Faculty of Science and Technology

BSc (Hons) Games Programming

May 2017

# Prototyping an electronic input device for mapping a single digit joint position to a 3D environment

by

James Ronald John Slowgrove

## DISSERTATION DECLARATION

This Dissertation/Project Report is submitted in partial fulfillment of the requirements for an honours degree at Bournemouth University. I declare that this Dissertation/ Project Report is my own work and that it does not contravene any academic offence as specified in the University's regulations.

### Retention

I agree that, should the University wish to retain it for reference purposes, a copy of my Dissertation/Project Report may be held by Bournemouth University normally for a period of 3 academic years. I understand that my Dissertation/Project Report may be destroyed once the retention period has expired. I am also aware that the University does not guarantee to retain this Dissertation/Project Report for any length of time (if at all) and that I have been advised to retain a copy for my future reference.

### Confidentiality

I confirm that this Dissertation/Project Report does not contain information of a commercial or confidential nature or include personal information other than that which would normally be in the public domain unless the relevant permissions have been obtained. In particular any information which identifies a particular individual's religious or political beliefs, information relating to their health, ethnicity, criminal history or personal life has been anonymised unless permission for its publication has been granted from the person to whom it relates.

### Copyright

**The copyright for this dissertation remains with me.**

### Requests for Information

I agree that this Dissertation/Project Report may be made available as the result of a request for information under the Freedom of Information Act.

Signed:
Name: James Ronald John Slowgrove
Date:
Programme: BSc GP

## Acknowledgements

I would like to thank my primary project supervisor Andrew Watson for his assistance and help with this project.

I would like to also thank Wen Tang, my secondary project supervisor for her input during my project.

Finally I would like to thank Gary Toms for his help in researching the possible sensor components to use with the project.

## Abstract

With the advancement of Virtual Reality (VR) there has been a recent surge in the concept of immersive input devices. One form of input that has yet to be fully utilised is mapping the joint movement of a hand. While there are many devices out there, no one method of joint mapping has become standard. This project will prototype a device that is able to track a single digit joint position and send this information to a computer for use with a 3D environment. This information can then be used to evaluate its use for a whole hand input device. This has applications for both gaming and medical purposes, either in VR games or medical functions, such as assisting in stroke rehabilitation. Research and testing will be undertaken to identify different methods of tracking joint movements within the hand.

# Contents

# Figures

# Chapter 1) Introduction

## 1.1) Introduction

With the recent popularity of devices such as the Oculus Rift and the HTC Vive, virtual reality has started to become more prevalent in the mainstream. Along with this surge of virtual reality devices, the idea of immersive input devices have also seen an increase of interest.

Immersive input devices are devices that are designed to help decrease the gap between the interaction between computers and humans to help surround the user within a virtual world. Examples of this include devices such as the Microsoft Kinect, which tracks the user's body motions in the real world and uses that to control the virtual interface. One type of immersive input that has yet to be fully utilised is a mapping the joint movement of a hand, as most input devices are more focused on the general position of the hand, rather and the positions and angles of the joints of the hand. Devices that do map joint positions of a hand tend to be expensive and seen as more of a specialist piece of equipment, rather than an immersive input device.

This project is intended to research the existing methods of joint sensors in order to prototype a device that can map a single joint position to a 3D environment, to evaluate its use for a whole hand input device.

This device has applications within both gaming and medical purposes, either in VR games or medical functions, such as assisting in stroke rehabilitation, or for training medical students on procedures that would be too dangerous to practice in the real world.

This project is intended to help both the average consumer of immersive devices for use with entertainment and virtual reality, alongside potential medical institutions that are looking for cheap methods of accurately tracking joint movements in a hand. This is done by evaluating existing methods of joint tracking to try and find an effective solution to create such a device.

## 1.2) Aims and Objectives

### 1.2.1) Aims

1. To design and prototype an electronic input device that allows the joint movement of a single digit to be mapped and this data to be sent to a program.
2. To design and prototype a software program that can understand the data generated by the input device and then be replicated in a 3D virtual environment.

### 1.2.2) Objectives

1. To research the available methods of measuring joint movements and to test them in order to decide which method is the most effective for this project.
2. To prototype the electronic input device to map a single joint and convert that data into a form that can be sent to a computer,
3. To prototype an evaluation application on a computer that can receive data sent from the prototype device and translate it to an angle value of the joint.
4. To prototype an application that can receive the data generated by the input device and translate it to a 3D environment within either a custom 3D program or a pre-existing engine such as Unity.
5. To evaluate a full hand input device using the information gathered from the single joint prototype by comparing it against other similar devices that currently exist for similar purposes and potentially communicate with a client with knowledge in the field for the medical aspect of the device.

# Chapter 2) Literature Review

## 2.1) Introduction

This literature review is intended to help provide some background research and information about the subject at hand. By researching and reviewing papers that are about or are similar to the project, this literature review will help identify the key points that need to be tested when prototyping the device. It will also help to identify potential methods of measuring the joint positions of a hand and integrating them into a 3D environment along with analysing and reviewing the methods that are used to evaluate these devices.

## 2.2) Input Devices

The aim of Input devices is to help decrease the gap between the interaction between computers and humans. This interaction between humans and computers is known as Human Computer Interaction (HCI).

According to a paper from 2014, HCI “is a conventional means that provides users to interact with computers and handheld devices” **[Prasad et al. 2014]**. In this paper the project was to design a human-computer interface to interact with VLC media player. They used a wireless data glove to detect human dynamic gestures. They then used a study about human gestures along with a learning algorithm to classify the gestures. According to the paper this kind of device can be used for many other applications other than just interacting with VLC media player. Examples that they give include space stations, satellite repair, health care and biomedical surgery.

When dealing with Virtual Reality, input devices start to take on a more immersive role as immersive input devices. The aim of these immersive devices is to help surround the user within a virtual world. Devices such as the Oculus Rift and the HTC Vive allow the user to be surrounded in these virtual environments via the user’s visual and audio senses, however without immersive input devices the illusion is lost.

The HTC Vive comes with controllers, *as shown in figure x1*, to help aid in the immersive experience. This device contains 24 sensors, a multi-function trackpad, a dual-stage trigger and HD haptic feedback **[HTC Corporation 2017]**. These are all to try and help improve the immersiveness of the device, and while this works for most forms of games that require the user to hold a device, such as a gun or sword, it is still a controller and anything that is designed to play as if you are using your hands to pick up and use objects loses immersion when it is held as a controller.

*Figure 1 - The HTC Vive Controller* ***[HTC Corporation 2017]***

A paper published in 2010 about enhanced user immersive experience with a virtual reality base FPS game interface **[Yoon et al. 2010]** took a head mounted display, a head tracker, 5.1 channel headphones and a pair of data gloves and used them to control the input of Unreal tournament 2004.They then took the proposed system and tested it for usability. They used hand gestures to take over the role of the input of the game, having different motions to act for different controls. Examples of the different actions that were used for inputs are horizontal movement, running, walking, stopping and grabbing items. This resulted in a more immersive gameplay with the users agreeing that they felt more absorbed in the proposed system.

While there are many different types of immersive input devices for different purposes, this project will be focusing on the input of the user’s hand.

## 2.3) Types of Hand Input

There are two main form of methods to track hand gestures: methods which use data gloves and methods which are vision based **[Murthy & Jadon 2009]**.

Vision based methods are devices that use external camera’s to follow and track the user’s hand movement. A well-known example of this kind of input device is the Microsoft Kinect. The Microsoft Kinect consists of an RGB camera, an infrared (IR) emitter and depth sensor, a multi-array microphone and a 3-axis accelerometer, *see figure 2*. The components that the Kinect uses for tracking the user’s position are the IR emitter and the IR depth sensor. As stated on Microsoft’s specification page “The emitter emits infrared light beams and the depth sensor reads the IR beams reflected back to the sensor. The reflected beams are converted into depth information measuring the distance between an object and the sensor. This makes capturing a depth image possible.” **[Microsoft 2017]**

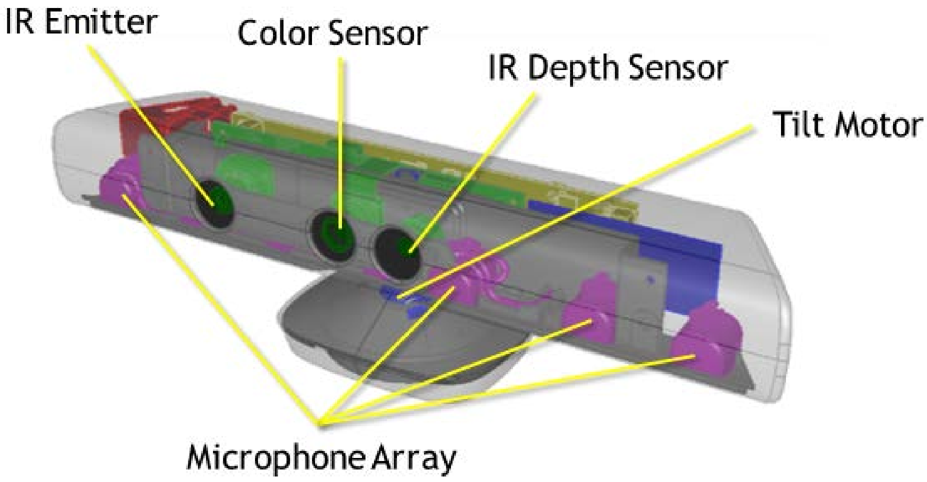


*Figure 2 - The internals of the Microsoft Kinect V1.0* ***[Microsoft 2017]***

By using the depth information that the Kinect collects it is able to identify parts of the user and track the user’s movements. Using these movements it can recognise gestures performed by the user and use that to control the software input.

Vision based methods are hard to design for generic usage and need to be in a controlled environment in order to be truly feasible **[Murthy & Jadon 2009]**. Due to this it can become problematic for the average person’s use, as they cannot dedicate an area to just using a device. As such, this project will be focusing on the other form of hand input, data gloves.

It would seem that the majority of devices made with the intention for precise hand mapping are in the form of data gloves. A data glove, *an example of which can be seen in figure 3*, consists of a glove that is worn on the user's hand. Each of the joints on the hand has a sensor on it in order to detect the angle of the joint. The gloves also often have other sensors on them in order to detect the motion and orientation of the hand. This data is then sent to a processing unit, either built in to the glove or via an external cable, which is then converted to a form that a computer can understand. This is then sent to the computer either via a cable or a wireless connection. Most data gloves tend to use a wireless connection as cables can make it difficult to freely move the user's hand, causing a lack of potential input.

*Figure 3 - An example of an existing Data Glove* ***[CyberGlove Systems 2016]***

A big problem with data gloves is that most of the devices have to fit the users hand, this means that they often require specific calibration for that user. This is because most of the devices are not built as a 'one size fits all' but are in fact designed for a specific size of hand. Interestingly enough a paper from 2013 proposed utilising a Microsoft Kinect in order to help calibrate the data gloves "in the wild" **[Vicente & Faisal 2013]**. They proposed using the Kinect to capture the touching of the fingertips and thumb in order to calibrate the data glove for the user's hands. This is an interesting method of combining the two methods of tracking hand gestures in order to create a reliable method of hand tracking.

As well as the calibration problem, data gloves are remarkably expensive normally costing from US$ 585.00 to more than US$ 1500.00 **[Pereira et al. 2013]**. This is normally due the high cost of the sensors and components used in the device.

While all of these devices contain some kind sensors in order to track the movement of the hand, no one method of sensing has yet to become the prominent method for tracking joint movement and as such there is plenty of room for experimenting.

## 2.4) Uses in Gaming

There are many types of input devices for gaming, especially since the rise of VR **[Anthes et al. 2016]**. With demand for more immersive gaming thanks to VR, the ability to perform more precise hand movements will likely become much more sought out for the gaming market.

A key part of the idea of utilising different types of input for gaming is the idea of making the experience more immersive than just using a basic controller. Devices such as the Kinect have helped make player movement part of controlling a game, however the control of the player’s hand has normally been to do to the position of the hand as a whole, rather than what the hand itself is doing.

Other examples of the use of such a device with gaming include using the tracking of the hand motion of the user in order to increase human computer interaction. A paper from 2009 about “3-D Hand Motion Tracking and Gesture Recognition Using a Data Glove” talks about a proactive computing system that can automatically recognise a user’s intention from their gestures **[Kim et al. 2009]**. This could be utilised in a gaming environment to help improve the input for the game’s actions.

A paper published in 2015 **[Lipovsky & Ferreira 2015]** talks about using wearable technology with video games in order to help hand rehabilitation. They used a “home-built robotic glove” for use with the Unity3D Engine in order to create a puzzle game aimed at hand rehabilitation by having increasingly difficult levels of puzzles, revolving around grabbing, holding, transporting and dropping a cube with their hands. While the paper states that they have only tested it with healthy participants at the time of it being written, with it needing more testing with actual patients, the system works by translating the movements of the hand to the 3D world.

Another paper that utilises Untity3D for the 3D environment was published in 2013 **[Silva et al. 2013].** This paper is about the development of a low cost data glove using an Arduino for virtual reality applications. They utilised Unity3D for use with the hand simulator. They created the simulator using simple meshes built using Blender that had been exported as .obj files for use with game engines. Initially they decided that in order to send the data to the engine, they would write the data for the glove to a file on a HTTP server every frame. A Unity script would then read this data from the file, using the information to position the objects in the scene.

Unfortunately this caused issues with instability and latency. In order to combat this they instead had a script open a COM port for use with a serial port and sent the data directly to the engine from the glove instead. As with most of the papers looked at, they conclude that with the new system they have devices that will not only work as a replacement for the current available systems, but will be considerably cheaper.

All of these papers point to the potential use that data gloves have for use within the gaming market, especially within VR. A common trait in the papers that have been looked at when researching the use of gaming with a data glove is that they are also being researched for use in conjunction with medical purposes.

## 2.5) Uses in Medicine

One of the most common research topics for the use of such a device is for medical purposes. A common example being research into a device that can measure hand movements for used with rehabilitation exercises for stroke victims.

A paper published in 2016 about A VR-based Self-rehabilitation System explains how the researchers used dry electrodes attached to subject's skin to collect signals from the subject's muscles to map the movement in a 3D OpenGL environment **[Guo et al. 2016]**. This allowed the electrodes to collect electromyography signals, which are biomedical signals, and use this data along with the 3D environment to provide enhanced visual feedback. One of the points of this paper was to help stroke patients to "realize self-rehabilitation exercise of upper limb at home". This was apparently a success, at least according to their conclusion.

Another paper researching the use of a data glove for rehabilitation of stroke victims **[Tavares et al. 2015]** talks about how "the recovery of the lost capabilities is possible throughout frequent stimulation promoted by rehabilitation exercises on a daily basis". This project does not utilise a computer to act as the user interface of the device, but instead uses a device called an HMI Module. This is a device that contains a touch-screen, which shows of the details of the exercise to do, along with logging the exercises. The glove uses a combination of angular position and pressure sensors to get the information about the action of the user's hand and they feel that device could be used for other fields of interest, including fields such as virtual or augmented reality.

A third paper on using VR with a data glove to help with stroke rehabilitation **[Jack et al. 2001]** looks into using a pre-made data glove, alongside a force feedback glove, to allow interaction within a virtual environment to perform rehabilitation routines. As with the other papers that looked in to using VR with stroke rehabilitation, the conclusion is positive, however this paper talks in much more detail about the effect that the rehabilitation system had on the three patients that took part in the study. They concluded with the opinion that "this result may be indicative of positive effects".

The three of these papers all help show that the data glove as an input device can have a large impact in improving the ways that stroke victims can help to recover using rehabilitation systems.

A similar medical use for a data glove for stroke rehabilitation, is written about in a paper published in 2014 **[Rahman et al. 2014]**. In this paper they talk about using a data glove to help with improving the physical therapy required to help people with monoplegia. Monoplegia is “type of unilateral spastic cerebral palsy”. The majority of patients are children, which is why the paper states that the therapy sessions need to be enjoyable. This is where the concept of using a data glove comes in. The data glove can serve two purposes, the first and foremost is to turn the therapy exercises into a game to help keep the children entertained, so that they are more willing to perform them. The second is to keep track of data from the glove so that it can be analysed to evaluate the effectiveness of the device. While the paper does not give any complete conclusions due to the long term type of research, they do seem confident that it will be effective.

Not all uses in medicine are just for rehabilitation however. A paper from 2016 **[Cao et al. 2016]** discusses the potential use with virtual surgery. While this was not the only focus of the paper, the other being making the glove cost effective, it does go in to detail on the use of a glove in virtual surgery. A problem that exists within training students’ surgery is that it requires either live animals or dead bodies to practice on. This means that it is impossible to repeat the processes as many times as is needed for accurate training. The paper talks about how, with the use of virtual reality, it would be possible to train without these problems if using a data glove. The problem with this is the cost of the gloves, which is the main focus of the paper, developing a cheap solution to the problem.

All of these papers help to demonstrate the potential use that data gloves have within the field of medicine, ranging from helping people to recover, to helping students in the field of medicine training to make less mistakes in reality.

## 2.6) Types of Sensors

Now the primary purpose of this project is to identify and prototype an electronic input device for mapping a single joint position to a 3D environment. In order to achieve this, an effective method of angle detection must be found.

Researching multiple papers that discuss the use of data gloves gives a large range of different types of sensors that are in use currently. Some of the possible methods include: Force Sensitive Resistors, Rotary Potentiometers, Dry Electrodes, Gyroscopes, Accelerometers and Magnetometers.

The first paper looked at was the paper that had been discussing the use of a data glove with virtual surgery **[Cao et al. 2016]**. This paper researches creating a motion capture data glove for use using Force Sensitive Resistors**.** One of the main purposes of the paper was to try and create a cheaper alternative to the current set of data gloves. They created a 3D printed external structure for the glove and attached force sensitive resistors under the joints of the glove structure. Using this they were able to calculate the angle of the joints in the glove by using the resistive value of the force sensitive resistor. According to their results the glove has a low accuracy but is within the acceptable range that they were looking for. The problem with this paper is that they have not gone into great details on how they actually tested the device. All they give is a table that stats the pitch, roll and yaw of the data. These are listed next to a list of gestures identified by a single number, without any other information as to what these gestures where. On top of that this paper does not mention anything to do with the size of the hand that they are testing with. As the device has a 3D printed aspect to the glove's external, it has to only be able to work with one size of hand. Without including this information it makes the results of the paper, stating that this would good enough for use with teaching surgery in university, less valid. However the results that they collected from a cost stand point are still valid.

The next paper that talked about Force Sensitive Resistors for use with a data glove is a paper on a data glove that uses force sensing sensors alongside 9-axis inertial measuring units (IMU's) **[Hsiao et al. 2015]**. The main focus of detecting the angle of the joints however is using the IMU's. Each IMU sensor contains a 3-axis gyroscope, a 3-axis accelerometer and a 3-axis magnetometer.

This matches another paper that was looked at, published in 2016 **[Fang et al. 2016]**. In this paper the use of a tri-axial gyroscope, accelerometer and magnetometer are used to detect the angle of the joints in the data glove, and are then analysed.

The first of these two papers **[Hsiao et al. 2015]** primary focus is using the sensors to create a flexible data glove for use with collecting useful information for physicians, along with creating a tool that can improve hand rehabilitation. They used 17 of the IMU's on the device and connected them to a microcontroller unit (MCU) on the device. This collects the data and encapsulates it in to a single packet of data. This is then sent to the connected PC wirelessly using Bluetooth, where the data is used to calculate the wanted values on the PC.

This is different to how the second paper works **[Fang et al. 2016]**. The primary focus of this paper is in creating a data glove using what they consider to be a novel approach, stating "which was few presented before to our knowledge" meaning that they had not found many other papers that had used the method. The second paper utilises only 15 sensors and the MCU processes the data before it sends the packet of data to the PC. It also uses a wired connection instead of a wireless one.

Both of these papers go into details on how the device takes the data and converts it into a readable form of data, and both of them conclude that the method of using this is not only a viable one, but that the device could be used to replace the current equivalents of the device.

Another paper looking at the development of an adjustable angle sensor **[Othman et al. 2016]** talks about using a rotary potentiometer for measuring finger flexion (bending of a joint). They decided to build a data glove that uses these potentiometers and compare the results to existing methods currently out there. They talk about the key to the potential of the potentiometer is the fact that it has a linear output. They created a basic external attachment for the gloves that allows the potentiometer to rotate along the angle of the hand joint. The results when compared to a bending sensor showed that it seemed to work just as well. However, unlike the bending sensor, the potentiometer did not degrade over time.

A very different method of sensing is talked about in the paper from 2016 for stroke rehabilitation **[Guo et al. 2016]** that uses dry electrodes in order to collect biomedical signals to map the movement of the user's hand. This does, however, require direct connection to the skin, making it not overly practical from a data glove perspective.

There are also many other papers that are not to do with data gloves that list different methods of sensing angle position that may not yet have been applied to use with data gloves. Looking at these papers may give an additional angle to the potential solution to the problem.

One such paper published in 2015 **[Song et al. 2015]** talks about the potential use of rotor position detection using a linear hall-effect sensor. A linear hall-effect sensor is a sensor that detects a magnetic field and outputs a voltage depending on how far away that field is. This has a linear output. While this paper is talking about its use with rotor position, by using two different sensors allowing the angle to be calculated, this would only work for the rotor positioning, not for use with a human hand. The concept of the sensor, however, could still be applied to a human hand, but some modifications would be required.

All of these papers have helped gleam an insight on the potential methods to help complete this project.

## 2.7) Conclusion

This literature review has helped to demonstrate the potential use a device like this could have. Its uses range from entertainment to medical studies, with all papers looked at in this review concluding that this kind of input device would have a positive outcome for the user.

The papers also help to highlight the range of methods that currently exist for tracking the movement of a joint. No one method has been selected as the de-facto best method for tracking. This is of course partially due to the fact that not all of the papers had the exact same goal in mind, for example a paper that was focused on a cost effective device would have chosen a different method of tracking the joint movement to one that was focused primarily on stroke rehabilitation.

All in all, however, these papers have helped to pick three possible methods of tracking joints that this paper will be testing for the potential use with a full hand. These methods are:

1) Rotary potentiometers, as they have a good linear output and are relatively easy to test and get to produce an output.
2) Force Sensitive Resistors, as they have been used with a force pushed on them via an external skeleton on a glove. However, it is possible that with the deformation caused when they are bent they may be able to produce an output to measure an angle from.
3) Linear Hall Effect Sensor, as while there has been no paper on its use with a data glove it has potential that could help make an effective sensor for one.

With the information gathered in this literature review the project can now move on to its development.

# Chapter 3) Design, development and testing processes

## 3.1) Introduction

In this section of the report, the intended design of the project will be discussed, alongside the how the project will be developed and how the testing processes will work.

The first thing that will be discussed will be the methodology that will be used in the development process, then the components and component testing processes will be discussed. After this we will go through the USB communication from the device to a PC, talking about how and why the Arduino has been chosen to do this.

Finally the C++ on the PC for mapping the output of the Arduino will be talked about. This will go through the stages that the C++ will be developed and how these stages will work, including what will be included in them.

This will give an explanation as to how the project will be designed, developed and tested.

## 3.2) Methodology

In the development of this project a combination of the waterfall and iterative methodologies will be used. The testing of the components will be using a waterfall method, with each component being tested one after each other.

Once the appropriate data has been collected, the development will enter an iterative methodology. This is where aspects of the code are implemented, then tested, then implemented, then tested and so on.

The Arduino code will be done like this initially and then, when that code is at a stage where the data can be sent to the PC, the development will switch to the development of the C++ code.

This will also use iterative developments in order to develop first the basic features for the project, followed by more advanced features. This will continue until either all possible features have been implemented, or until enough has been implemented to collect results if the time is running out to finish.

## 3.3) Component Testing Processes

In order to evaluate the best method of sensing the angle of a joint in a hand with a data glove, multiple components need to be tested. The three different methods that will be tested are: Potentiometers, Force Sensitive Resistors and Linear Hall-Effect Sensors.

*Figure 4 - The Digital Multimeter*

In order to test these, each component will be plugged in to a breadboard that will be powered by 5V from a USB cable. The voltage from the output of the component will be recorded using a digital multimeter that is connected to the breadboard. The component will be tested differently depending on the way it is supposed to function, however they will all be tested with the same equipment to help reduce potential external factors effecting the results.

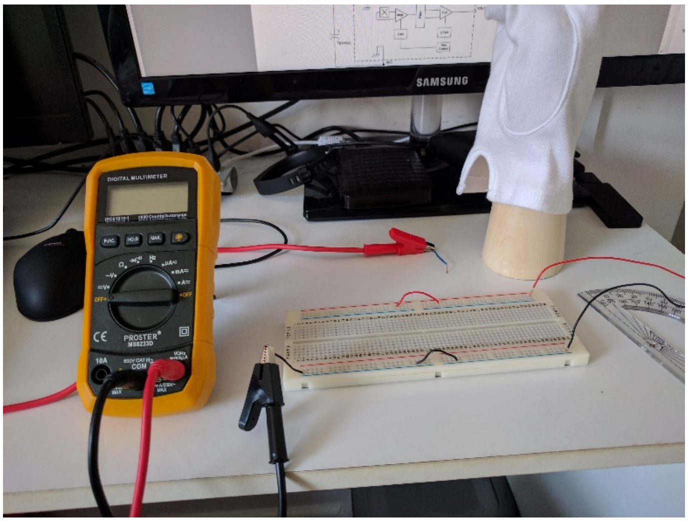


*Figure 5 - The basic setup*

For this project, it was assumed the joint is only able to bend up to a 90 degree angle and as such the outputs of the components will be measured at intervals of 15 degrees, with 3 sets of data for each component. In order to record the results, the joint that will be tested will be measured using a figurine hand to allow recording the angle without the hand shaking and effecting the data.

*Figure 6 - The figurine hand*

The reason that the testing board is being powered by USB is that while data gloves can be wireless, for this project we will be using USB to communicate between the device and PC. As such it makes sense to power the device from the USB cable as well, as it is able to serve two purposes at the same time. As USB power provides 5V, the input the testing of the components should also use 5V.

### 3.3.1) Potentiometers

*Figure 7 - A Potentiometer* ***[RS Components Ltd 2017]***

Potentiometers are three terminal resistors with a sliding or rotating contact that forms as an adjustable voltage divider **[IEEE 2000]**. A voltage divider outputs a different voltage to the input voltage, this is the reason that potentiometers have 3 terminals, the input voltage, the ground and the output voltage.

The potentiometer that will be used with this project will have a 10k resistance value. It will also be a rotating potentiometer so that the angle of the potentiometer can relate to the angle of the joint.

One of the problems with using a potentiometer as an angle sensor for a joint is that while the rotation of a joint is assumed at 90 degrees (for this project at least), potentiometers have a much larger rotational range. As such, when the data is sent from the potentiometer, its output will need to be measured between 0 and 90 degrees beforehand. This information will then need to be used to edit the output to the USB. Any data value that is greater than the voltage of the 90 degree angle will need to be automatically reset to 90 degrees, with the equivalent happening with the minimum 0 degree angle.

### 3.3.2) Force Sensitive Resistors

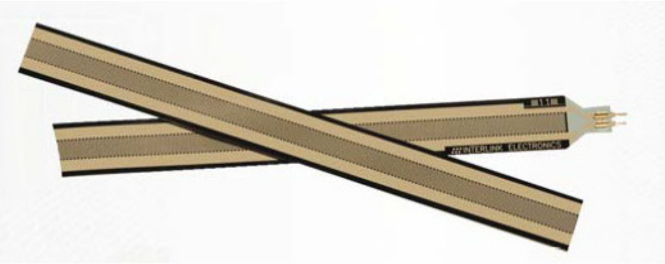


*Figure 8 - A Force Sensitive Resistor* ***[Interlink Electronics 2016b]***

A force sensitive resistor is a sensor whose resistance value changes depending on the force applied to it. The Force Sensitive Resistor (FSR) that will be tested in this project will be the Interlink Force Sensing Resistor 400 series, more specifically the FSR 408.

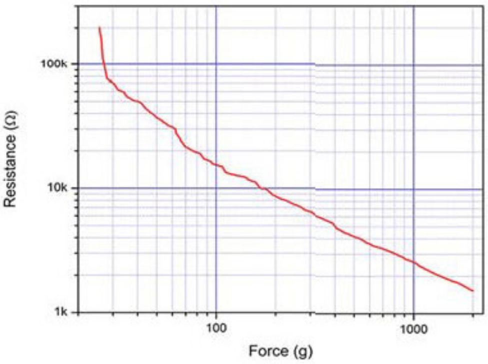


*Figure 9 - The force vs. resistance graph of the FSR* ***[Interlink Electronics 2016b]***

According to the datasheet **[Interlink Electronics 2016a]** “Force Sensing Resistors are robust polymer thick film devices that exhibit a decrease in resistance with an increase in force applied to the surface of the sensor.” This has a rough linear relation, *see figure 9*, which will make it potentially useful with calculating an angle from the force applied to the sensor.

The logic behind trying to use a FSR to get an angle of a joint is that when the sensor is bent, the deformation on the resistor should cause a force to be applied to the sensor. The hope is that this could potentially be used to calculate the angle of the deformation, and therefore the angle of the resistor.

### 3.3.3) Linear Hall Effect-Sensors

*Figure 10 - A Linear Hall Effect Sensor* ***[Allegro MicroSystems 2017]***

A linear hall-effect sensor is a sensor that can detect magnetic fields nearby. It then modifies the output voltage from the output pin in a linear relation to the distance the field is away from the sensor.

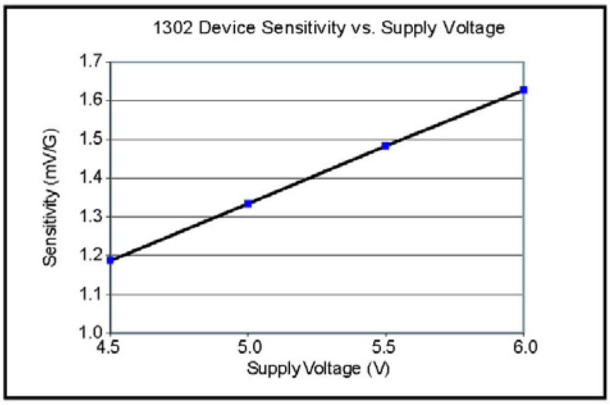


*Figure 11 - The typical characteristics of the A1302KUA-T* ***[Allegro MicroSystems 2017]***

The Linear Hall Effect Sensor that is going to be tested is the Allegro Microsystems A1302KUA-T.  The typical characteristics of this sensor is shown in figure 11. As the output has a linear relation to the distance between the magnetic field and the sensor it should be useful for detecting the angle of the joint, by placing a magnet under the sensor on the joint. Then, when the joint is bent, the magnet will move further away from the sensor, allowing an angle to be calculated.

## 3.4) USB Communication

The development of this device will not only require testing the components to find out their usefulness, but it will also require the ability to transfer the data collected to a PC so that it can be mapped to a 3D environment.

There are various methods to do this, from using microprocessors on a board that reads the data and forwards it on, to just sending the raw data straight to the PC using a USB cable. While performing the literature review, it became clear that a common device that was utilised to achieve this was by using an Arduino to act as a microprocessor.

One of the papers that used an Arduino **[Lipovsky & Ferreira 2015]** was using it to control and for communication with Unity 3D. Another paper **[Silva et al. 2013]** was using the Arduino to act as a low cost solution to the problem of creating a microcontroller for the data glove.

The Arduino can be programmed using the Arduino Language, which is effectively using C. As such, it is fairly easy to program for and can be used to convert the data from the glove into a set of data that can be easily read by the PC.

As such, the device that will be acting as the communication and translation of the data for the project will be an Arduino, more specifically an Arduino Uno.

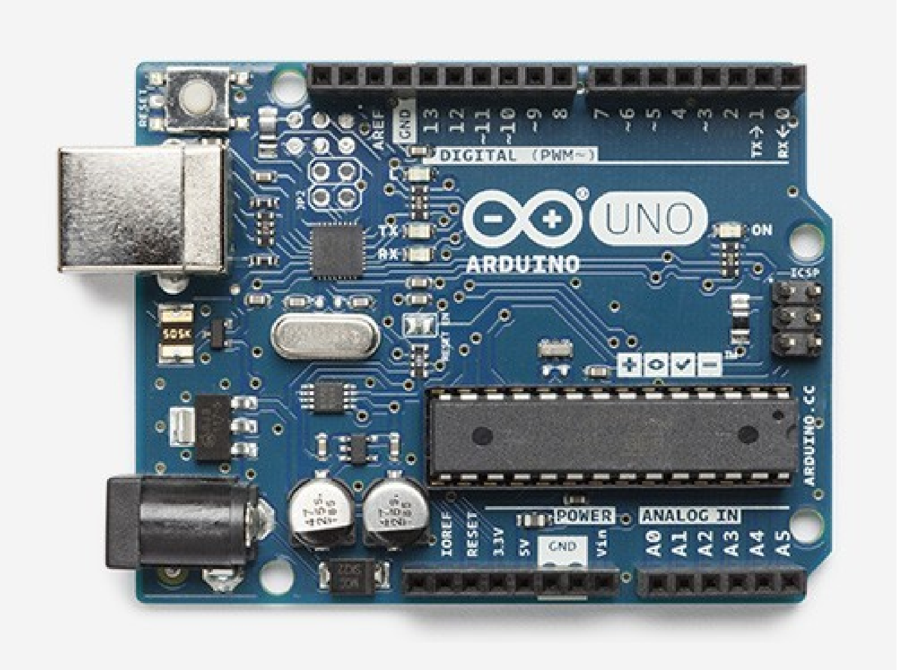


*Figure 12 - An Arduino Uno* ***[Arduino 2017d]***

### 3.4.1) Arduino Uno

In developing the USB communication using the Arduino Uno it will first require understanding how the Arduino Uno takes in data, how it is powered and how it communicates of the USB.

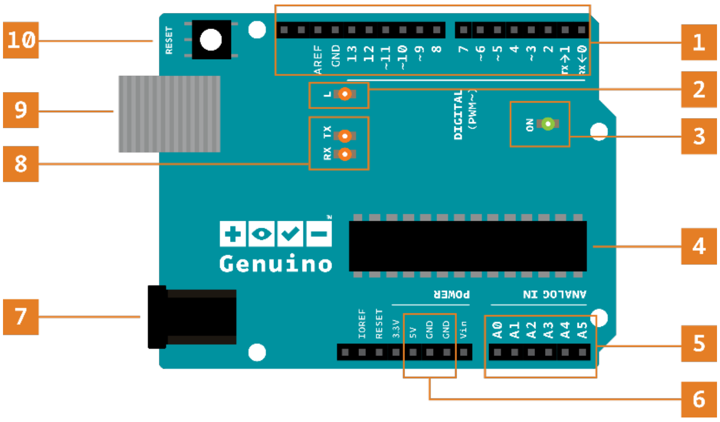


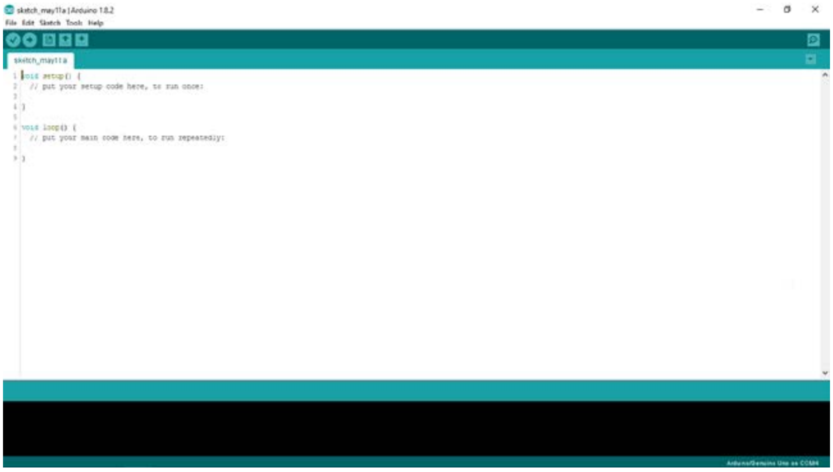
*Figure 13 - The Arduino Board Layout **[Arduino 2017e]***

As shown in figure 13 the Arduino has several parts to its board layout. The main parts that we are interested for this project however are sections 5, 6 and 10.

The Arduino Uno can be powered via the USB and includes output pins for 5V and ground that will be used to power an external breadboard that will house the components. The USB connection port is section on 9 on the board layout **[Arduino 2017a]**. It uses a USB type B for the port on the board, coming with a USB type A to type B cable.

Section 6 on the board is the output pins for the 5V and ground **[Arduino 2017a]**. This is then able to power the external breadboard via a wire that can be connected to the pin.

Section 5 on the board contains the Analogue inputs for the board **[Arduino 2017a]**. This allows the analogue voltage output from the tested component to be passed and read by the board, with the range of the voltage that can be read from 0V to 5V. The input from these pins can be read using the analogRead() command within the programming of the Arduino. The input of these pins will be a number between 0 and 1023, relating to the range of 0V to 5V **[Arduino 2017b]**. The number of available analogue pins for the Arduino Uno is 6, meaning that it can have 6 of the components data being fed to it. This would not be enough for a whole hands worth of joints, but will be enough for the testing of this project.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

*Figure 14 - A screenshot of the IDE*

The software on the Arduino will be written using the Arduino IDE. This IDE contains the ability to compile for the Arduino and upload it on to the Arduino. In order to code the software the Arduino uses its own language, which is effectively C with some additional functions specifically for the Arduino thrown in.

Each program needs to include at least: a setup() function for the code that needs to be run once at the software start up, and a loop() function which contains the code that will run repeatedly after the setup function.

In order to use the input pins, each pin needs to be declared in the code as an integer (for example *int pinIn = A0;).* This can then be used with the analogRead() function to get the value of the pin.

So that the Arduino can send data back down the USB cable to the PC it needs to open a serial. This is done using the Serial.begin() function **[Arduino 2017c]**. This needs the serial data rate to be included in the parameters of the begin function. In order to send the data through the serial the command Serial.println() needs to be used. This is to print the line of data included in the parameters through the serial that is open.

The output of the Arduino will be a 6 bit hexadecimal value that will be sent through the serial to the PC.

## 3.5) 3D Environment Mapping

In order to map the data from the joints to a 3D environment, a method of communication between the data and the software will need to be developed. The 3D environment will also have to be developed.

The code will be written in C++ with the aim of it being able to be re-used with different engines by making it a modular set of code that does not require any additional codebases. This is to enable the potential to use the same code with multiple external engines, rather than one specifically designed just for use with this code.

The development of this code will be in stages. The first stage will be to get the C++ to be able to convert a 6 bit hexadecimal value into an equivalent angle that can be used to set the rotation. The hex values will be entered manually by the user as a simulation of the input from the Arduino, the output of which will be in the form of values printed out of a console.

Next, this simulation code will be integrated to an external engine. The simulated data will again be entered by the user, however instead of printing the data as values in a console, they will be used to rotate a 3D rectangle.

After this the same methods will be applied, only using communication through the COM ports to get the outputted data from the Arduino instead of the manually entered data from the user.

### 3.5.1) C++ Console Implementation

For the basic C++ console implementation all that is needed is a simple console based C++ application. The data that needs to be converted will be sent from the user via the console. The functions for handling the input data will all be encapsulated within its own class. The basic UML of which is shown in figure 15.

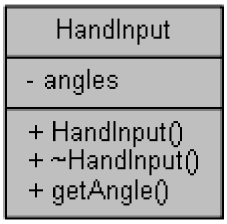


*Figure 15 - Basic input UML*

This get angle class will have a const char array as a parameter, as that is how the data will be received from the COM port when it is sent. It then takes this char array and reads it as a 6 bit Hexadecimal value. This value will correlate with one of the indexes in the angles array. This will contain the equivalent angle in the form of a float variable for degrees. This will then be the number that gets returned from the function.

Once this is written it will be able to be ported for use with the external engine. After that has been completed then the console application will be used to get the communication between the COM port and the code. The reason that this is happening after the engine integration is because if the communication to C++ is not possible in the time that is available, with the ability to simulate the data it is possible to still collect results about the implementation.

### 3.5.2) C++ External Engine Integration

The external engine that will be used in with this project will be a stripped out version of the engine that was built for a previous university assignment. This is just so that it is quicker to develop for, due to the already existing experience with the engine.

The code that converts and simulates the input data will be taken from the console based implementation and then integrated into this engine. In order to help both simulate what will be required when reading in data from the COM port, and to allow input via the console input for the 3D output, the in data will be written within a thread that runs at the same time as the rest of the functions in the engine. A UML diagram of the class that will wrap the thread is shown in figure 16.

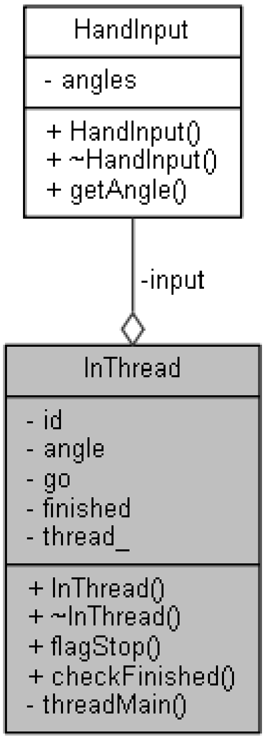


*Figure 16 - The thread UML*

The thread will contain a pointer to a float variable that will be used to set the rotation of a 3D cube. This means that when the variable is changed in the thread it will change the rotation of the rectangle.

With the COM port communication, it will run the same way, with the thread constantly listening to the output of the COM port and updating the angle accordingly.

# Chapter 4) Results

## 4.1) Component Testing – Potentiometer

The first component that was tested was the potentiometer. On the potentiometer there is a straight line that was used to align the angle with a protractor. The setup of the testing is shown in figure 17.

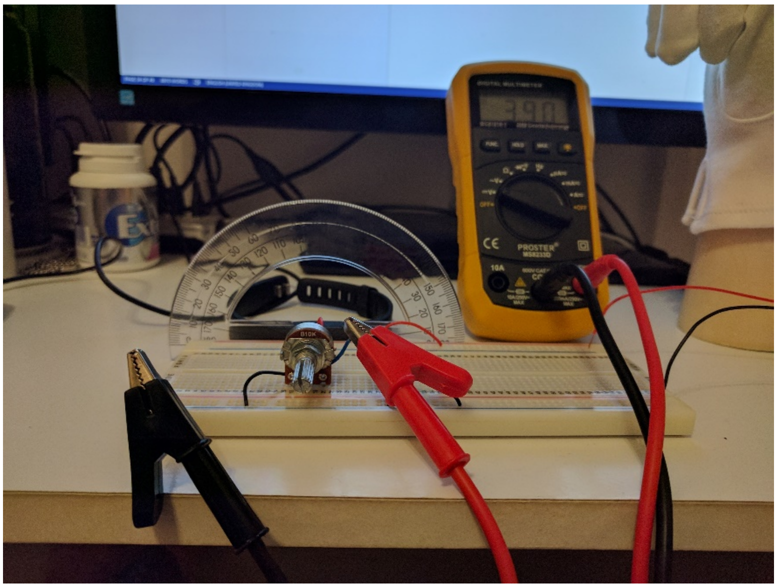
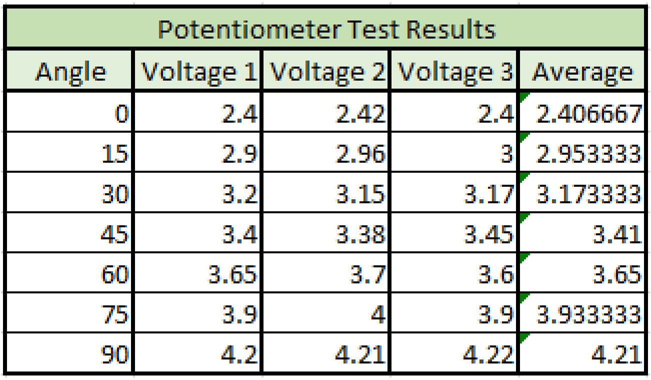


*Figure 17 - The potentiometer testing*

The angles to be tested were between 0 and 90 degrees. The voltage of the output at the angles was recorded, with the angles to measure going up in sets degrees of 15. This was repeated 3 times and then averaged out to give the final results. Figure 18 shows the testing results.



| Potentiometer Test Results | | | | |
|---|---|---|---|---|
| Angle | Voltage 1 | Voltage 2 | Voltage 3 | Average |
| 0 | 2.4 | 2.42 | 2.4 | 2.406667 |
| 15 | 2.9 | 2.96 | 3 | 2.953333 |
| 30 | 3.2 | 3.15 | 3.17 | 3.173333 |
| 45 | 3.4 | 3.38 | 3.45 | 3.41 |
| 60 | 3.65 | 3.7 | 3.6 | 3.65 |
| 75 | 3.9 | 4 | 3.9 | 3.933333 |
| 90 | 4.2 | 4.21 | 4.22 | 4.21 |

*Figure 18 - The Potentiometers testing results*

If the average results from this testing is shown in a line graph, see figure 19, there is quite clearly a fairly linear correlation between the angle of the potentiometer and the voltage output. The output is not 100% linear, however that could just be a result of the testing process as getting the angle is entirely prone to human error as the angle measurement is done by eye.

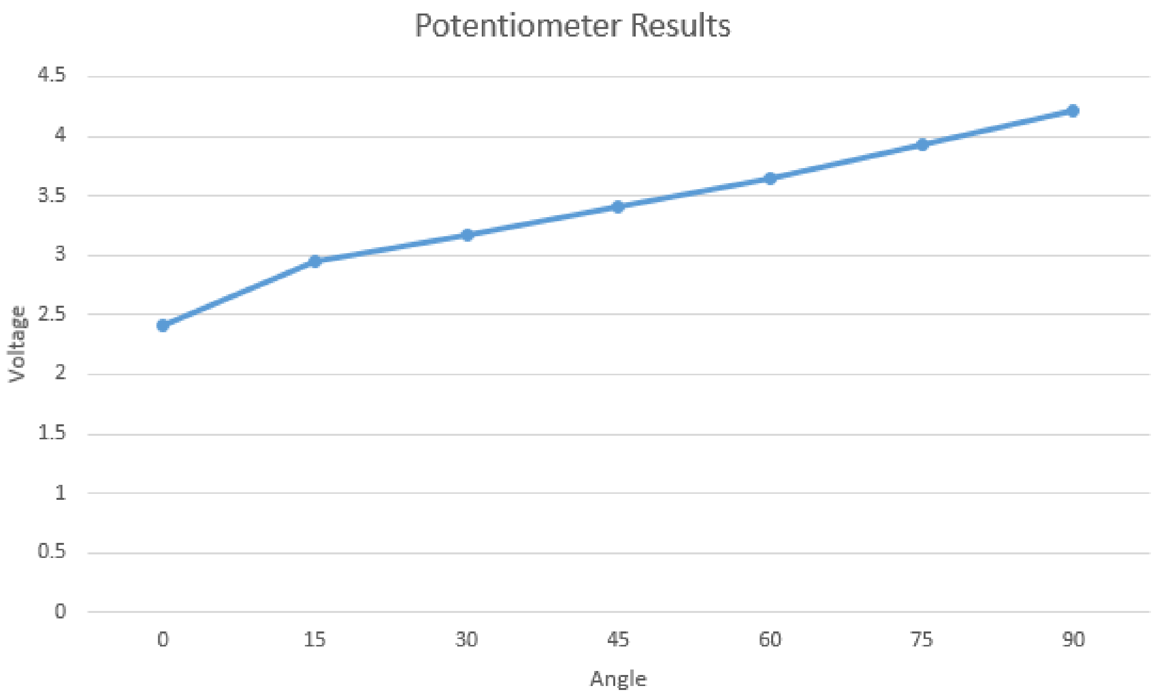


*Figure 19 - A graph showing the Potentiometer results*

Using this information it would definitely be possible to calculate the angle of a joint in a hand using this method of tracking angles.

## 4.2) Component Testing – Force Sensitive Resistor

The next component to be tested was the force sensitive resistor. In order to test this, the sensor was initially bent at 90 degrees both forwards and backwards to decide which would be the better direction to cause deformation on the sensor. The setup for the testing is shown in figure 20.

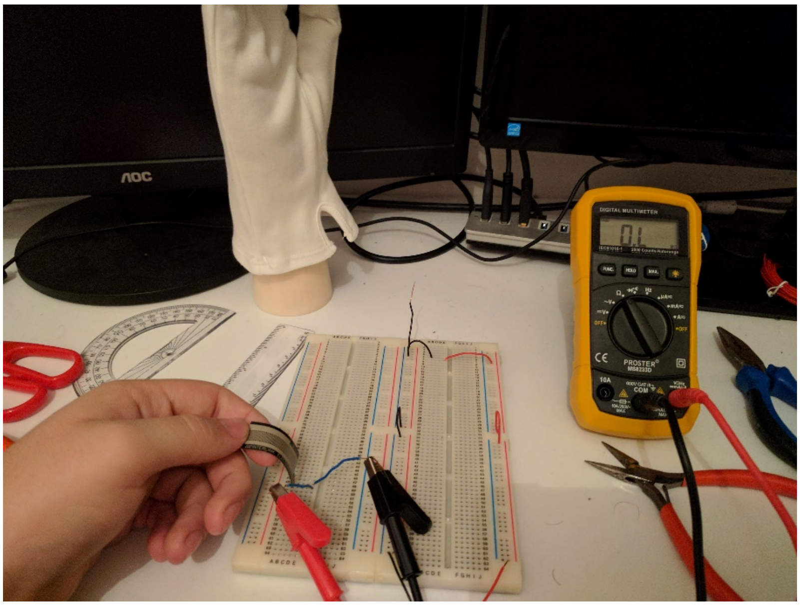


*Figure 20 - The testing setup for the FSR*

The point of testing this component was to see if it could be used for sensing an angle via the resistance change caused due to force applied on the sensor during its deformation. Unfortunately this does not seem to be the case of what would actually happen. Initially during testing, while the sensor was being bent it would change the output, however, it turned out that this was being caused by the force being held on it during the bending process. This meant that the data collected was invalid and had to be rejected. After reviewing the testing process it was decided to hold it at an angle that did not cause additional pressure from movement. When this test was performed it became clear that the deformation of the sensor was not able to cause any real difference to the output. This was the same either direction the sensor was bent.

As such this is not a plausible method of getting the angle of a joint and so was not used when the Arduino was being developed.

## 4.3) Component Testing – Linear Hall-Effect Sensor

The final component to be tested was the Linear Hall-Effect Sensor. During the researching of this component the sensor was demonstrated, via the use of an oscilloscope, to have a linear output when moved towards and away from a small magnet. This data, however was not recorded so is unable to be shown in these results.

It is important to note that while the results from the testing during this project was unable to replicate this happening in the time frame that was left to complete this project and even though the data shown and talked about here would seem to indicate that it is not applicable, it may still be a viable option for measuring the angle of a joint.

However, as far as this projects testing is concerned this is not what happened. The setup of the testing is shown in figure 21.

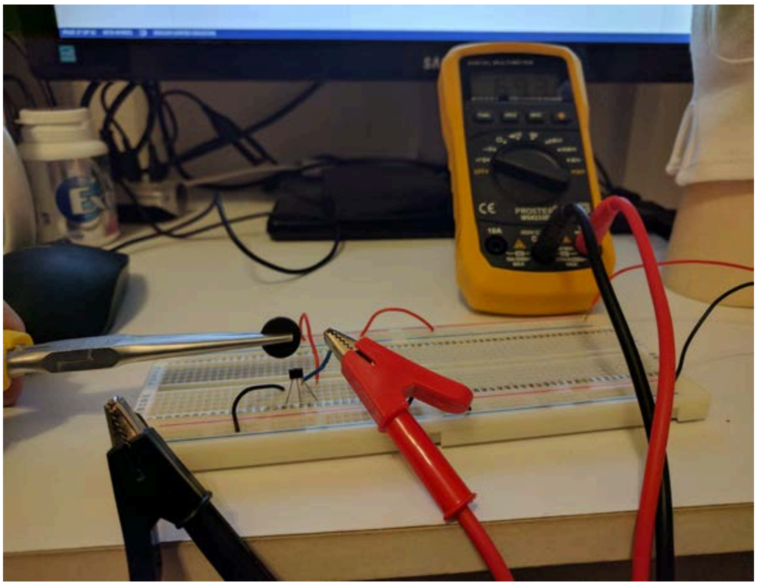


*Figure 21 - Testing setup of the Linear Hall-Effect Sensor*

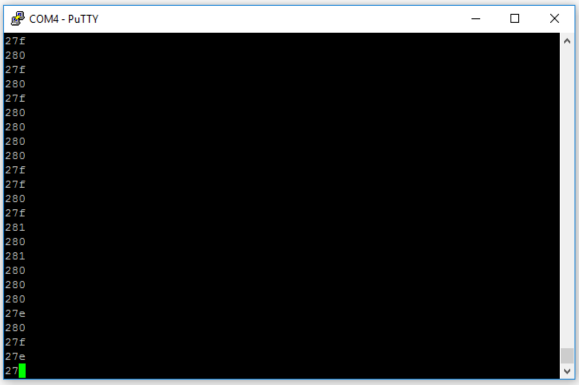
When the magnet was moved near the sensor the output did not change. No matter what was tried the output would not change and after a certain amount of time this had to be abandoned due to the time constraints on the project. As such it had to be concluded that this was also not a viable option and would not be used with the Arduino.

## 4.4) Arduino Testing

After testing the components, it was concluded that the component that could be used in conjunction with the Arduino was the potentiometer. As such the Arduino was setup with the potentiometer for testing, see figure 22.

*Figure 22 - The Arduino setup*

The initial code that was written for the Arduino was a simple setup where the input from the potentiometer was sent to A0 (Analogue input 0). This was then converted from its numerical value to a hex string. The Arduino then printed the value of this hex string out via the serial. This meant that the value of the output was from the whole range of the 0-5V that was sent out from the potentiometer, as it was not being modified to make sure it was the calculated 0-90 degrees.



```
27f
280
27f
280
27f
280
280
280
280
27f
27f
280
27f
281
280
281
280
280
280
27e
280
27f
27e
27
```

*Figure 23 - PuTTY Screenshot*

In order to see what the output was, the COM port was opened up using PuTTY, so that the output from the Arduino could be seen, see figure 23. This showed that the output from the Arduino was in fact correctly outputting the data from 0 to 3ff which was the data that was expected for the whole range.

With the data being outputted correctly from the Arduino it was now time to change the code to make sure that the output only applied to the range from 0 to 90 degrees. From the testing with the potentiometer it was known that the range was from 2.4V to 4.2V. As such anything that was higher than 4.2V needed to return 3ff and anything below 2.4V needed to return 0. The range between 2.4V and 4.2 V being 1.8V.

This meant that in order to get the range of each number for 0 to 1023 the number to use to convert could be calculated from 1024 / 1.8. The value from the analogue pin needed to first be converted to a voltage value, this meant multiplying it by 0.0049 **[Arduino 2017a].** It was then necessary to make sure the value was in the range 2.4V – 4.2V, by setting it to max or min if it was outside these values and then take 2.4V from the number to get a number between 0 and 1.8. Next this number was converted by the number calculated earlier for the conversion and the result converted to a whole integer.

After this the data was sent to the serial in the form of a string hex value. When the output was observed in PuTTY the expected data was correct. This was tested using the setup in figure 24.

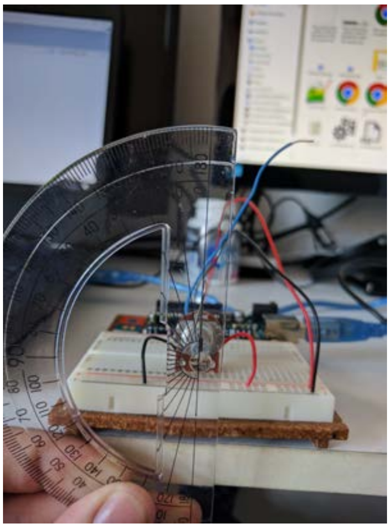


*Figure 24 - Arduino Testing Setup 2*

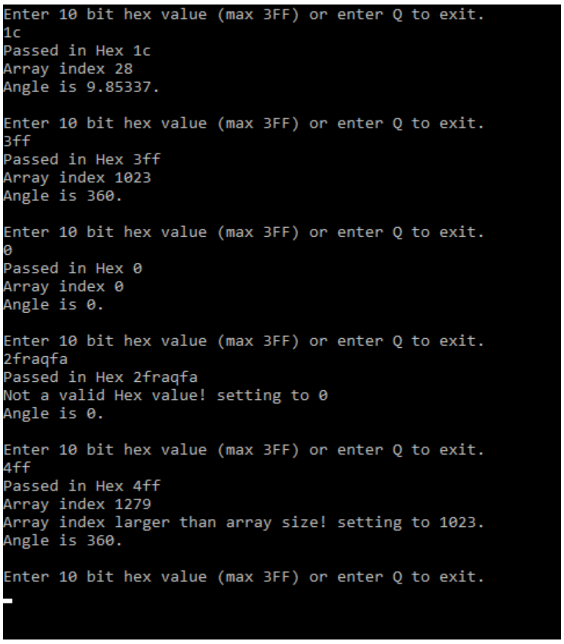
The results of this was that when the potentiometer was rotated lower than the 0 degrees mark, it was still outputting 0, and when it was passed 90 it outputted 3ff. The output between the two points appeared to be a linear as expected. This showed that the Arduino setup worked using the potentiometer for getting an angle of a joint.

## 4.5) C++ Testing

The final thing to be done was the C++ program. The first part was creating a C++ console application that could convert a user inputted hex value to an equivalent angle. As such, a class was created that contained both an array of angles, and a number that could be used to calculate the angles in the array.

As the range of the output was a 6 bit hex value it meant the number of angles that could be represented was 1024. As such, an array of size 1024 was declared in the header of the class. A define called convNum was added to the header as well. This define was for the number 360 / 1023. This meant the range of the angles was from 0 to 360. Upon initialisation of the class the array is filled by a loop that uses the current index of the loop multiplied by the convNum to get the equivalent angle stored.

After this a function called getAngle() was created. This function took in a const char array that would contain the hex value to translate. The function checks that the data in is a valid hex value, if not it sets the angle index as 0 and prints out an error. If it is valid it then converts the hex to an integer. If the integer is greater than 1023 it prints an error and sets the angle index to 1023. It then returns the contents of the angle array at the angle index. The output of this is shown in figure 25.



```
Enter 10 bit hex value (max 3FF) or enter Q to exit.
1c
Passed in Hex 1c
Array index 28
Angle is 9.85337.

Enter 10 bit hex value (max 3FF) or enter Q to exit.
3ff
Passed in Hex 3ff
Array index 1023
Angle is 360.

Enter 10 bit hex value (max 3FF) or enter Q to exit.
0
Passed in Hex 0
Array index 0
Angle is 0.

Enter 10 bit hex value (max 3FF) or enter Q to exit.
2fraqfa
Passed in Hex 2fraqfa
Not a valid Hex value! setting to 0
Angle is 0.

Enter 10 bit hex value (max 3FF) or enter Q to exit.
4ff
Passed in Hex 4ff
Array index 1279
Array index larger than array size! setting to 1023.
Angle is 360.

Enter 10 bit hex value (max 3FF) or enter Q to exit.
```

*Figure 25 - C++ Console Output*

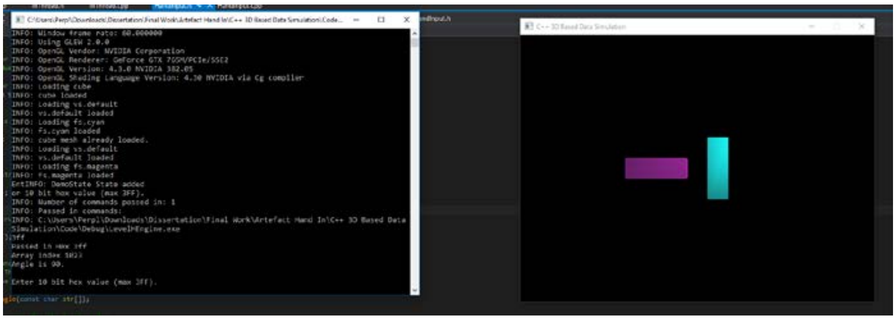
This shows that the console program works as intended.

The next stage of the C++ program was integrating this console program into the existing 3D engine that was being used. The HandleInput class from the console code was included into the engine structure. Then a new class for handling the thread for the input was created.

In this class a pointer to a float variable for the angle was included. This was so that the angle could be modified within the thread so that it was readable in the rest of the engine. To help with thread safety, the class contained Boolean flags for if the thread should go or has finished, along with the id of the thread and actual thread variable. It also contained functions for checking and setting the thread externally.

The thread itself handles the input from the console that allows the user to pass in the hex data to test with. This is also where the HandleInput class is handled. There are two reasons to utilise threading to do this, the first being that when communicating with a COM port, the code would have to use a thread to deal with listening to the output from the port. As such this helped with simulating the full version. It also is needed because otherwise the 3D environment would never render as it would be waiting for input before every frame could be rendered.

When the hex value is entered the float pointer for the angle is updated. This then is used by the updated function of the engine, rotating the 3D rectangle to the new angle. Unlike the console, the range is set to 0 – 90 degrees to help with the simulation of the Arduino output. As shown with figure 26 this worked well.



```
INFO: Window frame rate: 60.000000
INFO: Using GLEW 2.0.0
INFO: OpenGL Vendor: NVIDIA Corporation
INFO: OpenGL Renderer: GeForce GTX 765M/PCIe/SSE2
INFO: OpenGL Version: 4.3.0 NVIDIA 382.05
INFO: OpenGL Shading Language Version: 4.30 NVIDIA via Cg compiler
INFO: Loading cube
INFO: cube loaded
INFO: Loading vs.default
INFO: vs.default loaded
INFO: Loading fs.cyan
INFO: fs.cyan loaded
INFO: cube mesh already loaded.
INFO: Loading vs.default
INFO: vs.default loaded
INFO: Loading fs.magenta
INFO: fs.magenta loaded
EntINFO: DemoState State added
er 10 bit hex value (max 3FF).
INFO: Number of commands passed in: 1
INFO: Passed in commands:
INFO: C:\Users\Perpl\Downloads\Dissertation\Final Work\Artefact Hand In\C++ 3D Based Data
Simulation\Code\Debug\LevelHEngine.exe
3ff
Passed in hex 3ff
Array index 1023
Angle is 90.

Enter 10 bit hex value (max 3FF).
```

*Figure 26 - 3D C++ Demo*

Unfortunately due to time constraints it was not possible to get the Arduino to communicate directly to the C++, however the simulation uses the same data that the Arduino has been shown to output, proving that it would work.

# Chapter 5) Discussion and Conclusions

## 5.1) Conclusions

After working on this project, several conclusions can be made.

The first being to do with the potential use of the components that were tested within this project. From purely a results stand point the potentiometer was the clear winner. It has a good linear output that can be mapped well to the angle of a joint. This is in comparison to the FSR which did not work and is quite clearly not designed with this purpose in mind. It may have worked if there had been a way to apply pressure to it externally when a joint was bent, but with the testing done it was not useable. The linear hall-effect sensor seemed to have potential, but due to time constraints in this project it was never really give enough time to successfully be used.

The potential problem with this conclusion is due the fact that this was just for one joint. When applied to a whole hand this could become a problem as both the potentiometer and the FSR would require some kind of external exoskeleton attached to the glove to work. This would mean that it would require calibration for each person’s hand size, which could be problematic. If the Linear Hall-Effect Sensor had worked it would not have needed this, meaning that it would have been the preferred option.

In the end though, out of the three methods, this project would have to conclude that the best one for mapping a single digit joint position to a 3D environment would be the potentiometer.

Using an Arduino to act as the microprocessor to send the data to the PC is completely a viable option. It is easy to use and is the part of this project that worked 100% as intended and is an excellent conclusion to make for a cheap data glove.

The C++ aspect of this project has worked well, given the problem of the time constraint. Even though it does not communicate with the Arduino it does simulate the data exactly how it would do and outputs the response that would be expected, allowing the angles to be mapped to a 3D environment.

Overall this project has had its problems, but has achieved its goal of prototyping an electronic input device for mapping a single digit joint position to a 3D environment.

## 5.2) Critical Reflection

During the process of this project there have been many opportunities for critical reflection. The biggest of which falls down to poor planning and failing to take into account the time constraints set. When planning this project the author failed to consider other factors when planning the time schedule. The main one being failing to take into consideration the high number of other large university assignments that were required to be done at the same time. This meant that the scope had to be cut down to prototyping a single joint and evaluating its use for a whole hand.

It also meant that not all of the intended testing was performed. One test was asking a client in the medical field their opinion on the device for use with medical applications. Due to the device not being ready on time this never happened.

Another was in the way that the components were tested. In an ideal scenario they would have been given more time to be tested, however, when the author was unable to get the FSR to work, too much time was spent trying to fix it, which in turn caused the Linear Hall-Effect Sensor to not get the time it deserved for testing.

Not everything was a failure however. The Arduino and C++ was mostly a success, with the only issue being the lack of communication between the two which, once again was due to a lack of time.

After finishing this project the author has learnt a lot about how to correctly plan a project on this scale and how to make sure the project is a higher success.

## 5.3) Future Work

If given the opportunity to add to this project the first thing that should be completed would be to have to get the Arduino to actually communicate with the C++. The possibility of using the Linear Hall-Effect Sensor should be researched some more to give it a fair chance to show it’s possible uses. The project could also be expanded to cover testing the whole hand, not just a single joint and the device could be checked by a client in the medical field for their opinion on its usefulness.

# References

Anthes & García-Hernández & Wiedemann & Kranzlmüller, C.A & R.J.G & M.W & D.K, 2016. State of the art of virtual reality technology. Aerospace Conference, 5-12 March 2016. IEEE. Available from: http://ieeexplore.ieee.org/document/7500674/ [10th May 2017].

Arduino, A., 2017a. analogRead(). arduino.cc: Arduino. Available from: https://www.arduino.cc/en/Reference/analogRead [10th May 2017].

Arduino, A., 2017b. Arduino/Genuino Uno Board Anatomy. arduino.cc: Arduino. Available from: https://www.arduino.cc/en/Guide/BoardAnatomy [10th May 2017].

Arduino, A., 2017c. Serial begin(). arduino.cc: Arduino. Available from: https://www.arduino.cc/en/Serial/Begin [10th May 2017].

Cao & Gao & Wang & Li, Q.C & C.G & Y.W & T.L, 2016. The motion capture data glove device for virtual surgery. IEEE International Nanoelectronics Conference (INEC), Chengdu, China, May 9-11, 2016. IEEE. Available from: http://ieeexplore.ieee.org/document/7589461/ [10th May 2017].

Fang & Sun & Liu & Guo, B.F & F.S & H.K & D.G, 2016. A novel data glove for fingers motion capture using inertial and magnetic measurement units. IEEE International Conference on Robotics and Biomimetics, Qingdao, China, December 3-7, 2016. IEEE. Available from: http://ieeexplore.ieee.org/document/7866639/ [10th May 2017].

Guo & Liu & Zhang & Zhang & Yamamoto, S.G & Y.L & Y.Z & S.Z & K.Y, 2016. A VR-based self-rehabilitation system. International Conference on Mechatronics and Automation, August 7 - 10, Harbin, China. IEEE. Available from: http://ieeexplore.ieee.org/document/7558728/ [10th May 2017].

HTC Corporation, H.C., 2017. Controller. vive.com: HTC Corporation. Available from: https://www.vive.com/us/accessory/controller/ [10th May 2017].

IEEE, 2000. The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition. http://ieeexplore.ieee.org: IEEE. Available from: http://ieeexplore.ieee.org/document/4116787/ [10th May 2017].

Jack & Boian & Merians & Tremaine & Burdea & Adamovich & Recce & Poizner, D.J & R.B & A.S.M & M.T & G.C.B & S.V.A & M.R & H.P, 2001. Virtual reality-enhanced stroke rehabilitation. IEEE Transactions on Neural Systems and Rehabilitation Engineering, Volume 9 Issue. 3, Page 308-318.

Kim & Thang & Kim, J.K & N.D.T & T.K, 2009. 3-D hand motion tracking and gesture recognition using a data glove. IEEE International Symposium on Industrial Electronics, Seoul Olympic Parktel, Seoul, Korea, July 5-8, 2009. IEEE. Available from: http://ieeexplore.ieee.org/document/5221998/ [10th May 2017].

Lipovsky & Ferreira, R.L & H.A.F, 2015. Hand therapist: A rehabilitation approach based on wearable technology and video gaming. IEEE 4th Portuguese Meeting on Bioengineering (ENBENG), Porto, Portugal, February 26-28, 2015. IEEE. Available from: http://ieeexplore.ieee.org/document/7088817/ [10th May 2017].

Microsoft, M., 2017. Kinect for Windows Sensor Components and Specifications. msdn.microsoft.com: Microsoft. Available from: https://msdn.microsoft.com/en-us/library/jj131033.aspx [10th May 2017].

Murthy & Jadon, G.R.S.M & R.S.J, 2009. A review of vision based hand gestures recognition. International Journal of Information Technology and Knowledge Management, Volume 2 No. 2, Page 405-410.

Othman & Hamzah & Hussain & Baharudin & Rosli & Ani, A.O & N.H & Z.H & R.B & A.D.R & A.I.C.A, 2016. Design and development of an adjustable angle sensor based on rotary potentiometer for measuring finger flexion. 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Batu Ferringhi, Malaysia, November 25-27, 2016. IEEE. Available from: http://ieeexplore.ieee.org/document/7893640/ [10th May 2017].

Pereira & Aroca & Dantas, L.C.P & R.V.A & R.R.D, 2013. FlexDGlove: A Low Cost Dataglove with Virtual Hand Simulator for Virtual Reality Applications. 2013 XV Symposium on Virtual and Augmented Reality, 28-31 May 2013. IEEE. Available from: http://ieeexplore.ieee.org/document/6655794/ [10th May 2017].

Prasad & Kumar & Sinha, S.P & P.K & K.P.S, 2014. A wireless dynamic gesture user interface for HCI using hand data glove. Seventh International Conference on Contemporary Computing (IC3), Noida, India, August 7-9, 2014. IEEE. Available from: http://ieeexplore.ieee.org/document/6897148/ [10th May 2017].

Rahman & Hoque & Zinnah & Bokhary, Y.A.R & M.M.H & K.I.Z & I.M.B, 2014. Helping-Hand: A data glove technology for rehabilitation of monoplegia patients. 9th International Forum on Strategic Technology (IFOST), Cox's Bazar, Bangladesh, October 21-23, 2014. IEEE. Available from: http://ieeexplore.ieee.org/document/6991104/ [10th May 2017].

Silva & Dantas & Pantoja & Pereira, L.S & R.D & A.P & A.P, 2013. Development of a low cost dataglove based on arduino for virtual reality applications. IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), Milan, Italy, July 15-17, 2013. IEEE. Available from: http://ieeexplore.ieee.org/document/6617395/ [10th May 2017].

Song & Fang & Han, X.S & J.F & B.H, 2015. High-Precision Rotor Position Detection for High-Speed Surface PMSM Drive Based on Linear Hall-Effect Sensors. IEEE Transactions on Power Electronics, Volume 31 Issue. 7, Page 4720-4731.

Tavares & Abreu & Quintas, R.T & P.A & M.Q, 2015. Instrumented glove for rehabilitation exercises. 3rd Experiment@ International Conference (exp.at'15), Ponta Delgada, Portugal, June 2-4, 2015. IEEE. Available from: http://ieeexplore.ieee.org/document/7463229/ [10th May 2017].

Vicente & Faisal, A.P.V & A.A.F, 2013. Calibration of kinematic body sensor networks: Kinect-based gauging of data gloves “in the wild”. IEEE International Conference on Body Sensor Networks (BSN), Cambridge, MA, USA, May 6-9, 2013. IEEE. Available from: http://ieeexplore.ieee.org/document/6575526/ [10th May 2017].

Hsiao & Yang & Lin & Lee & Chou, P.H & S.Y & B.L & I.L & W.C, 2015. Data glove embedded with 9-axis IMU and force sensing sensors for evaluation of hand function. 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, August 25-29, 2015. IEEE. Available from: http://ieeexplore.ieee.org/document/7319426/ [10th May 2017].

Yoon & Jang & Cho, J.Y & S.J & S.C, 2010. Enhanced user immersive experience with a virtual reality based FPS game interface. IEEE Symposium on Computational Intelligence and Games (CIG), Copenhagen, Denmark, August 18-21, 2010. IEEE. Available from: http://ieeexplore.ieee.org/document/5593369/ [10th May 2017].

Interlink Electronics, 2016a. FSR 400 Series Data Sheet. robotshop.com: Interlink Electronics. Available from: http://www.robotshop.com/media/files/pdf/datasheet-30-61710.pdf [10th May 2017].

**Figure 1:**

HTC Corporation, 2017. Controller. vive.com: HTC Corporation. Available from: https://www.vive.com/us/accessory/controller/ [10th May 2017].

**Figure 2:**

Microsoft, 2017. Kinect for Windows Sensor Components and Specifications. msdn.microsoft.com: Microsoft. Available from: https://msdn.microsoft.com/en-us/library/jj131033.aspx [10th May 2017].

**Figure 3:**

CyberGlove Systems, 2016. CyberGlove III. cyberglovesystems.com: CyberGlove Systems. Available from: http://www.cyberglovesystems.com/cyberglove-iii/ [10th May 2017].

**Figure 7:**

RS Components Ltd, 2017. Alps RK163 Series Carbon Film Potentiometer with a 6 mm Dia. Shaft, 10kΩ, ±10%, 0.1W, Panel Mount (Through Hole). uk.rs-online.com: RS Components Ltd. Available from: http://media.rs-online.com/t_large/F2499238-01.jpg [10th May 2017].

**Figure 8 & 9:**

Interlink Electronics, 2016b. FSR 400 Series Data Sheet. robotshop.com: Interlink Electronics. Available from: http://www.robotshop.com/media/files/pdf/datasheet-30-61710.pdf [10th May 2017].

**Figure 10 & 11:**

Allegro MicroSystems, 2017. A1301 and A1302. farnell.com: Allegro MicroSystems. Available from: http://www.farnell.com/datasheets/1934044.pdf?_ga=2.97987999.1664467009.1494462820-763961893.1492542598 [10th May 2017].

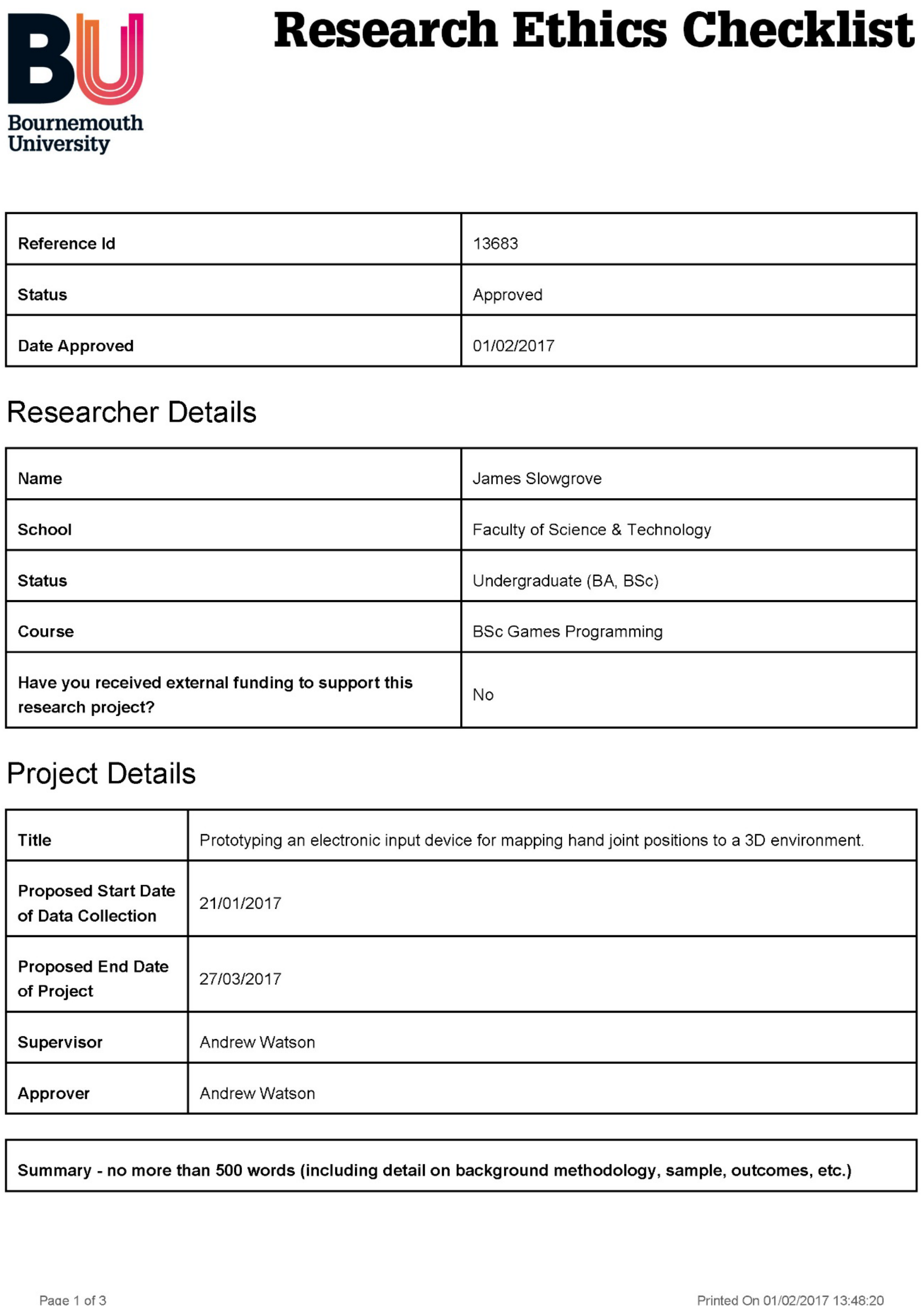
**Figure 12:**

Arduino, A., 2017d. Arduino Uno Rev3. arduino.cc: Arduino. Available from: https://store.arduino.cc/arduino-uno-rev3 [10th May 2017].

**Figure 13:**

Arduino, A., 2017e. Arduino/Genuino Uno Board Anatomy. arduino.cc: Arduino. Available from: https://www.arduino.cc/en/Guide/BoardAnatomy [10th May 2017].

# Appendices 1 – Ethics



## Research Ethics Checklist

| | |
|---|---|
| **Reference Id** | 13683 |
| **Status** | Approved |
| **Date Approved** | 01/02/2017 |

### Researcher Details

| | |
|---|---|
| **Name** | James Slowgrove |
| **School** | Faculty of Science & Technology |
| **Status** | Undergraduate (BA, BSc) |
| **Course** | BSc Games Programming |
| **Have you received external funding to support this research project?** | No |

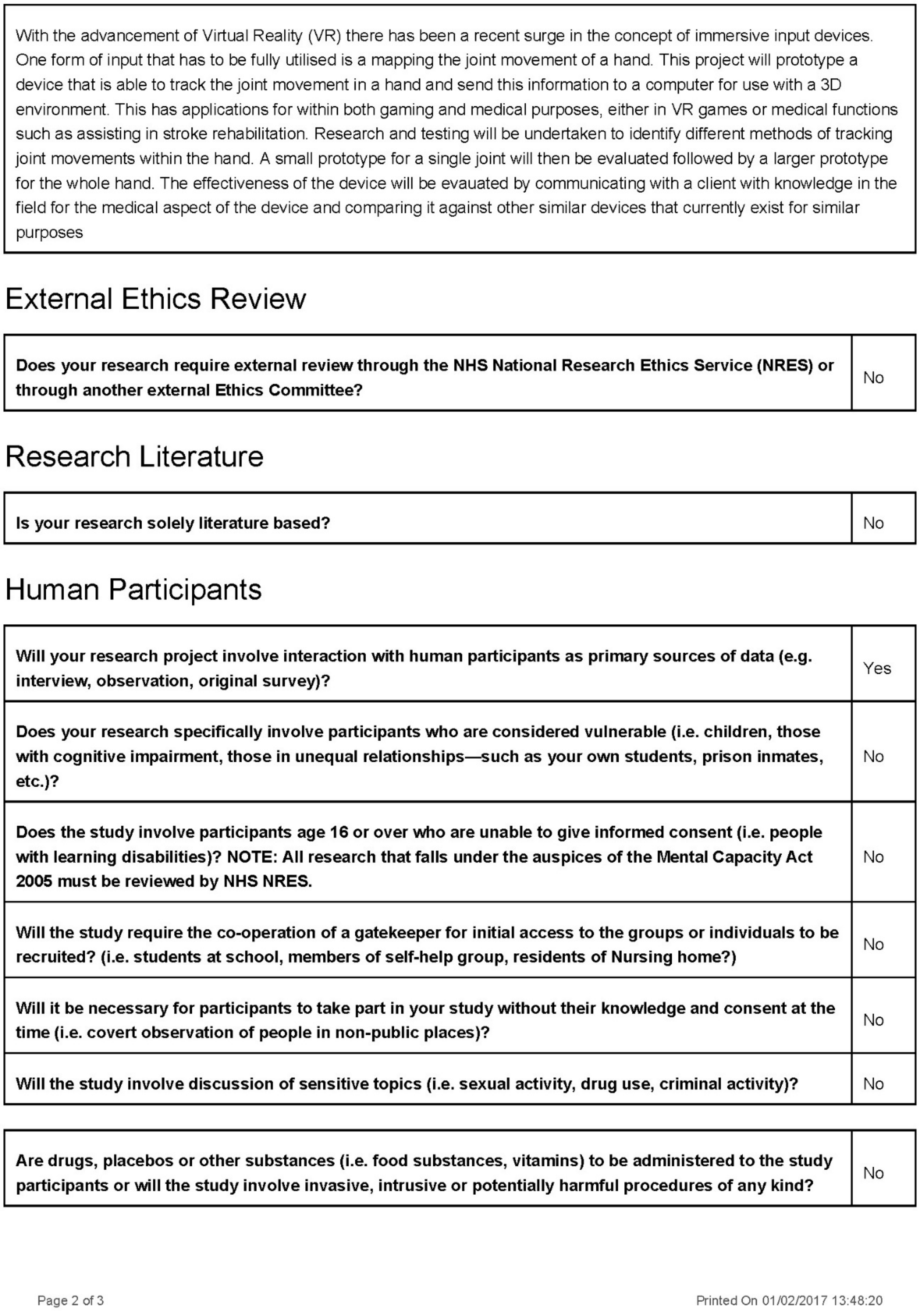
### Project Details

| | |
|---|---|
| **Title** | Prototyping an electronic input device for mapping hand joint positions to a 3D environment. |
| **Proposed Start Date of Data Collection** | 21/01/2017 |
| **Proposed End Date of Project** | 27/03/2017 |
| **Supervisor** | Andrew Watson |
| **Approver** | Andrew Watson |

| |
|---|
| **Summary - no more than 500 words (including detail on background methodology, sample, outcomes, etc.)** |

Page 1 of 3

Printed On 01/02/2017 13:48:20



With the advancement of Virtual Reality (VR) there has been a recent surge in the concept of immersive input devices. One form of input that has to be fully utilised is a mapping the joint movement of a hand. This project will prototype a device that is able to track the joint movement in a hand and send this information to a computer for use with a 3D environment. This has applications for within both gaming and medical purposes, either in VR games or medical functions such as assisting in stroke rehabilitation. Research and testing will be undertaken to identify different methods of tracking joint movements within the hand. A small prototype for a single joint will then be evaluated followed by a larger prototype for the whole hand. The effectiveness of the device will be evauated by communicating with a client with knowledge in the field for the medical aspect of the device and comparing it against other similar devices that currently exist for similar purposes

## External Ethics Review

| | |
|---|---|
| **Does your research require external review through the NHS National Research Ethics Service (NRES) or through another external Ethics Committee?** | No |

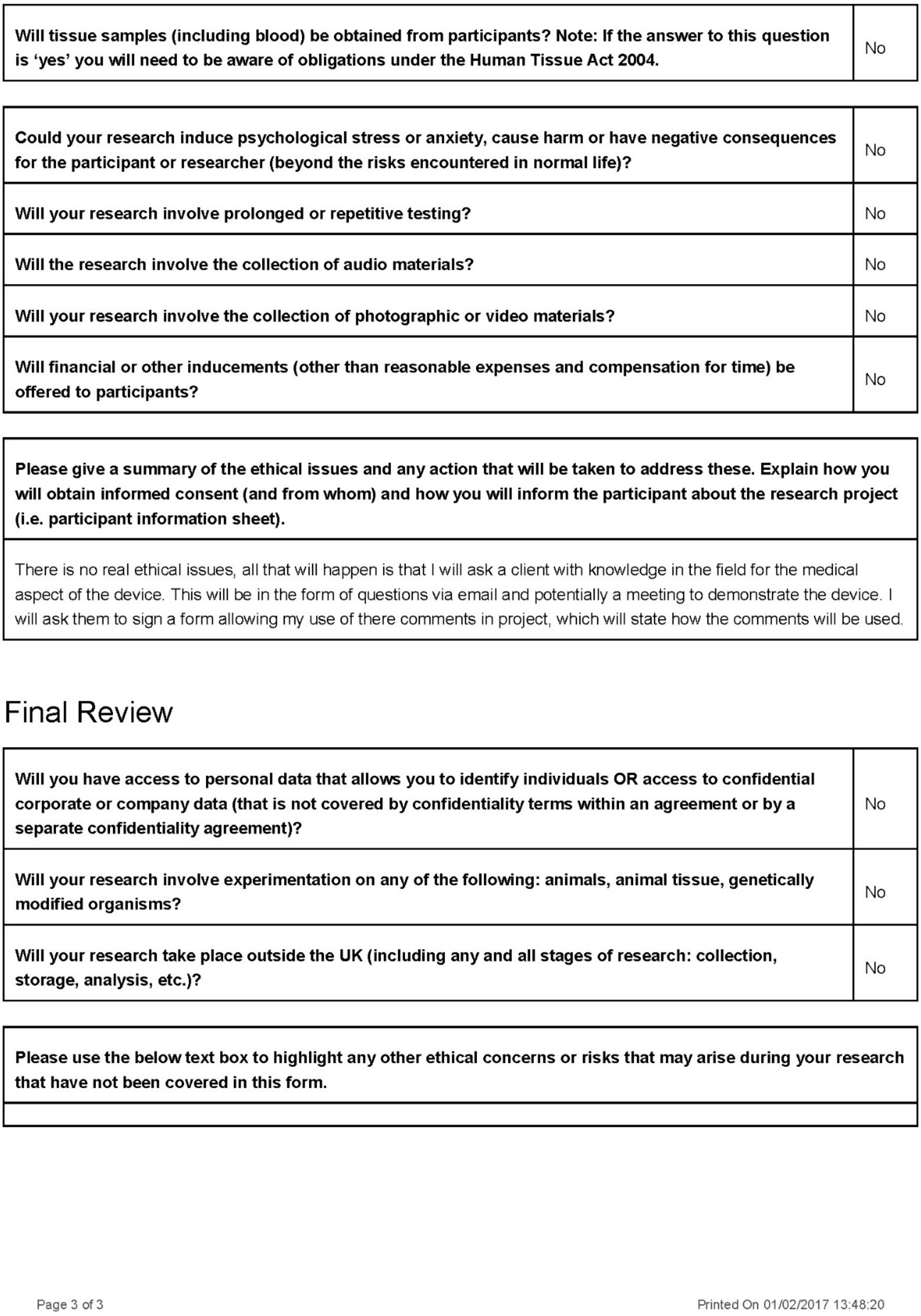
## Research Literature

| | |
|---|---|
| **Is your research solely literature based?** | No |

## Human Participants

| | |
|---|---|
| **Will your research project involve interaction with human participants as primary sources of data (e.g. interview, observation, original survey)?** | Yes |
| **Does your research specifically involve participants who are considered vulnerable (i.e. children, those with cognitive impairment, those in unequal relationships—such as your own students, prison inmates, etc.)?** | No |
| **Does the study involve participants age 16 or over who are unable to give informed consent (i.e. people with learning disabilities)? NOTE: All research that falls under the auspices of the Mental Capacity Act 2005 must be reviewed by NHS NRES.** | No |
| **Will the study require the co-operation of a gatekeeper for initial access to the groups or individuals to be recruited? (i.e. students at school, members of self-help group, residents of Nursing home?)** | No |
| **Will it be necessary for participants to take part in your study without their knowledge and consent at the time (i.e. covert observation of people in non-public places)?** | No |
| **Will the study involve discussion of sensitive topics (i.e. sexual activity, drug use, criminal activity)?** | No |

| | |
|---|---|
| **Are drugs, placebos or other substances (i.e. food substances, vitamins) to be administered to the study participants or will the study involve invasive, intrusive or potentially harmful procedures of any kind?** | No |



| | |
|---|---|
| **Will tissue samples (including blood) be obtained from participants? Note: If the answer to this question is 'yes' you will need to be aware of obligations under the Human Tissue Act 2004.** | No |

| | |
|---|---|
| **Could your research induce psychological stress or anxiety, cause harm or have negative consequences for the participant or researcher (beyond the risks encountered in normal life)?** | No |
| **Will your research involve prolonged or repetitive testing?** | No |
| **Will the research involve the collection of audio materials?** | No |
| **Will your research involve the collection of photographic or video materials?** | No |
| **Will financial or other inducements (other than reasonable expenses and compensation for time) be offered to participants?** | No |

| |
|---|
| **Please give a summary of the ethical issues and any action that will be taken to address these. Explain how you will obtain informed consent (and from whom) and how you will inform the participant about the research project (i.e. participant information sheet).** |
| There is no real ethical issues, all that will happen is that I will ask a client with knowledge in the field for the medical aspect of the device. This will be in the form of questions via email and potentially a meeting to demonstrate the device. I will ask them to sign a form allowing my use of there comments in project, which will state how the comments will be used. |

## Final Review

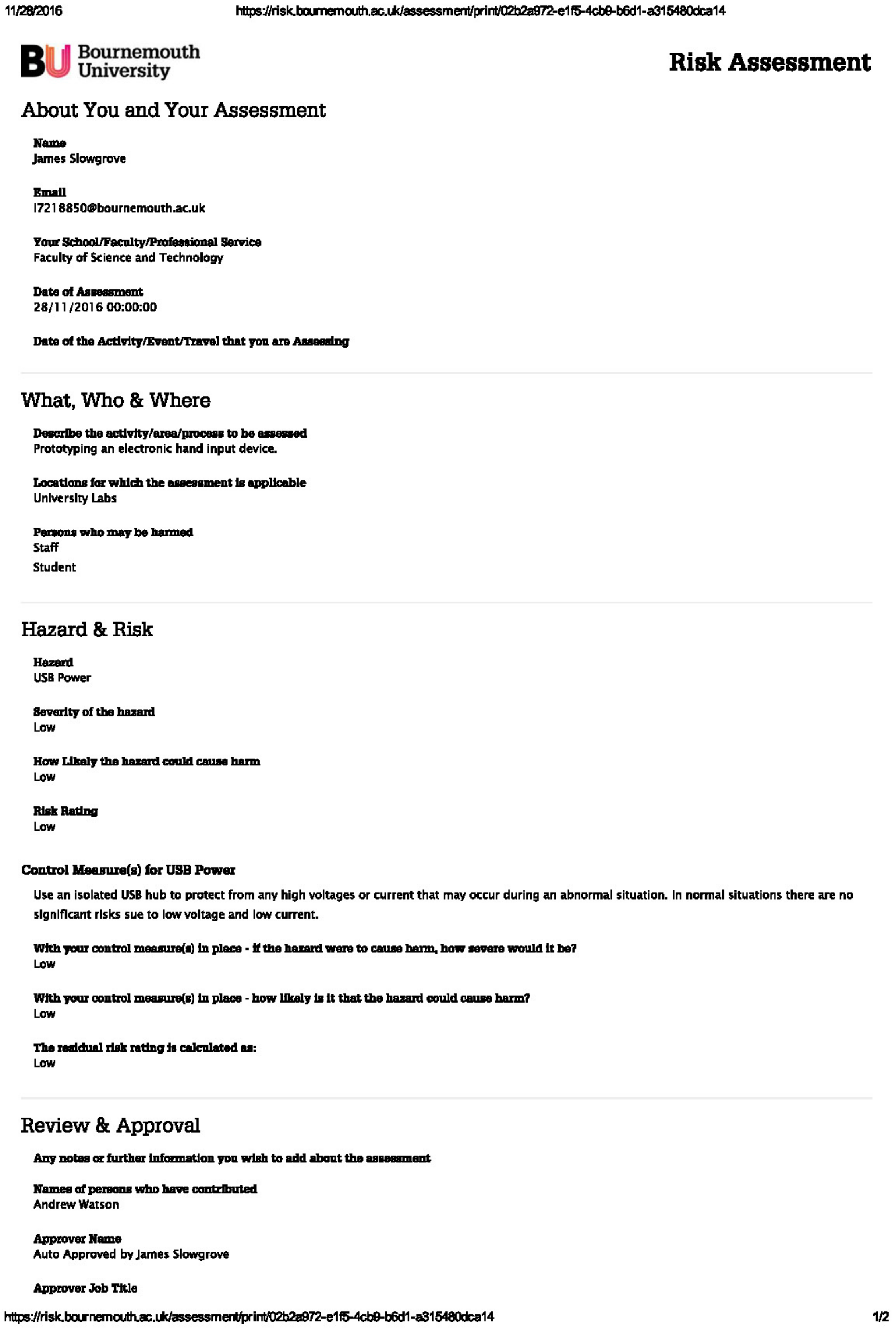
| | |
|---|---|
| **Will you have access to personal data that allows you to identify individuals OR access to confidential corporate or company data (that is not covered by confidentiality terms within an agreement or by a separate confidentiality agreement)?** | No |
| **Will your research involve experimentation on any of the following: animals, animal tissue, genetically modified organisms?** | No |
| **Will your research take place outside the UK (including any and all stages of research: collection, storage, analysis, etc.)?** | No |

| |
|---|
| **Please use the below text box to highlight any other ethical concerns or risks that may arise during your research that have not been covered in this form.** |
| |

Page 3 of 3

Printed On 01/02/2017 13:48:20

# Appendices 2 – Risks



11/28/2016 https://risk.bournemouth.ac.uk/assessment/print/02b2a972-e1f5-4cb9-b6d1-a315480dca14

Bournemouth University

## Risk Assessment

### About You and Your Assessment

**Name**
James Slowgrove

**Email**
I7218850@bournemouth.ac.uk

**Your School/Faculty/Professional Service**
Faculty of Science and Technology

**Date of Assessment**
28/11/2016 00:00:00

**Date of the Activity/Event/Travel that you are Assessing**

### What, Who & Where

**Describe the activity/area/process to be assessed**
Prototyping an electronic hand input device.

**Locations for which the assessment is applicable**
University Labs

**Persons who may be harmed**
Staff
Student

### Hazard & Risk

**Hazard**
USB Power

**Severity of the hazard**
Low

**How Likely the hazard could cause harm**
Low

**Risk Rating**
Low

**Control Measure(s) for USB Power**

Use an isolated USB hub to protect from any high voltages or current that may occur during an abnormal situation. In normal situations there are no significant risks sue to low voltage and low current.

**With your control measure(s) in place - if the hazard were to cause harm, how severe would it be?**
Low

**With your control measure(s) in place - how likely is it that the hazard could cause harm?**
Low

**The residual risk rating is calculated as:**
Low

### Review & Approval

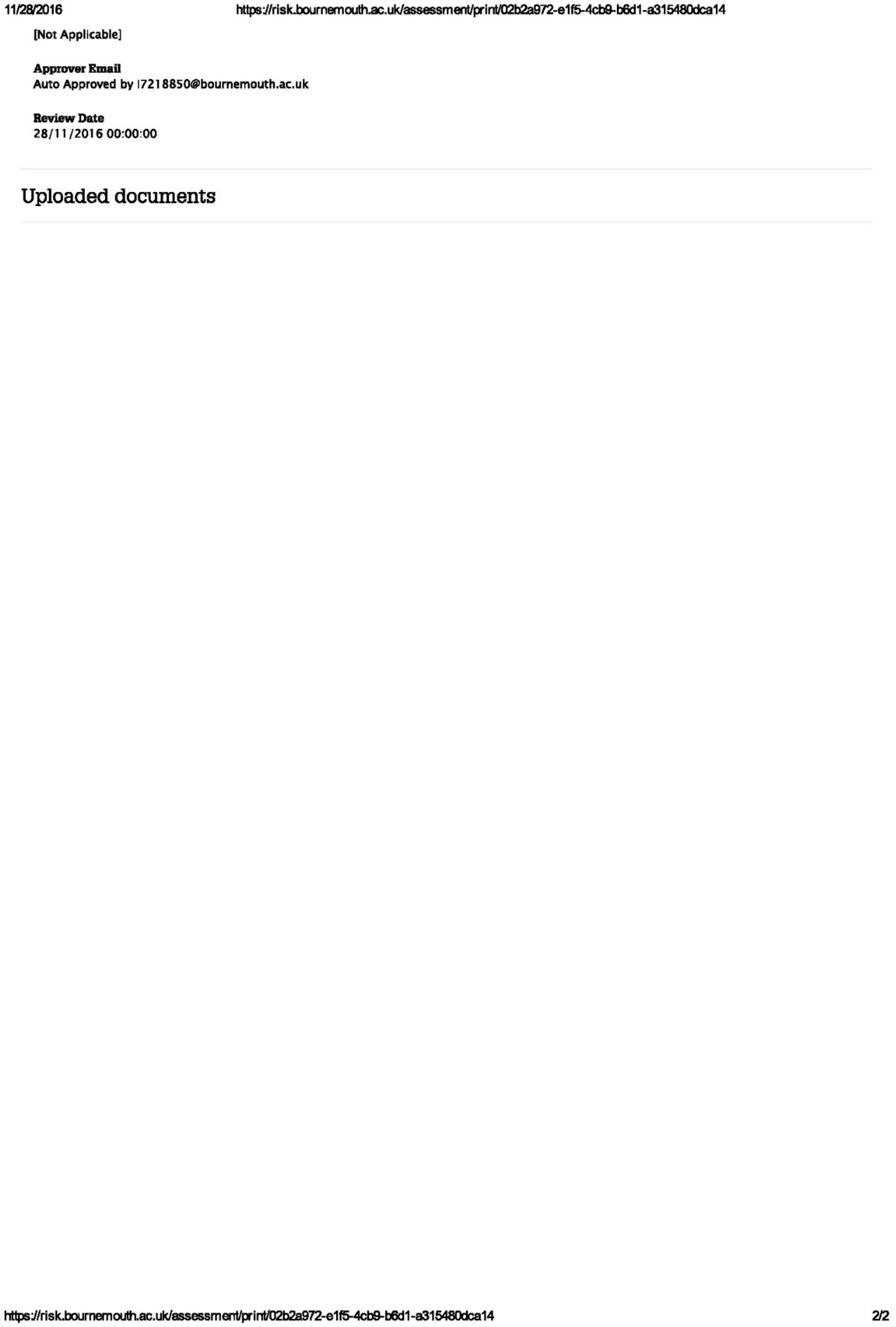
**Any notes or further information you wish to add about the assessment**

**Names of persons who have contributed**
Andrew Watson

**Approver Name**
Auto Approved by James Slowgrove

**Approver Job Title**

https://risk.bournemouth.ac.uk/assessment/print/02b2a972-e1f5-4cb9-b6d1-a315480dca14 1/2



11/28/2016 https://risk.bournemouth.ac.uk/assessment/print/02b2a972-e1f5-4cb9-b6d1-a315480dca14

[Not Applicable]

**Approver Email**
Auto Approved by i7218850@bournemouth.ac.uk

**Review Date**
28/11/2016 00:00:00

## Uploaded documents

https://risk.bournemouth.ac.uk/assessment/print/02b2a972-e1f5-4cb9-b6d1-a315480dca14 2/2

# Appendices 3 - Trello Log Book

To see the Trello log book please view the online Trello board or if that is unavailable there is an offline version include with the artefact hand in.