would have increased awareness of other required features. Most issues were overcome with further research or through perseverance when unexpected behaviors were misunderstood.

If the application could be repeated, further research would be conducted into improving cognitive detection. More time would be allocated to further expand the robot move set. Using the skills and knowledge gained from this project, the application would be much easier to port to other devices.

8.2 Future considerations

The following considerations are outside of the scope of the current project, instead offer an insight into future additional functionalities that could be applied to improve the developed application.

8.2.1 User interface

- Battery indicator

The UI currently has a signal indicator to keep the user notified about the EPOC headset connection strength. Using an extra API call the battery information could also be displayed. This simple addition can prevent issues that are related to low battery power.

- Cognitive Training

When a user profile is created or modified an improved training interface would improve, usability of the application and on-demand conscious thought skills. Using a game-like tutorial, by simply displaying on screen instructions would allow practise of cognitive actions before they are recorded, resulting in an improved recorded data set.

- EEG display

More live data related to the current processing could be displayed, resulting in a better user experience. Emotive and affective data could support a rollback feature, simply by dragging the chart timeline the user could see past information. Live raw electroencephalogram data could be added with an additional purchase of the Emotiv raw data format license. A recording feature could let users save a session, this could be shared and viewed by others. If the application supported these features then diagnoses of epilepsy and other health conditions could be performed remotely. These features would extend the functionality and scope of the application but are important for future considerations.

- Cross-platform

Developing a Cross-platform solution which targets all devices would allow the application have a much wider applicable scope. Java is portable and syntactically similar to C#. The current application could easily be converted to work using the Swing framework. These changes would provide a cross-platform solution without much deviation from the current architecture. Lightweight Linux distributions or UNIX based operating systems, such as BSD could have command-line interface (CLI) support. A web based solution would also provide support for multiple devices. Apache Cordova using the AngularJS framework allows standard web technologies to be packaged into hybrid applications, providing a mobile “app” for Android and iOS. By extending the platforms the application can be executed on, the range of uses for the application also extends.

- Accessibility

The application currently does not support accessibility options. Improvements in this area will allow people who are partially sighted, colour blind or have other impartments to use the application. Human-computer interaction (HCI) methods in these areas will need to be investigated in order to properly tailor the application for people with these issues. The project application has been demonstrated to Barnsley hospital, by implementing the future considerations, the range of users who could use this technology for communication expands.

- Customization

Improved customization to the application could be added. Simple but standardised settings which are expected from enterprise grade software. The power for cognitive detection currently, is manually set, the application could use reinforcement learning techniques to understand how the user’s brain works. Similar techniques are applied in the Google AlphaGO project (Silver et al., 2016). Other simpler improvements could be to allow the user to choose the directory, file name and auto saving behaviour of the logging feature.

- User notifications

Using user notifications when important activities occur would bring improvements to the application. Non obtrusive notifications that alert the user when the battery is getting low or messages about high noise levels would improve the user experience.

8.2.2 Robot control

- Zumo Arduino

The current zumo code takes advantage of pointers using a single algorithm to calculate the speeds but could be expanded to support classes. Another improvement could be to set the motor speeds based on the power of the cognitive action, so a higher power will result in a faster speed.

- Communication

Further investigation into an alternate communication method would allow for a wider range of robots to be controlled. Controlling a robot remotely using web technologies, like socket connections would allow the user to control a robot located anywhere in the world. A webcam live streaming the video using UDP would provide the user with the latest visual information. Aside from standard networking protocols, the RFB and ICA protocol offer useful benefits for Cloud WAN remote sessions. Further investigation could be carried out to create a custom network protocol optimizing communication in order to minimize latency for advanced control.

- Support

A wider variety of robots including drones control could be established in future projects. Being able to simultaneously control a fleet of drones via the brain which streams video back to the user almost seems like science fiction, however, the findings from this project prove it is feasible.

- Movement

This project found the cognitive actions detected by the BCI to be limited, but how these actions are used could be vastly expanded. A particular thought could activate a welcome message, or make the robot perform a comedic dance.

8.2.3 Process changes

The main problem highlighted in this project stemmed from detection issues with EEG. It was shown that the physical device does not receive the most precise information, however, approaching computation from a different architectural standpoint could improve the recognition. Alternative areas to offload computation to the GPU or by using distributed computing techniques could dramatically improve the accuracy of the EPOC recognition. It would not improve the data that the headset sends but instead improve the cognitive recognition for the application.

8.2.4 Post-WIMP

In human-computer interaction (HCI) the WIMP “windows, icons, menus, pointer” paradigm is archaic, it originated at Xerox PARC in 1973 (Dam, 1997). WIMP is predominantly still used in many computer devices today. The research and experience gained from developing a BCI application makes it obvious that an application needs to feel native to the device being used. A standard windows-like application does not best suit the functionalities of a BCI. Future research is required to create a new natural HCI interface paradigm. A brain-computer interface could be one part of future devices, coupled with virtual reality the potential for new BCI interfaces expands tremendously.

8.2.5 Techniques

This section looks at the alternative techniques that could be employed when reading electroencephalography.

- Decision filtering

One of the main difficulties in the project was differentiating similar EEG signals. (Thobbi et al., 2010) offers a potentially good solution, called "decision filtering". They state that the signals directly from the EPOC SDK are not optimized and can give false positives. They used a heuristic technique to reduce the incorrect actions. By considering the power of a particular action over a period of time, false negatives can be negated. This is likely to delay the robots movements but would have a much higher accuracy of detection. The authors state "For the future works investigation needs to be done to derive optimal algorithms for pattern recognition in EEG signals for robot motion control".

- Error-related negativity

By using error-related negativity (ERN) it is possible to detect that a user is conscious of a mistake and undo the previous movement (Ferrez, 2008). This type of undo is best suited for games, where a player can "rewind", but this could be applied to robot control. A robot could move only after no ERN is detected, although this would result in delayed movement. Alternatively a similar "rewind" function could also be applied to a robot, it would simply revert to the previous position before the error was detected.

8.3 Ethical issues

This section looks into the ethical issues that surround the technologies and techniques used in this project. Not all of the issues directly relate to the developed application, but topics of importance have been raised. The Emotiv EPOC + headset has access to a limited amount of EEG information, and most of the issues raised in the following points are undetectable without further technological purchases, a more advanced BCI device, or medical neuroscience expertise to understand the data in detail.

Initial discussions with a frontal lobe brain injury survivor found that, in their particular case, conscious thoughts are much more difficult to control and they are easily fatigued as a result. A study by (Glannon, 2014) found that "Ordinarily, motor skills are performed unconsciously and automatically following an initial period of conscious attention and learning. For those with severe paralysis, however, sustained attention is required both while being trained to operate the interface and effectively operating it to execute motor tasks. Subjects whose cognitive capacity for planning has been impaired by injury to the central nervous system may have difficulty in translating their thoughts into actions or fail to do so. Failure to meet the expectation to produce certain actions may cause distress and psychological harm."

Similar EEG event-related potential techniques can be used in radically different ways. Brain fingerprinting is used to demonstrate knowledge or facts about a crime or other information. A court case in Iowa, USA used this technique to reverse a murder conviction (Illes and Racine, 2005). An EEG-based BCI could be used for different reasons than what the users expects. To the user, they could be controlling a robot on a screen, but information on how they make informed decisions could be used with malevolent intentions. Affective information when using the EPOC headset is collected, the user's happiness or frustration levels are measured and the applications uses of this data might not be inherently obvious to the user.

Self-proclaimed non-invasive BCI devices may not have undergone extensive testing and regulation checks, potentially causing discomfort or unknown effects to the user.

As BCI and EEG technology continues to improve, the range of information that can be collected about the user expands and ethical implications arise. Potential invasion of privacy inherent in the possibility that a BCI might be used to obtain information without the users consent (Wolpaw et al., 2006).

The EEG data that can be collected and interpreted by the Emotiv EPOC + is limited. It is a non-invasive off the shelf research product, which complies with Regulations in Europe (**APPENDIX Q**). Due to some potential ethical issues the decision for the developer to solely demonstrate the deliverable was made. The device does work with other users, however, a consent form needs to be created and signed. Enough time for the user to be accustomed to using the device will be required.

8.4 Summary

Issues raised over specific processes and future considerations for improvement have been established. Multiple sources of research have been used to gain a solid understanding of the domain areas. Difficulties with EEG cognitive detection and other features that were missed in the design stages have been elaborated on.

As a first time project, attempts have been made to gain a quality understanding of the broad topics required to deliver a prototype application. The main goals originally devised have been met, while improvements could still be made in the detection of cognitive actions.

Chapter 9 – Personal and Professional development

The project as a whole has been immensely efficacious, contributing greatly towards improving personal and professional skills.

The technology developed in this project has already been demonstrated to Barnsley hospital, opening up a whole new category of assistive technology that can change people's lives. People with conditions such as locked-in syndrome are now, for the first time able to communicate with their families by using this kind of technology. Further proving the feasibility of the project, the necessity for improving research in this area and the demand for such types of devices. Working with Barnsley Hospital proves the increase in professional opportunities that arises from perusing this project idea. Discussions did take place which could have pivoted the project into focusing on assistive technology uses, but the main project aim remained. Future developments in this area will take place outside of this project.

A large amount of knowledge has been acquired in multiple subject areas. This will prove essential for future job prospects or post graduate study. The exact skills gained do not necessarily need to be transferred over to future work. Perusing this type of project demonstrates the capacity to independently research and create a prototype application in a subject area with little experience.

Every fortnight a meeting with a senior professor who supervised the project took place. Discussing project plans and updates on the latest endeavours helped keep the project managed in a timely manner. Support was also given, by loaning equipment required to complete the project. The workload was generally spread out well and sensible goals were set based on the time available.

Ethical issues have been researched and the impact they have on the project has been sensibly considered. A full list of ethical issues raised and how they influenced decisions can be found in Chapter 8.3.

This project will be a welcome addition to a growing personal portfolio, it will be displayed on a professional looking website and will be available publically as an open-source project allowing the community to expand upon its findings.

Chapter 10 – Conclusion

This investigation has provided a comprehensive look into the feasibility of using electroencephalography (EEG) to control a robot via brain-computer interface (BCI).

The research carried out discovered many BCI devices utilise a range of techniques to read thoughts from the brain, and are not limited to only using EEG. All of these areas were not explored in detail but investigation proved that robot control through a brain-computer interface was possible.

The knowledge gained from the research was used to develop a prototype application. The prototype application followed a software design and architecture that made the development conceivable. An Emotiv EPOC + headset was used with its SDK to collect data from the brain. Event-related potential (ERP) techniques were employed to make it possible to match thoughts with distinct cognitive actions. Using the cognitive actions that were detected commands are issued and a robot's movements are manipulated in *real-time*. A test-driven agile approach was used throughout development to ensure the features in the application were robust.

Research and development found many issues with using EEG to control a robot. Many of these lie with the quality of BCI devices and the accuracy of understanding the EEG data. The quality of the EEG is not the same as expensive medical equipment or invasive devices. As a result, it was found that EEG can be used to control a robot, although a high precision of control is not possible with current technology.

The software developed for this project has been demonstrated at Barnsley hospital in the UK to show the potential uses of how a device can aid patients in their everyday life. This further demonstrates the viability of the project and research area.

Despite the challenges, a working prototype application was developed and this project lays down the foundations for future development in this area, showing that this technology holds great promise.

References

AldebaranRobotics.com. (no date). Cool robots. [online] Available at: https://www.aldebaran.com/en/cool-robots [Accessed 14 Nov. 2015].

Allison, B.Z. and Neuper, C. (2010). Could anyone use a BCI?. In Brain-computer interfaces (pp. 35-54). Springer, London.

Amelan, R. (1996). Software testing blamed for Ariane failure. Nature, 382(6590), pp.386-386.

Arduino.com. (no date). Arduino - Reference. [online] Available at: https://www.arduino.cc/en/Reference/HomePage [Accessed 2 Apr. 2016].

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R. and Kern, J. (2001). The agile manifesto.

Bellingham, J. and Rajan, K. (2007). Robotics in Remote and Hostile Environments. Science, 318(5853), pp.1098-1102.

Blain, L. (2009). Honda's Brain-Machine Interface: controlling robots by thoughts alone. [online] Gizmag. Available at: http://www.gizmag.com/honda-asimo-brain-machine-interface-mind-control/11379/ [Accessed 2 Apr. 2016].

Blankertz, B., Tangermann, M., Vidaurre, C., Fazli, S., Sannelli, C., Haufe, S., Maeder, C., Ramsey, L., Sturm, I., Curio, G. and Müller, K. (2010). The Berlin Brain–Computer Interface: Non-Medical Uses of BCI Technology. Front. Neurosci., 4.

Bock, G. and Marsh, J. (1993). Experimental and theoretical studies of consciousness. Chichester: Wiley, p.126.

Boehm, B. (1984). Software Engineering Economics. IEEE Transactions on Software Engineering, SE-10(1), pp.4-21.

Bogue, R. (2010). Brain ‐ computer interfaces: control by thought. Industrial Robot, 37(2), pp.126-132.

Bulling, A., Ward, J., Gellersen, H. and Tröster, G. (2011). Eye Movement Analysis for Activity Recognition Using Electrooculography. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(4), pp.741-753.

Chapin, T. and Russell-Chapin, L. (2013). Neurotherapy and neurofeedback: Brain-Based Treatment for Psychological and Behavioral Problems. New York, Routledge.

Ciortea, L., Zamfir, C., Bucur, S., Chipounov, V. and Candea, G. (2010). Cloud9: a software testing service. ACM SIGOPS Operating Systems Review, 43(4), pp.5-10.

Coenen, A. and Zayachkivska, O. (2013). Adolf Beck: A pioneer in electroencephalography in between Richard Caton and Hans Berger. Advances in Cognitive Psychology, 9(4), pp.216-221.

Dijkstra, E. W. (1970). Notes on structured programming.

Duvinage, M., Castermans, T., Petieau, M., Hoellinger, T., Cheron, G. and Dutoit, T. (2013). Performance of the Emotiv Epoc headset for P300-based applications. BioMedical Engineering OnLine, 12(1), p.56.

Ekanayake, H. (2010). P300 and Emotiv EPOC: Does Emotiv EPOC capture real EEG?. Web publication http://neurofeedback. Visaduma.

Emotiv.com. (no date). Does EMOTIV really measure signals from my brain?. [online] Available at: https://emotiv.zendesk.com/hc/en-us/articles/200782309-Does-EMOTIV-really-measure-signals-from-my-brain- [Accessed 1 Apr. 2016].

EngineeredArts.co.uk. (no date). RoboThespian | Engineered Arts Ltd. [online] Available at: https://www.engineeredarts.co.uk/robothespian/ [Accessed 15 Feb. 2016].

EpilepsySociety.org.uk. (2015). EEG (electroencephalogram). [online] Available at: http://www.epilepsysociety.org.uk/eeg-electroencephalogram [Accessed 16 Jan. 2016].

Ferrez, P.W. (2008). Error-related EEG potentials generated during simulated brain–computer interaction. Biomedical Engineering, IEEE Transactions on, 55(3), pp.923-929.

FirstLegoLeague.org. (no date). FIRST LEGO League |. [online] Available at: http://www.firstlegoleague.org/ [Accessed 29 Mar. 2016].

Folgieri, R., Bergomi, M.G. and Castellani, S. (2014). EEG-Based Brain-Computer Interface for Emotional Involvement in Games Through Music. In Digital Da Vinci (pp. 205-236). Springer New York.

Freeman, A. and Sanderson, S. (2012). Pro ASP.NET MVC 4. Dordrecht: Springer.

Git-Scm.com. (no date). Git - About Version Control. [online] Available at: https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control [Accessed 1 Apr. 2016].

Glannon, W. (2014). Ethical issues with brain-computer interfaces. Front. Syst. Neurosci., 8.

Grierson, M. and Kiefer, C. (2011). Better brain interfacing for the masses: progress in event-related potential detection using commercial brain computer interfaces. In CHI'11 Extended Abstracts on Human Factors in Computing Systems (pp. 1681-1686). ACM.

Groover, M. (2007). Automation, production systems, and computer integrated manufacturing. 3rd ed., Englewood Cliffs, N.J.: Prentice-Hall.

Guneysu, A. and Akin, H.L. (2013). An SSVEP based BCI to control a humanoid robot by using portable EEG device. In Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE (pp. 6905-6908). IEEE.

Herr, H. (2014). The new bionics that let us run, climb and dance. [online] Ted.com. Available at: https://www.ted.com/talks/hugh_herr_the_new_bionics_that_let_us_run_climb_and_dance [Accessed 20 Mar. 2016].

Illes, J. and Racine, E. (2005). Imaging or Imagining? A Neuroethics Challenge Informed by Genetics. The American Journal of Bioethics, 5(2), pp.5-18.

Invest Hong Kong (2016). StartmeupHK Venture Forum - Elon Musk on Entrepreneurship and Innovation. [online]. Presented by Kristie Lu Stout.
https://www.youtube.com/watch?v=pIRqB5iqWA8&feature=youtu.be&t=37m20s

ISO, ISO 9166. (2001). 9126 Software product evaluation–quality characteristics and guidelines for their use. ISO/IEC Standard, 9126.

Jacks, A. (2003). Spontaneous retinal venous pulsation: aetiology and significance. Journal of Neurology, Neurosurgery & Psychiatry, 74(1), pp.7-9.

Kaku, M. (2014). The future of the mind: The Scientific Quest to Understand, Enhance and Empower the Mind. New York, DoubleDay.

Kamen, G. and Graham, E. (2004). Research methods in Biomechanics. Electromyographic kinesiology. Champaign, Illinois.

Kandel, E., Schwartz, J. and Jessell, T. (2000). Principles of neural science. 4th ed., New York: McGraw-Hill, Health Professions Division.

Le, T. (2010). A headset that reads your brainwaves. [online] Ted.com. Available at: https://www.ted.com/talks/tan_le_a_headset_that_reads_your_brainwaves [Accessed 1 Apr. 2016].

Loudin, J., Simanovskii, D., Vijayraghavan, K., Sramek, C., Butterwick, A., Huie, P., McLean, G. and Palanker, D. (2007). Optoelectronic retinal prosthesis: system design and performance. J. Neural Eng., 4(1), pp.S72-S84.

McCloy, R. and Stone, R. (2001). Virtual reality in surgery. British Medical Journal, 323(7318), p.912.

McFarland, D. and Wolpaw, J. (2011). Brain-computer interfaces for communication and control. Communications of the ACM, 54(5), p.60.

Mohammed, S., Moreno, J., Kong, K. and Amirat, Y. (2015). Intelligent assistive robots. Springer

MSDN.com. (no date). Task Parallel Library (TPL). [online] Available at: https://msdn.microsoft.com/en-us/library/dd460717(v=vs.110).aspx [Accessed 1 Apr. 2016].

Myers, G., Sandler, C. and Badgett, T. (2012). The art of software testing. Hoboken, N.J.: John Wiley & Sons.

Nath, V. and Levinson, S. (2014). Autonomous robotics and deep learning. Munich, Springer.

Nest.com. (2016). Home. [online] Available at: https://nest.com/uk/ [Accessed 19 Mar. 2016].

NHS.uk. (2015). Electroencephalogram (EEG) - NHS Choices. [online] Available at: http://www.nhs.uk/conditions/eeg [Accessed 1 Apr. 2016].

Niedermeyer, E., Schomer, D. and Lopes da Silva, F. (2011). Niedermeyer's electroencephalography. Philadelphia: Wolters Kluwer/Lippincott Williams & Wilkins Health.

Nijholt, A., Bos, D. and Reuderink, B. (2009). Turning shortcomings into challenges: Brain–computer interfaces for games. Entertainment Computing, 1(2), pp.85-94.

PwC.com. (no date). The new hire: How a new generation of robots is transforming manufacturing. [online] Available at: http://www.pwc.com/us/en/industrial-products/next-manufacturing/robotic-trends-changing-manufacturing.html [Accessed 24 Mar. 2016].

RCP-Vision.com. (no date). TDD & Continuous Integration con Jenkins. [online] Available at: http://www.rcp-vision.com/solutions/tdd-and-contiuous-integration-with-jenkins/?lang=en [Accessed 1 Apr. 2016].

Rebolledo-Mendez, G., Dunwell, I., Martínez-Mirón, E.A., Vargas-Cerdán, M.D., De Freitas, S., Liarokapis, F. and García-Gaona, A.R., 2009. Assessing neurosky's usability to detect attention levels in an assessment exercise. In Human-Computer Interaction. New Trends (pp. 149-158). Springer Berlin Heidelberg.

Research.Microsoft.com. (no date). Computational User Experiences: Brain-Computer Interfaces. [online] Available at: http://research.microsoft.com/en-us/um/redmond/groups/cue/bci/ [Accessed 17 Mar. 2016].

Sarwar, S., Muhammd, M., Shahzad, S. and Ahmad, I. (2013). Cyclomatic complexity: The nesting problem. In Digital Information Management (ICDIM), 2013 Eighth International Conference on (pp. 274-279). IEEE.

Schalk, G., McFarland, D., Hinterberger, T., Birbaumer, N. and Wolpaw, J. (2004). BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. IEEE Transactions on Biomedical Engineering, 51(6), pp.1034-1043.

Shi, W. (2015). Recent Advances of Sensors for Assistive Technologies. JCC, 03(05), pp.80-87.

Shinjuku-robot.com. (no date). ロボットレストラン. [online] Available at: http://www.shinjuku-robot.com/pc/ [Accessed 30 Mar. 2016].

Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. and Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), pp.484-489.

Sommerville, I. (2000). Software engineering. Harlow, England: Addison-Wesley.

Taha, M. (2012). Cebit's pole-dancing droids and other new technologies - BBC News. [online] Available at: http://www.bbc.co.uk/news/technology-17302657 [Accessed 29 Mar. 2016].

Telecommunications: Glossary of Telecommunication Terms. (1998). Choice Reviews Online, 35(07), pp.35-3647-35-3647.

Teplan, M. (2002). Fundamentals of EEG measurement. Measurement science review, 2(2), pp.1-11.

TheRoboticsChallenge.org. (no date). Home | DRC Finals. [online] Available at: http://www.theroboticschallenge.org/ [Accessed 29 Mar. 2016].

Thobbi, A., Kadam, R. and Sheng, W. (2010). Achieving remote presence using a humanoid robot controlled by a non-invasive BCI device. International Journal on Artificial Intelligence and Machine Learning, 10, pp.41-45.

Thorsteinson, T.J. and Withrow, S., 2009. Does unconscious thought outperform conscious thought on complex decisions? A further examination. Judgment and Decision Making, 4(3), p.235

Van Dam, A. (1997). Post-WIMP user interfaces. Communications of the ACM, 40(2), pp.63-67.

Vaughan, T., Mcfarland, D., Schalk, G., Sarnacki, W., Krusienski, D., Sellers, E. and Wolpaw, J. (2006). The Wadsworth BCI Research and Development Program: At Home With BCI. IEEE Trans. Neural Syst. Rehabil. Eng., 14(2), pp.229-233.

Vidaurre, C. and Blankertz, B. (2009). Towards a Cure for BCI Illiteracy. Brain Topogr, 23(2), pp.194-198.

Waibel, M. (2011). BrainDriver: A Mind Controlled Car. [online] Available at: http://spectrum.ieee.org/automaton/transportation/human-factors/braindriver-a-mind-controlled-car [Accessed 23 Mar. 2016].

Weiser, M. (2002). The computer for the 21st Century. IEEE Pervasive Comput., 1(1), pp.19-25.

Wolpaw, J., Loeb, G., Allison, B., Donchin, E., Do Nascimento, O., Heetderks, W., Nijboer, F., Shain, W. and Turner, J. (2006). BCI Meeting 2005—Workshop on Signals and Recording Methods. IEEE Trans. Neural Syst. Rehabil. Eng., 14(2), pp.138-141.

Wójcik, G., Wierzgała, P. and Gajos, A. (2015). Evaluation of Emotiv EEG neuroheadset. Bio-Algorithms and Med-Systems, 11(4).

Yanagisawa, K., Asaka, K., Sawai, H., Tsunashima, H., Nagaoka, T., Tsujii, T. and Sakatani, K. (2010). Brain-computer interface using near-infrared spectroscopy for rehabilitation. In Control Automation and Systems (ICCAS), 2010 International Conference on (pp. 2248-2253). IEEE.

Zanzotto, F.M. and Croce, D. (2010). Comparing EEG/ERP-like and fMRI-like techniques for reading machine thoughts. In Brain Informatics (pp. 133-144). Springer Berlin Heidelberg.

Zhou, S., Xi, J., McDaniel, M., Nishihata, T., Salesses, P. and Iagnemma, K. (2012). Self-supervised learning to visually detect terrain surfaces for autonomous robots operating in forested terrain. Journal of Field Robotics, 29(2), pp.277-297.

Glossary

API – Application Programming Interface

BCI – Brain-computer Interface

CLI – Command-line Interface

EEG – Electroencephalography

EMG – Electromyography

EOG – Electrooculography

ERN – Error-related Negativity

ERP – Event-related Potential

GUID – Globally Unique Identifier

HCI – Human-computer Interaction

ICA – Independent Computing Architecture

IDE – Integrated Development Environment

MRI – Magnetic Resonance Imaging

NIRS – Near-infrared Spectroscopy

RFB – Remote Framebuffer

SDK – Software Development Kit

TDD – Test-driven Development

UDP – User Datagram Protocol

WAN – Wide Area Network

WCF – Windows Communication Foundation

WIMP – Windows Icons Menus Pointer

XML – Extensible Markup Language

Appendix A – Project Specification

PROJECT SPECIFICATION - SEGM 2015/16

Student:	**Shaun Webb**
Date:	**10/10/2015**
Supervisor:	**Peter Collingwood**
Degree Course:	**BEng Software Engineering**
Title of Project:	**Real-time robotics control using Electroencephalography**

Elaboration

The aim of this project is to use measured voltage fluctuations from within the neurons of the brain to manipulate the movements of a robot in real-time. The data will be collected using an Emotiv EPOC+ headset [already available] which will provide EEG data that can be interpreted programmatically. This project spans over multiple disciplines, such as Neuroscience but the primary focus is on the Software Engineering/computational aspects and the practical uses of this application.

The project was motivated by a visit to Miraikan (The National Museum of Emerging Science and Innovation) in Tokyo, Japan where the potential for robots such as ASIMO to be controlled in more natural and intuitive ways was demonstrated. The project will prove this is already possible by designing and developing bespoke software that will use commercial devices APIs to interoperate together. Learning about software interoperability in the process and creating a software deliverable that will work with sophisticated devices.

The main objective of this project is to demonstrate that brain–computer interfaces (BCI) can be used to control real world entities. Every 10-15 years a new computing platform emerges, from mainframes to PCs, to web and then mobile. The next evolutionary step in computing could be immersive devices that we will use in our everyday lives. Devices connected to our vision will be likely thrive initially, with products like Google Glass, Oculus Rift and Microsoft HoloLens driving innovation. Then after the indoctrination period people will be more accustomed to these immersive devices and BCI augmentation will eventually be used in most aspects of life.

This project will also demonstrate organization and time managing skills as realistic deadlines will be set throughout. Technical skills will be improved dramatically as multiple programming languages will be used. Ways to understand APIs that will likely be implemented using completely different techniques will have to be integrated.

Project Aims

The aims of this project are as follows:

- Gain an understanding of Electroencephalography and how it can be used with computers.
- Use the Emotiv EPOC and API to successfully register thoughts and learn how the included software works.
- Control a robot such as the Aldebaran Nao in real-time using the provided API.
- Link together multiple APIs which will allow a robot to be controlled using the Emotiv EPOC.
- Design, develop and test a deliverable which demonstrates that BCI can be used to control another device.
- Learn more about robots and how to make multiple devices work in parallel.
- Devise and create a Project Report which will also aid the development of the deliverable. Critically evaluate the work that has been completed. Reflect on what was learnt and what could have been done differently.

Project deliverable(s)

The project deliverable will demonstrate that a robot can be controlled via thoughts from EEG readings using the software I created and by taking advantage of the available APIs and SDKs of each device.

Action plan

Report	
Task	Completed by
Initial background investigation	03/11/2015
Understand scope of the report	17/11/2015
Get information review checked by tutor before submission	01/12/2015
Ensure required topics are understood for the written report.	15/12/2015
Finalise project report	29/03/2016

Deliverable	
Task	Completed by

Conduct research into the EPOC headset and a Robot API	17/11/2015
Successfully get EPOC API working in a programming environment based on EEG readings	01/12/2015
Get real-time control working with the robot	08/12/2015
Start developing the software to allow interoperability of APIs	15/12/2015
Demonstrate basic version of the deliverable	19/01/2016
Demonstrate deliverable	18/04/2016 – 29/04/2016

Deadlines	
Task	Completed by
Project Specification + Ethics form	23/10/2015
Information Review	04/12/2015
Provisional Contents Page	12/02/2016
TurnItIn Submission	11/04/2016
Project Report and electronic copies of deliverable.	13/04/2016
Demonstration	18/04/2016 – 29/04/2016

Ethics

See ethics form attached.

I will be the human participant for the testing and development of my project deliverable.

Appendix B – Ethics Checklist

Ethics Checklist – SEGM Final Year Project 55-6727-00L

If the answer to any question is 'yes' the issue **MUST** be discussed with your project supervisor.

Question		***Yes/No***
1.	Does the project involve human participants? This includes surveys, questionnaires, observing behaviour, testing etc.	Yes (Only myself)
2.	Does the project involve the use of live animals?	No
3.	Does the project involve an external organisation? If yes, please write the name of the organisation here:	No
4.	Does the project require access to any private or otherwise sensitive material?	No
5.	Does the project require the reproduction (beyond normal academic quotations) of materials authored by a source other than yourself?	No

Adherence to SHU policy & procedures

Declaration
I can confirm that: • I have read the Sheffield Hallam University Research Ethics Policy (available at http://www.shu.ac.uk/_assets/pdf/research-ethics-policy.pdf) • I agree to abide by its principles. Signature …S.Webb……………………………Print Name………Shaun Webb………… Date ……15/10/2015…………..

Appendix C

Honda's ASIMO being controlled using electroencephalography (EEG) and near-infrared spectroscopy (NIRS).

Appendix D

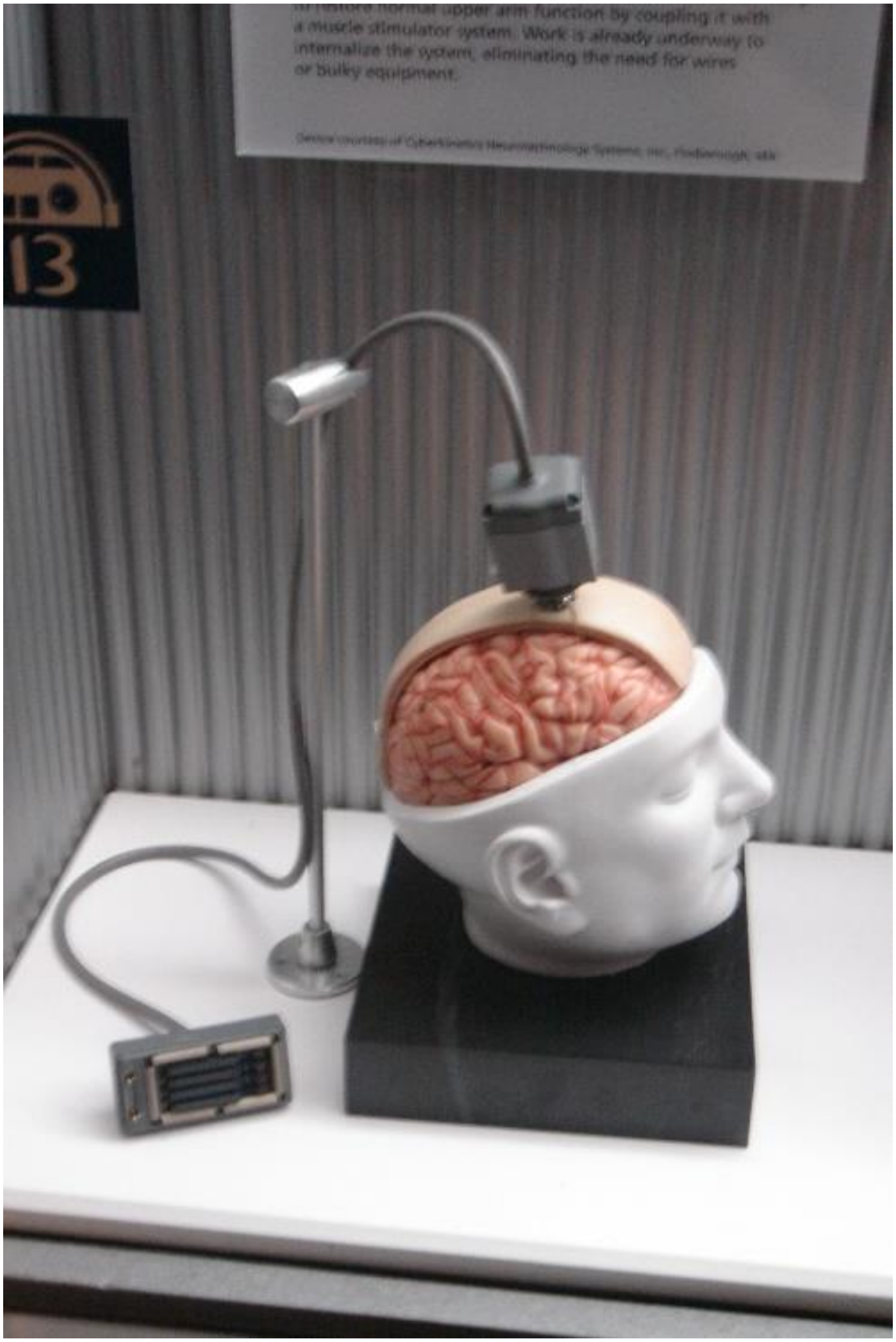
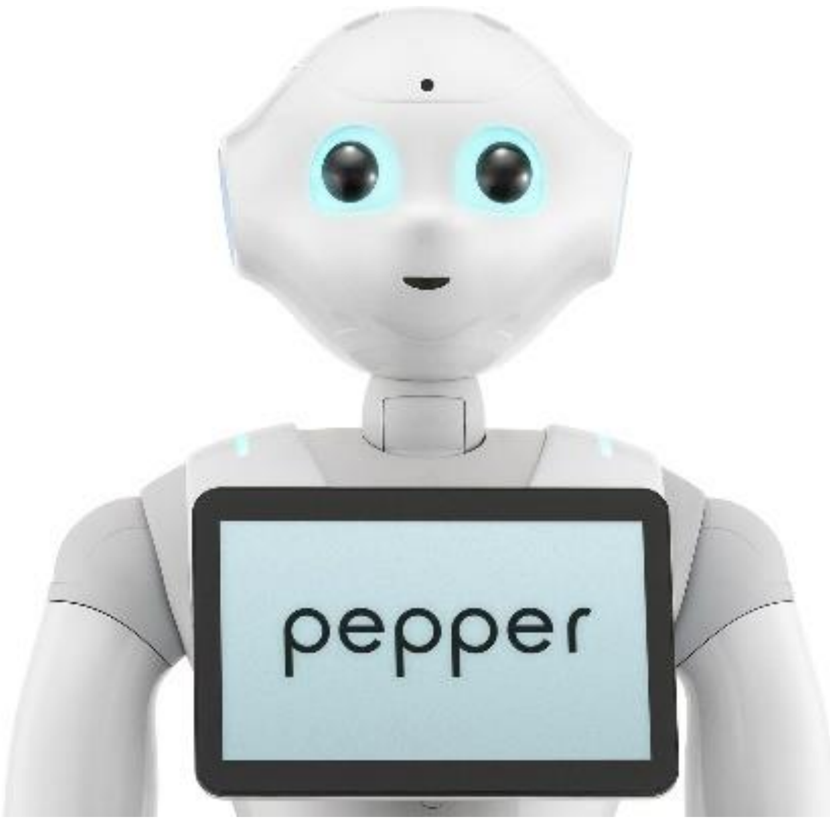


Illustration of the design of a BrainGate interface developed by Cyberkinetics.

Appendix E

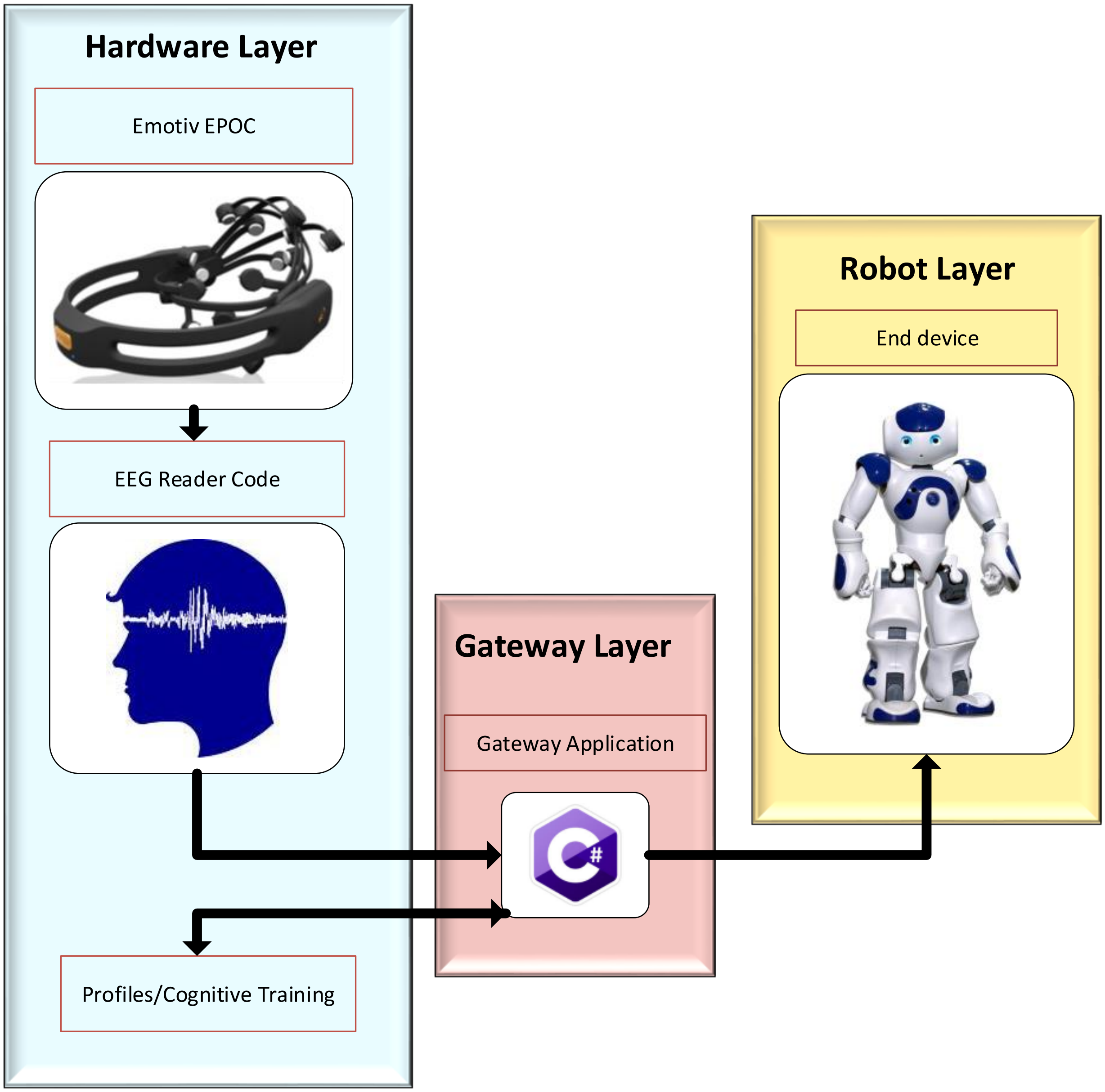


Alderbaran’s Pepper Robot.

Appendix F

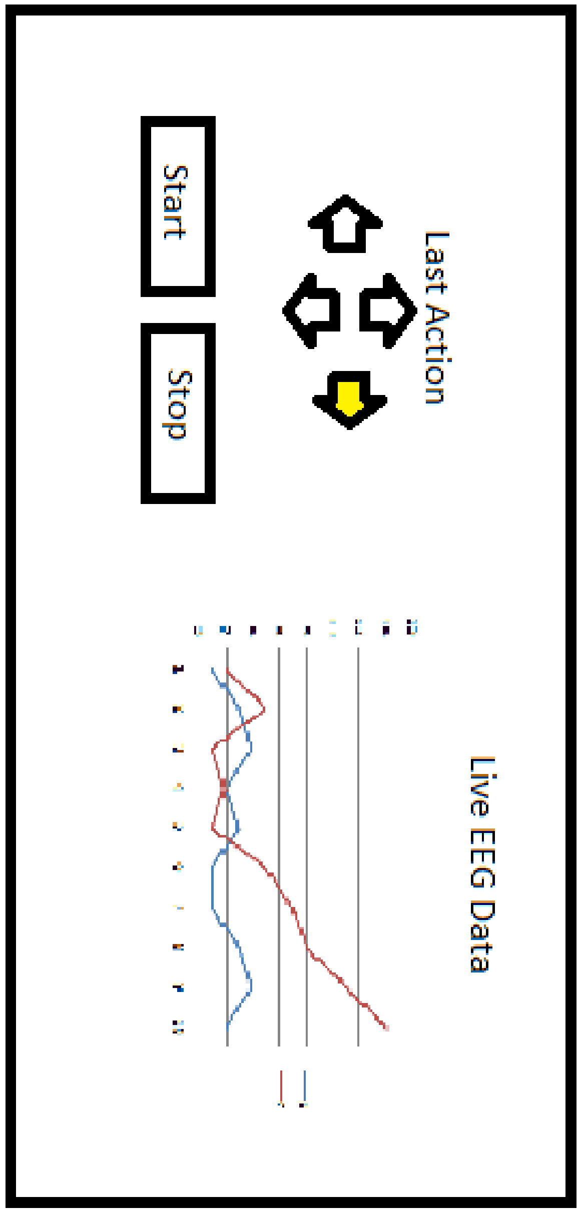
Conventional EEG system (left) and an Emotiv EEG system (right).

Appendix G



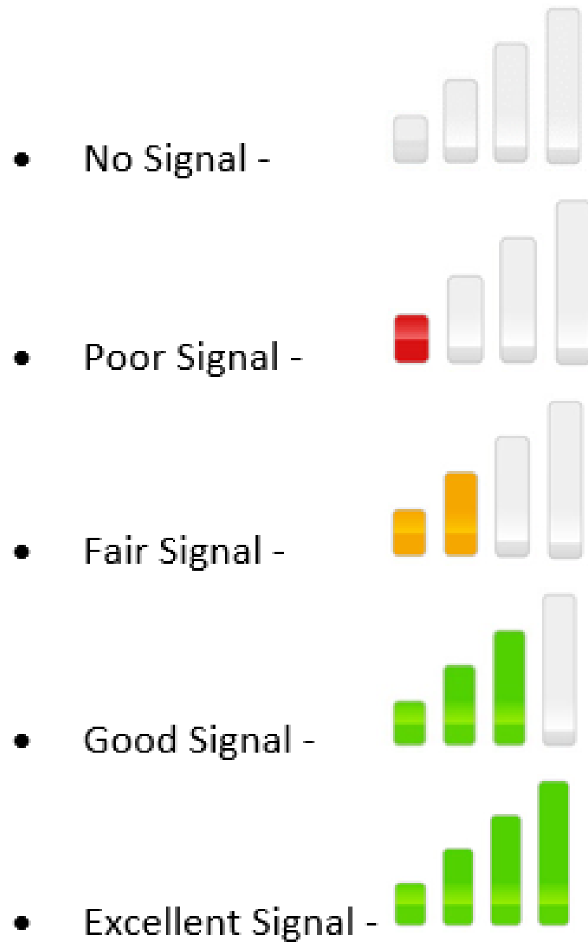
Conceptual Architecture

Appendix H



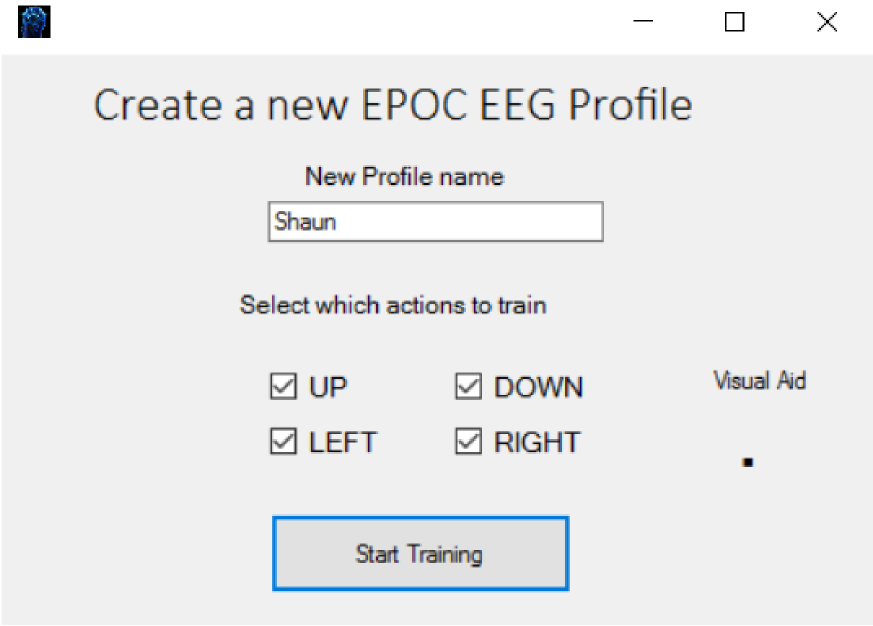
Basic UI Design

Appendix I



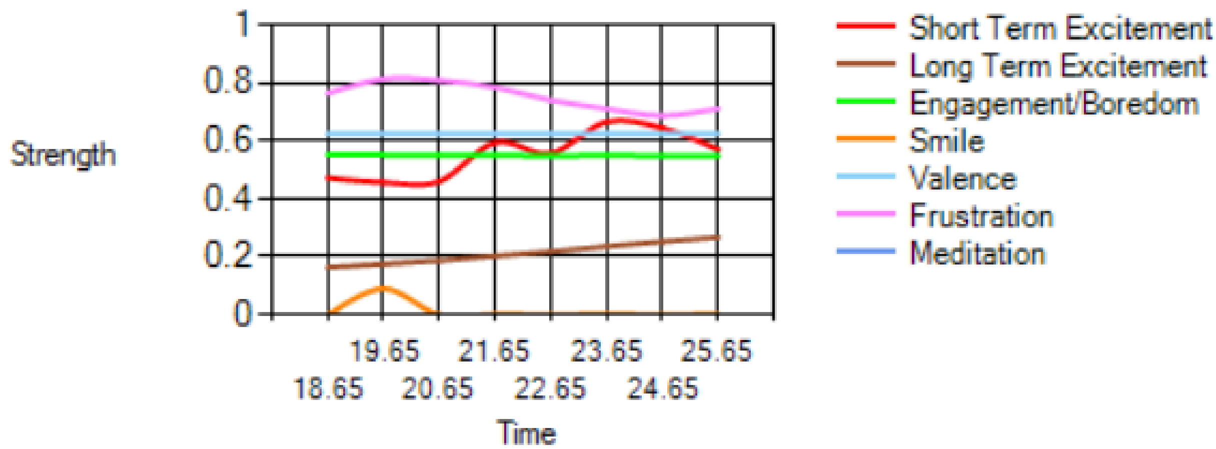
Application Signal Display.

Appendix J



Application New Profile Interface

Appendix K



Affective Data being displayed on the UI.

Appendix L

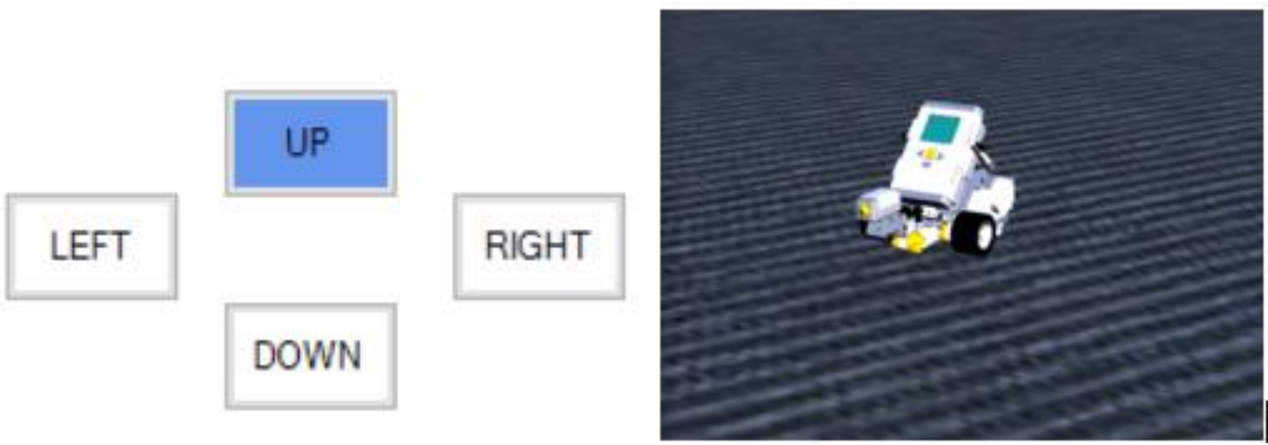
```
Log Entry : 7:27:52 PM Wednesday, March 23, 2016
Name: ShortExcitementLevel, Value: 0.0007831208
Name: LongExcitementLevel, Value: 0.0004284323
Name: FrustrationLevel, Value: 0.3647476
Name: MeditationLevel, Value: 0.3316737
Name: ValenceLevel, Value: 0.624995
Name: EngagementBoredomLevel, Value: 0.5530405
Name: SmileLevel, Value: 0
Name: Busy, Value: False
Name: SignalStrength, Value: GOOD_SIGNAL
Name: Timestamp, Value: 20.32997

-------------------------------

Log Entry : 7:27:52 PM Wednesday, March 23, 2016
COG_PULL - 0.4728535
```

Log Data Example.

Appendix M



UI direction with robot simulator.

Appendix N

Zumo Arduino Based Robot.

Appendix O

```
//-----------------------------------------------------------------------
//  Author: Shaun Webb
//  University: Sheffield Hallam University
//  Website: shaunwebb.co.uk
//  Github: TehWebby
//
//  Zumo Robot code
//-----------------------------------------------------------------------
#include <ZumoMotors.h>
#define LED_PIN 13

ZumoMotors motors;
int x;
int str;
int speed = 0;
int limit = 400;

void setup(){
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
  Serial.setTimeout(100);
  digitalWrite(LED_PIN, LOW);
}

void loop(){
  if (Serial.available()){
    str = Serial.parseInt();

    //used a custom timer when coding to increase performance
    unsigned long start = micros();

    //ensure the direction entered is valid
    if (isValid(str))
      setMotors(str);

    resetValues();

    //used a custom timer when coding to increase performance
    unsigned long end = micros();
    unsigned long delta = end - start;
    Serial.println(delta);
  }
}

void setMotors(int dir){
  //use pointer to ref speed as we can keep the existing speed if we don't change it
  int *lSpeed = &speed;
  int *rSpeed = &speed;
  if (dir == 2)//Reverse
    setReverseSpeed();
  else if (dir == 3)//Left
    lSpeed = 0;
  else if (dir == 4)//Right
    rSpeed = 0;

  //loop from 0 to 400 unless reverse is active then speed=-400,limit=0
  for (speed; speed <= limit; speed++){
    motors.setSpeeds(*lSpeed, *rSpeed);
    delay(2);
  }
  //after the movement has completed, stop the motors (set speed to 0)
  stopMotors();
}

bool isValid(int dir){
  //ensure the direction entered is valid
  if (dir > 0 && dir < 5){
    return true;
  }
```

```
  else//maybe remove else?
    return false;
}

void setReverseSpeed(){
  //set the speed to negative if command is reverse (2)
  speed = limit*-1;
  limit = 0;
}

void resetValues(){
  speed = 0;
  limit = 400;
}

void stopMotors(){
  motors.setSpeeds(0,0);
}
```

Zumo Arduino Robot code

Appendix P

Electroencephalography-Robot

Main Page | Packages | Classes | Files | Search

Class List | Class Index | Class Hierarchy | Class Members

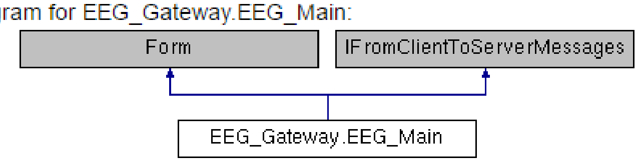
EEG_Gateway 〉 EEG_Main 〉

Public Member Functions | Static Public Member Function
Public Attributes | Protected Member Function
List of all membe

EEG_Gateway.EEG_Main Class Reference

EEG_Main Gateway Application More...

Inheritance diagram for EEG_Gateway.EEG_Main:



Public Member Functions

	EEG_Main () Is the simulator active. More...
void	**OnApplicationExit** (object sender, EventArgs e) Handles application closes More...
void	**Instance_EmoStateUpdated** (object sender, EmoStateUpdatedEventArgs e) Emotion state updated More...
void	**logEEG_Data** (EdkDll.EE_CognitivAction_t currentAction, float power, string file) Log EEG Data More...

Doxygen code documentation.

Appendix Q

Complies to the following Regulations in Europe

Europe, Australia, New Zealand

Product Name and Model: Emotiv EPOC Model 1.0

Product description: EPOC Neuroheadset, USB-01 Transceiver, Hydrator Pack + charger or charge cable conforms to the following Product Specifications and Regulations:

EMC and Telecom: Class B, ETSI EN 300 440-2 V1.4.1, EN 301 489-1, EN 301 489-3, AS/NZS CISPR22 :2009, AS/NZS 4268 :2008, FCC CFR 47 Part 15C (identifiers XUE-EPOC01, XUE-USBD01)

Safety: EN 60950-1:2006, IEC 60950-1:2005 (2nd Edition), AS/NZS 60950.1:2003 including amendments 1, 2 & 3, CB Certificate JPTUV-029914 (TUV Rheinland)

The product here complies with the requirements of the Low Voltage Directive 2006/95/EC, the EMC Directive 2004/108/EC, the R&TTE Directive 1999/5/EC, and carries the CE and C-Tick marks accordingly.