University of Plymouth

Faculty of Science and Engineering

School of Computing, Electronics and Mathematics

PROJ 325:

Haptic Feedback using Shape Memory Alloys

Ben Wickenden

Student ID: 10557913

BSc (Hons) Robotics

1 Abstract

This paper covers the use of shape memory alloys being implemented into a Haptic Glove to provide physical feedback to the user. The shape memory alloy being used is Flexinol, which is a brand of Nitinol, a wire that's an alloy made from Nickel and Titanium. I've designed a 3D printed exoskeleton that attaches the Nitinol wire to the glove, when activated the Nitinol provides a suggestive force onto the user, simulating the user bending a finger. The Haptic Glove also has a secondary feedback which is provided by six coin-cell vibration motors. The feedback is controlled by a simulation created in Unity that sends a message to an Arduino when a key is pressed, that then triggers a feedback response. Whilst the project is success, I later explore the potential use for Nitinol to be used in the rehabilitation of stroke survivors and the use of Nitinol as a replacement actuator in Robotic hand.

2 Acknowledgments

I would like to thank and acknowledge Martin Simpson, Jake Gibson Shaw-Sutton and James Rodgers for their help and support along the way with this project and unrivalled knowledge and passion for their craft.

Paul Davey for being a supportive personal Tutor

David Jenkins (my project supervisor) for guiding me through this project and offering his advice and ideas along the way.

3 Table of Contents

Contents

4 Introduction

For my project I wanted to create a Haptic Feedback device that would provide a different type of sensation than the current directional rumble motors that are used in gaming controllers today [PlayStation (2019)] [Xbox (2019)] with the projects main focus and inspiration being the games industry. After weighing up my options, such as creating a Haptic device using heating and cooling elements or the use of piezo electrics, I decided to use a shape memory alloy to be the main application of physical feedback. I also decided that the medium that I would provide the force would be in the form of the glove. I wanted to provide a force on the fingers of the user that would provide the sensation as if something was applying a force to them. One of my original ideas for an application of this device could be for people to be able to communicate via touch over a video call. In the early stages of the project my plan was to create spring out of a shape memory alloy that which when activated would provide a compression force to the user's hand. However, later on in the project I decided to create a glove with an exoskeleton that would then have Nitinol wire attached to the underside of the hand that would then contract providing a force to the user. The feedback would be controlled by a simulation in Unity that would send signals to an Arduino to activate the feedback. With Haptic feedback being the focus of my project, I explored where the demand for devices that provided a form of physical feedback came from and explored how its evolved over the past two decades looking at both the development of console controllers as well as the implementation of more advance Haptics that are being integrated into Virtual Reality environments.

5 Aims and Objectives

The main focus of my project is to create a Haptic Feedback device that could would apply a physical force onto the user that could be used to improve a user's immersion and experience whilst playing a video game, as well as having a series of other potential applications. The device will be in the form of a glove and my primary force that will be providing the feedback will be Nitinol wire, which is a shape memory alloy and will allow the user to feel a suggestive force onto the user's fingers that will simulate the fingers movement. The secondary force feedback will be provided by five DC coin cell vibration motors and will be placed on knuckle of each finger and thumb. To demonstrate the haptic glove, I will create a simulation in the Unity engine that will allow me to control the glove by sending strings to an Arduino which will then trigger for a feedback module to be fired. I will also be using Blender to create 3D assets for the simulation and Auto Desk Fusion 360 to create the stl files to 3D print components for the glove.

Objectives

- Create a Simulation in Unity that can output Data to Arduino
- Design the Physical glove that the user will wear
- Control motors using Data from the Simulation
- Create a Shape Memory Alloy that can be used to provide a force onto a hand
- Control the Shape Memory Alloy using the data provided to the Arduino from the Unity simulation and to have all systems working together

6 Project Planning and Management

In order to plan and keep on track of my project I implemented a few different methods of project management. I used a series of Gantt charts throughout my project, with the original being a forecast of the work ahead [Figure1], the second being a revaluation of the progress that I had made and changing my planning based on this [Figure 2], and the final being a closer representation of what happened [Figure 3]. At the end of the project I created a Gannt chart that was a comparison of the original forecast and the final [Figure 4]. From the chart I can see that I was unrealistic in the expectations on what I hoped to achieve before the module started (28^{th} January 2019), as I originally planned to have the Unity simulation completed and to be have already selected a shape memory alloy to begin testing. In actuality I found that it took me a lot longer to learn how to use software such as Blender and Unity than I had previously allowed for. In addition, I hadn’t taken into consideration the selection process that I went through to choose the Nitinol wire that I wanted to use in my project, nor the shipping times or 3D printing times that would restrict how quickly I could progress. Overall the use of Gantt charts was useful to make sure that I was keeping up with the work load, as a measure of when I should have achieved progress in my project, however it’s difficult to predict what may go wrong in the future and so they aren’t to be followed exactly. Whilst the timings were not correct all the tasks on the original gantt chart were completed.

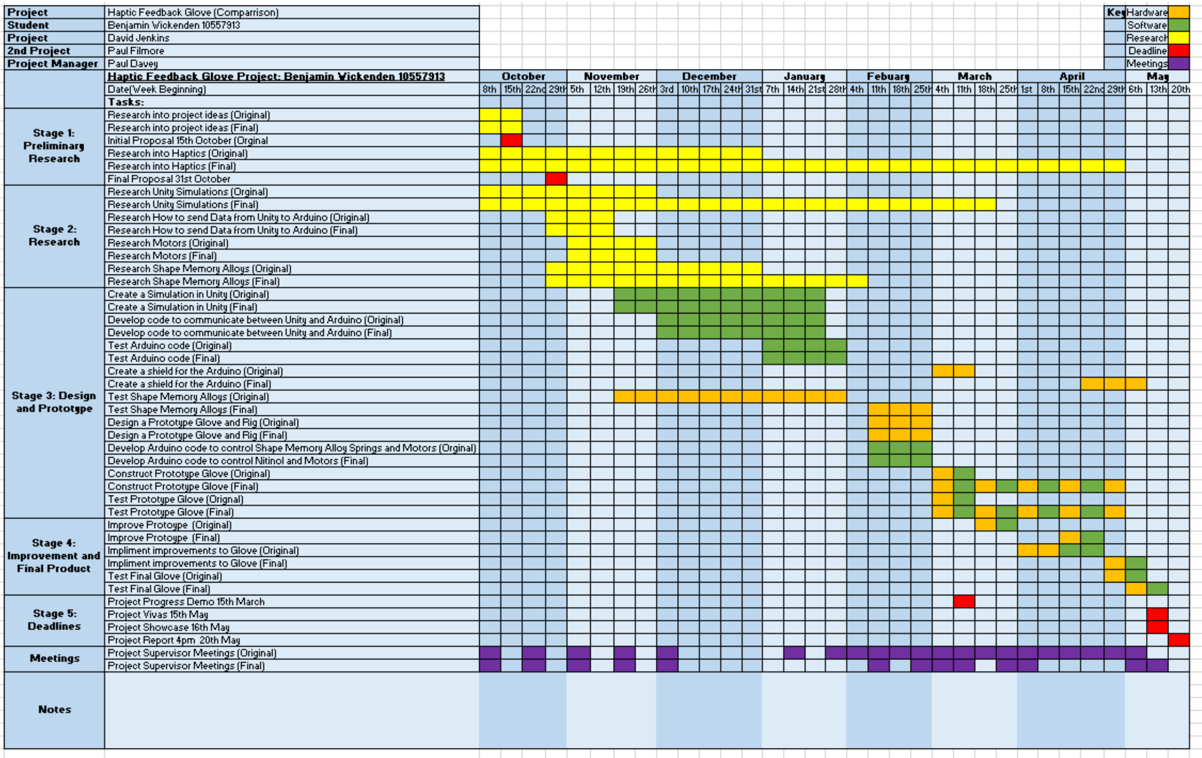


Figure 4: Shows comparison between original and final Gantt charts

Another method that I used was Microsoft planner, which I used to set a series of tasks which could be placed in an order of importance, with comments on what has been achieved for each task and what could be done next to improve the given task if needed. [Figure 5]. Whilst I had previously found Microsoft planner with its Kanban style of management quite useful in the past when working in groups for a project, I found that for my own personal project Microsoft planner wasn't as useful as other methods such as notes in my Lab book to note how tasks went and what I need to do next. Whilst it does have its benefits with task designation and keeping notes for each task together, it's not as accessible as my log book in comparison.

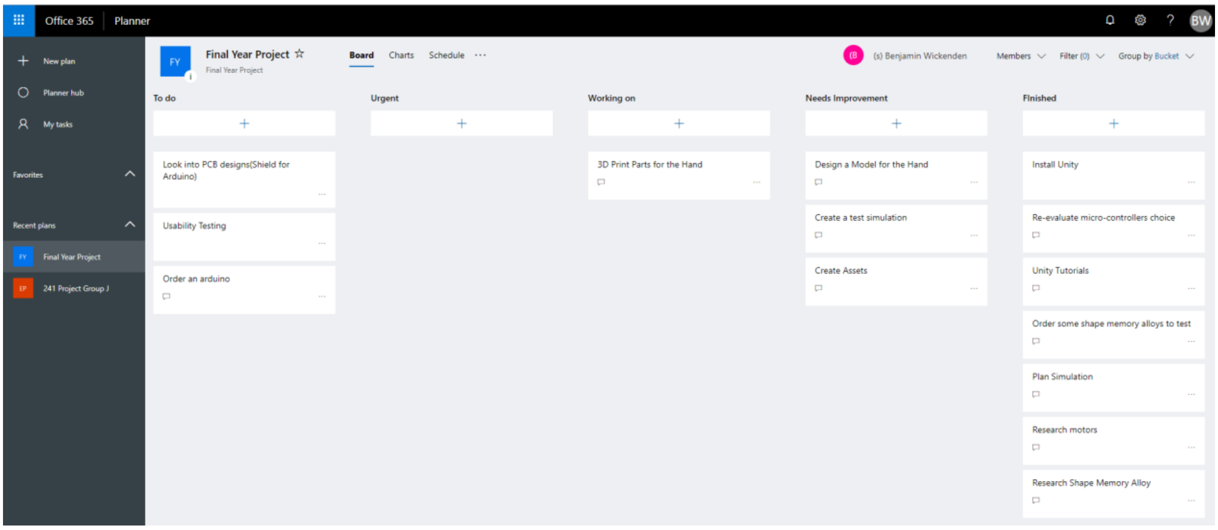


Figure 5: Microsoft planner

To keep my code organised I created a GitHub repository [Figure 6], where I uploaded various versions of my code for Unity, Blender files, Arduino code, stl files and documentation. I found GitHub very useful throughout my project as it provides code versioning and allows me to make changes to code via the repo without needing to open an IDE (Integrated Development Environment), as well as being able to store other file types.

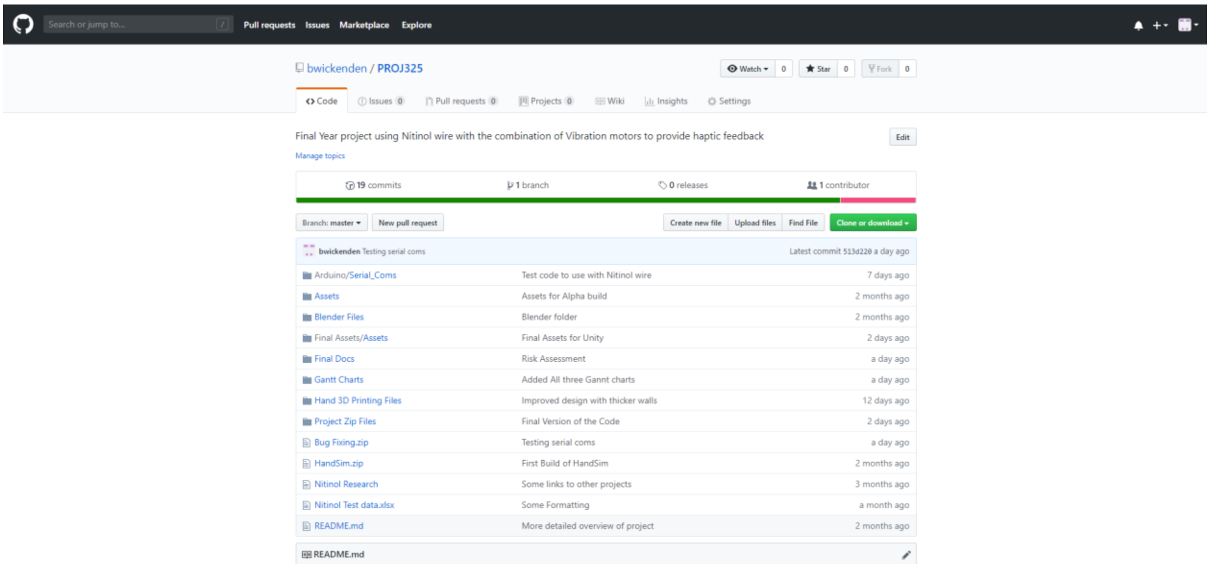


Figure 6: GitHub Repo

Lastly, I kept a paper Journal as a log book, which I then uploaded to a digital copy. The log book was my most important method of planning and management as it was always readily available and contains my real-time notes on the project. With my log book I was able to use it to take notes on what tasks needed to be done, with notes on how a task went and what needs to be done going forwards. I also used the lab book to write guides on how to use a software such as Blender and Unity, which I could use when returning to previous tasks.

6.1 Costing

For my project I had the budget of £100 to spend in total I spent £41.98 of the budget and spent an additional £12.54 of my own money. The reason for the personal cost was purely out of convenience.

Budget

Nitinol wire 5m: £28

5x LM317: £0.19

4x 9V battery: £3.92

Veroboard: £0.86

Resistors: Free Stock

ZTX651 Transistors: Free Stock

Arduino Uno: Borrowed from Uni

Delivery: £9.01

Total Budget spent: £41.98

Personal cost

8 x DC Coincell Vibration Motors: £6.55

USB Type B Cable: £5.99

Total Personal cost spent: £12.54

7 Industry Background

Haptic feedback has been in games for years to enhance the users experience by helping the user to feel more immersed in the environment around them. In April 1997 Haptic feedback was first brought to household consoles with the introduction of the Rumble Pak for the Nintendo N64 controller. [Nintendolife (2013)] This was the first time that a controller had provided the user with a form of physical feedback. With the success of the N64 Rumble Pak, Nintendo's main competitors Sony (PlayStation) and Microsoft (Xbox) had to develop their own gaming controllers that utilised Haptics in order to keep up with the trend that had now been set. Sony's answer to the N64 Rumble Pak was the Dual shock controller in November 1997, which implement two rumble motors to already in use Dual Analog controller. The way in which the controller provides feedback was by a small motor spinning a series of unbalanced weights attached to an axel which caused the controller to vibrate. Since 1997 Sony have released three more versions of the Dual Shock controller which brings us the present-day Dual Shock 4 controller which is being used alongside the PlayStation 4. [PlayStation (2019)] Through innovation of the original design, the Dual Shock 4 now uses two analogue motors which allow developers to provide feedback to the user at varying intensities, helping to improve the users experience. [Push Square (2019)] This differs from previous models of the Dual Shock controllers which used digital motors or a combination of both.

Keeping with the industry standard Microsoft also started to develop its own controller that made use of haptics for the Xbox gaming console. The controller was launch alongside with the consoles release on 2001 and was nicknamed the "Duke" due to its size. The Original Xbox controller implemented two Rumble motors in the main body of the controller and this is a feature that was later passed onto the Xbox 360 controller and to current generation controllers for the Xbox one. Microsoft has acknowledged the importance of Haptic feedback to the users of its product as an essential part of the controller's design, that is shown by the further development and customisability that the Xbox One controllers now have. The most recent Xbox One controller implements impulse triggers which allow the user to experience directional feedback whilst playing games and can also provide feedback when pressing the on the dual triggers. [Xbox (2019)] In addition to this Microsoft has allowed the user to control the settings of the Rumble feedback from the controller via an app. This allows the user to tailor the feedback to their own personal requirements, enhancing each individual experience with the controller.

Whilst Microsoft's main focus when it comes to the gaming industry is the Xbox consoles, they also recognise that the push for interaction with an environment in a Virtual Reality setting is something that is in high demand. Thus, Microsoft has several research products that are aiming to provide a fully immersive experience in a VR setting which allows the user to interact with the environment around them as if it was reality. One of their projects is called Haptic Links which has the aim of proving a variable stiffness controller. [(Evan Strasnick, Christian Holz, Eyal Ofek, Mike Sinclair, Hrvoje Benko). (2018)] This device works by taking two Virtual Reality controllers and joining them together with an electro-mechanically actuated physical connector. The device has the ability to change the resistance between the two controllers, with an example being that they can have a user drawing back a bow and arrow. As the user draws back the string of the bow, the movement of the link becomes increasingly stiffer which simulates the increasing amount of force required to keep pulling the bow string back. Once the box string is released the stiffness of the bow returns to its free state. The project has the aims of proving an immerse experience in a Virtual Reality by mimicking the physics that would be expected to be implemented onto the user whilst taking part in the same activities in reality.

As well as large companies such as Microsoft are looking to develop new ways of providing Haptic Feedback to a user, smaller companies have also undertaken the challenge. A small company under the names TN games has developed a Haptic Vest that that be used with a variety of games and even films. The vest has eight pneumonic pistons which are used to provide the force for the feedback. The vest uses a small air compressor to the drive the pistons in the vest whenever the user has an interaction within the virtual environment that provide a contact force. An example of this would be if the user was playing a first-person shooter, the vest would provide directional feedback of where the character model had taken damage with the impact being represented by the impact of the piston. [TN Games (2019)] The vest, whilst being quite primitive is very effective at improving the user's immersion as it provides an intuitive feedback that a hand-held controller would not be able to translate.

Another company that is trying to push the boundaries of Haptic Feedback is a company called TeslaSuit. They have developed a full body Haptic Feedback suit that covers the whole body of the users, similar to how a wet suit would fit. Tesla's suit has four main features, it uses Twenty Haptic feedback point of contact that can be used to transmit Haptic sensations. They incorporate full body motion capture with eleven points track points that allow the user to be have their motion tracked to high precision, which can then be used to animate their avatar in the given environment. The suit also has full body climate control which allows the user to feel the temperate change that their character would be experiencing, further immersing them into the digital environment around them. Finally, the suit has a biometric system that can collect data from the users, which TeslaSuits have plans to use with machine learning to further develop their product. [TeslaSuit (2019)] TeslaSuits pitches their product as a Virtual Reality multitool. On one hand TelsaSuits pitch their product as something that you would find in a gaming arcade that would allow a user to become immersed in an interactive digital environment, but on the other hand TeslaSuits recognises that there is a growing market in applying Haptics with VR beyond the gaming industry. [TeslaSuit business (2019)] Companies such as BioflIightVR [BIOFLIGHTVR (2019)] and Provr [PROVR (2019)] provide a service where they create simulations of a work environment which is then used to train inexperienced employees. These simulations are a cheaper and safer way to train employees to undertake tasks whilst eliminating potential risk to the user of the environment around them. BioflightVR specialising is creating interactive medical simulations where the user could learn to safely perform new or practice complex operations without there being a potential risk to life.

Whilst some companies focus on providing a whole-body experience, others have set themselves with a goal of creating smaller devices that would be more user friendly that could be used within the average household. HaptX have developed a glove that allows the user to feel the shape and texture of objects in a 3D environment by using 130 points of feedback. The glove uses a lightweight exoskeleton that can apply up to four pounds of force to the user in the form of resistance. The glove also utilises magnetic motion tracking which provides a highly accurate interaction between the user and 3D objects. [Haptx (2019)] Another device that's being developed by researchers at the Simon Fraser University is the Flex-N-Feel glove which purpose is to allow users to have interactions whilst using some form of communication service. [4] The idea is that it provides a way for people who are in long distance relationships or friend ships to stay connected by emulating the touch of a hand. [New ATLAS (2017)]

From my research into the Haptics industry I have discovered how the market and demand has grown over the past two decades, a journey that began with a simple rumble Pak in the N64 controller has now evolved into devices that can make us believe that we are in completely different environments. Haptics have developed to the point where we can have realistic interactions with 3D object as seen with the HaptX gloves, or to make us feel like we are in the snow, wind or summers heat with the TeslaSuit. The market in the present day is still expanding with the production of more Haptic devices that are both intuitive to the user and create high levels of immersion, however as more precise devices are developed they are branching off from the games industry and being applied to more serious applications such as training for hazardous jobs or surgical operations. For my project I have decided to focus providing Haptic feedback localised to the hand, with the goal allowing the hand to be interacted with rather than for the hand to interact with the environment.

8 Theory

8.1 Mathematics

8.1.1 Equation 1

Ohms Law where V = Volts, I = Current, R = Resistance:

$$V = IR$$

Rearranged to find Current: $I = \frac{V}{R}$

For Voltage over Nitinol Wire:

R = 7.7 Ω, Current = 0.4A

$$V = 0.4 \times 7.7$$

V = 3.08V

For the Resistor value going form the Arduino output pins to the base pin of the transistor.

V over transistor = 3.3V- 0.6V = 2.7V

I = 4mA

$$R = \frac{2.7}{0.004}$$

R = 1.35k Ω

8.1.2 Equation 2

Precision Current- Limiter Circuit. The equation is used to calculate R2 in relation with the desired current limit as 400mA. With I = Current(A), R2 = Resistance (Ω)

$$I\ limit = \frac{1.2}{R2}$$

8.1.3 Equation 3

Power equation. Where Power = Watts(W), R = (Resistance(Ω), I = Current(A)

$$Power = R \times I^2$$

For the LM317:

Resistance = 3Ω, Current = 0.4A

$$Power = 3 \times 0.4^2$$

Power = 1.2W

8.1.4 Equation 4

Thermal Equation for the LM317 Circuits. Operating temperature = Celsius (^{0}C), Thermal Resistance = Resistance multiplier per Watt (Ω), Power= Power going through circuit (W), Room temperature = Average temperature of room (^{0}C)

$$\text{Operating temperature} = (\text{Thermal Resistance} \times \text{Power}) + Room\ temperature$$

For the LM317:

Thermal Resistance = 23.5Ω, Power = 1.2W, Room Temperature = 25^{0}C

$$Operating\ temperature = (23.5 \times 1.2) + 25$$

Operating temperature = 53.2^{0}C

8.1.5 Equation 5

Stress of the Nitinol Wire

$$Stress = \frac{Force}{Cross\ sectional\ Area}$$

Where

$$Force = mass\ \times acceleration$$

Force = Newtons(N), mass = Mass of penny weights (kg), acceleration = Acceleration due to gravity = 9.81 (m/s)

Where

$$Cross\ Sectional\ Area = \pi r^2$$

r = radius of Nitinol wire = 75x$_{10}^{-6}$m

8.1.6 Equation 6

Strain of the Nitinol Wire. Extension (m), Original Length (m)

$$Strain = \frac{Extension\ of\ wire}{Original\ Length}$$

8.1.7 Equation 7

Resistance of parallel resistors, R = Resistance (Ω)

$$\frac{1}{R\ total} = \frac{1}{R1} + \frac{1}{R2} + .. + \frac{1}{Rn}$$

8.2 Nitinol

For my project I wanted to implement Nitinol wire to be used as an actuator to provide a force to the user, simulating a contact force. Nitinol stands for Nickle Titanium Naval Ordnance Lab and has shape memory and superelastic properties. This means that the Nitinol can experience deformation at one temperature, but then return to its original shape once heated. The two phases of Nitinol are called the Martensite and Austenite phases, where the Martensite phase is the alloy at resting temperature and the Austenite phase is when the alloy is at the state where it has transitioned after heat being applied. [Chemistry Learner (2019)] At low temperatures the Martensite structure is stable, in this state the alloy is easily deformed by an external force. However, when the alloy increases in temperature its Austenite structure starts to become stable. Once the alloy has transitioned to its Austenite state the crystalline structure of the alloy becomes uniform, returning the alloy to its original state, it is this ability that gives the alloy its shape memory name. [(Jaronie Mohd Jani, Martin Leary, Aleksander Subic, Mark A. Gibson). (2013)]

The process of changing shape by changing state makes shape memory alloys ideal to be used as light weight actuators that can be used on a micro scale, with one use being in biometrics with using shape memory alloys as artificial muscles. [(Jaronie Mohd etc.). (2013)] [(H. Fujita). (1989)] With this is mind, it makes Nitinol an ideal choice to be used when trying to simulate a force on the users' fingers.

8.3 Choices for Circuits

Due to the Nitinol wire requiring a large current to cause it to transition from its Martensite phase to its Austenite phase I couldn't use the Arduinos PWM output pins to directly power the wire. Therefore, I had to implement a way to provide 400mA to the Nitinol wire. Originally, I was just going to use a power supply, however I wanted to make the Haptic Feedback devices only wired connection the input via the USB Type B cable that powers the Arduino. This would then allow the device to be operated in a home environment. To do this I needed to implement a current regulator circuit. My choice of component to achieve this was a LM317 3-Terminal Adjustable Regulator. With the resistance of the Nitinol wire being 7.7Ω it means that 3.03V being supplied to the Nitinol wire in order to provide 400mA [Equation 1]. The LM317 has several key features that are essential for my project: [Texas Instruments (2016)]

- Input to output differently voltage: Min 3V and max 40V which fits the parameters from my circuit.
- Output Current: Min 0.01A and Max 1.5 A
- Thermal Resistance 23.5. With the power of the circuit being 1.2W [Equation 3] The temperature of the circuit would run at 53.2^{0}C [Equation 4]
- The Operating Virtual temperature junction temperature is Min 0^{0}C and Max 125^{0}C

After looking what the LM317 was capable of I decided to implement it into my circuit as it met all my requirements. Whilst I could have calibrated the circuit at a higher Current, I wanted to keep the temperature of the circuit down to avoid over heating as well as stress on other components.

With the Nitinol now using an external current circuit I needed a way to be able to control the supply to the Nitinol wire. To do this I implemented a transistor circuit. The chosen transistor would need to be able to operate with 400mA at a low voltage. The transistor that I selected was the ZTX651 which had the following features:

- Max Emitter-Base Voltage 5V
- Peak Pulse Current 6A
- Continuous Collector Current 2A
- h_{FE} of 200

The ZTX651 fulfils all the requirements that I needed it for. The current that I will be using it with will be 400mA and the Emitter-Base Voltage is less than 5V. With a back voltage of 0.6V, I connected with transistors base pin to the Arduinos output pins with a 1.35kΩ resistor in series to provide a 4mA signal [Equation 1].

8.4 3D printing

I created several 3D printed components to make the exoskeleton for the Haptic Feedback Glove. I used Auto desk Fusion 360 to create each component in a 3D digital space. Once I had the required model I could then export the file to a .stl file, which could be sent to 3D printer. All of my components were printed using a PLA filament. Although the melting point for PLA is 145-160^{0}C I needed to ensure that the walls of each component were think enough to withstand the heat of the Nitinol Wire, whilst still maintain a sleek profile [Farnell PLA (2017)].

8.5 Arduino

For my choice of microcontroller, I decided to use an Arduino Uno [Farnell Arduino (2019)] due to its small size and that its very simple to send data to via the serial port. The Arduino Uno also has enough output pins for my project. Although the Clock isn’t the fastest available in the market, the code that I’m wring doesn’t require a large amount to computing power.

- Operating Voltage 5V
- 14 Digital I/O pins
- DC Current per I/O Pin: 40mA
- DC Current for 3.3V Pin: 50mA
- 16 MHz clock
- In built UART serial communication
- Size 75mm x 50mm
- USB Type B cable 1.5A

9 Design

My design was to have a glove with an exoskeleton made out of 3D printed parts attached to it that would support a Nitinol wire on each finger. The wire would be toggled on and off at the press of a key in the Unity simulation. When the key is pressed it will activate the animation for the required finger and will then send a string to the Arduino over the serial port. This then toggles a PWM pin on the Arduino that send a signal to a transistor that has a power source providing 400mA connected to the Nitinol wire. In addition, a key press in the Unity simulation can also trigger a vibration motor which works in the same way but is powered directly from the Arduino.

9.1 The Glove

In order to create a Haptic Feedback glove, I first had to select a glove that I could use the foundation of my design. Initially I opted to use a heavy works glove [Figure 7] as I was very conscious that the Nitinol wire would be operating at 90^{0}C, and I was concerned about a large heat transfer to the user's hand. However, after testing I found that there was a negligible heat transfer to the user and that the original glove was too thick and cumbersome, with it affecting the user's movement and ability to feel the subtle suggestive feedback.

Figure 7: Original choice for glove (Workers Glove)

After initial testing I decided that the workers glove was not suitable for my project and needed to be replaced with a glove that has greater flexibility and was more comfortable for the user to wear, along with being able to not interfere with the haptic feedback acting on the user. After looking at other types of gloves such as cycling and golf gloves, I came across wetsuit gloves which was made out a flexible elasticated material that meet all the requirements. [Figure 8]

Figure 8: Wetsuit glove

With an appropriate glove selected I then used Auto Desk fusion 360 to design a series of models that would be 3D printed to create the exoskeleton. I designed four components that would be attacked to the glove to make the full exoskeleton that would hold the Nitinol wire in place. Originally, I planned to have three pieces per a finger but found that the design worked much better with just the two pieces on each finger, with one being positioned on both knuckles. I created two designs for the finger tips, with the first having a rounded tip that would encase the top of the user's finger and a groove on one side that was designed to prevent a restriction in the user's fingers. [Figure 9] The second had an open top design which retained access the user's fingers which would allow them to use a mouse and keyboard whilst wearing the gloves [Figure 10]. In addition, I added two loops that the Nitinol wire could be threaded through and I added four holes (Two mirrored on each side) that would be used to attaching the 3D printed piece to the glove by stitching. For the middle sections of the fingers I took the design for the open top finger tips and removed the bevel [Figure 11]. This design became my final designed as I concluded that there was no need for a fingertip segment and only two pieces on each finger gave greater mobility which still proving the haptic feedback.

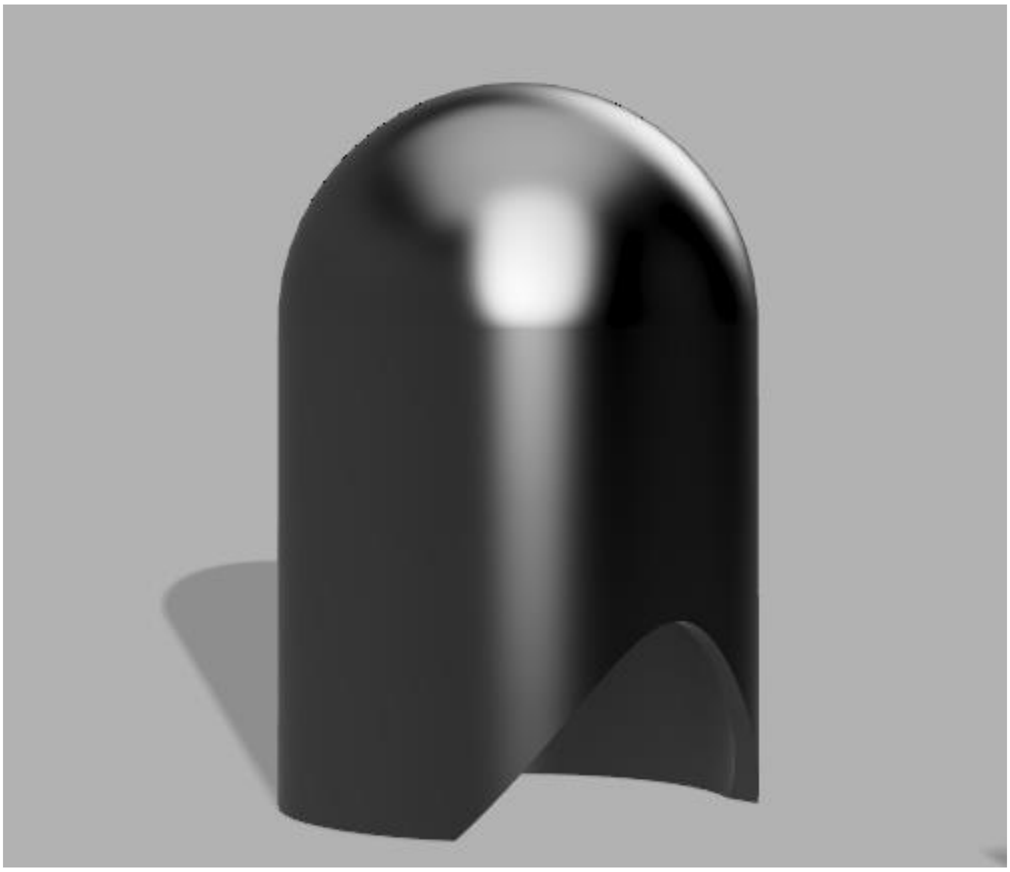


Figure 9: Fingertip model with rounded top

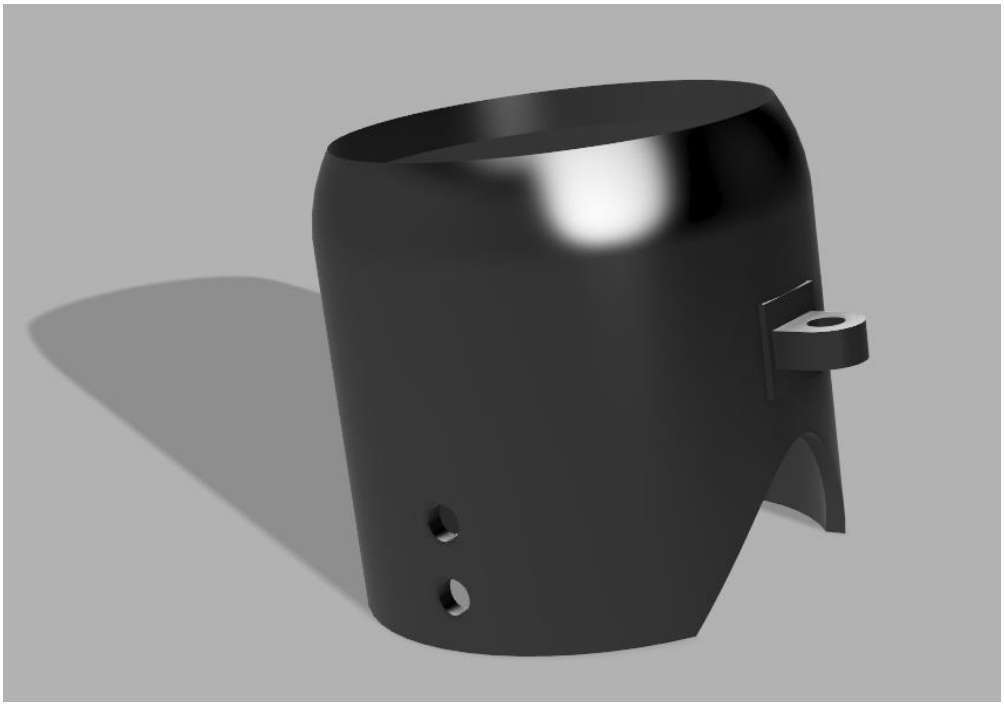


Figure 10: Fingertip, open top

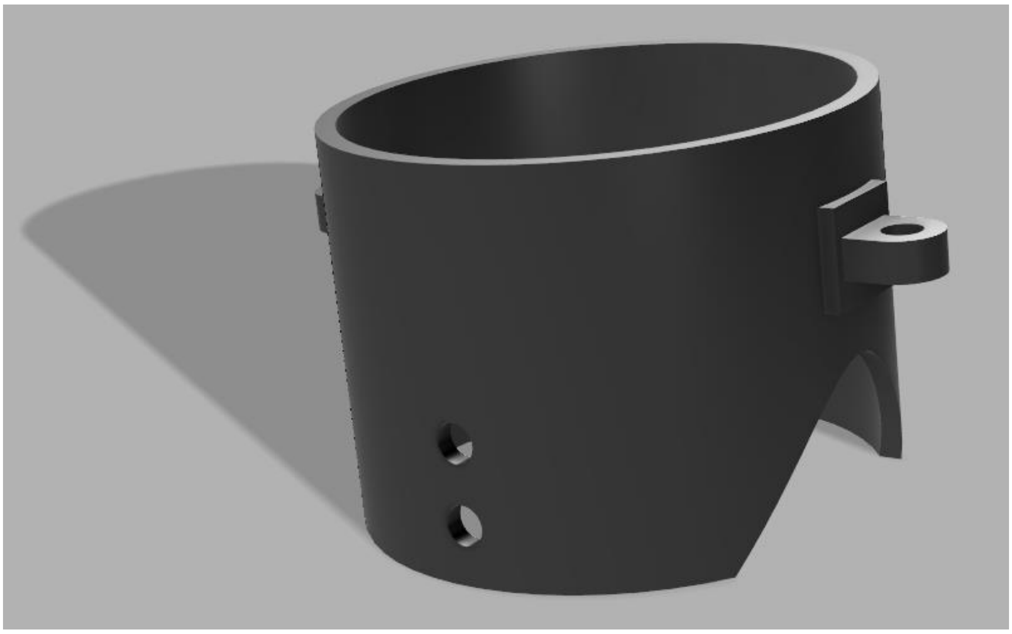


Figure 11: Final design for finger piece

I then designed two pieces that would be placed on the top [Figure 12] and underside [Figure 13] of the hand that would be used to gather the wires into a small area that could then manipulated into a cable that would then go from the glove to the Arduino. As my designed changed throughout the project I decided that I didn’t need the topside piece as I didn’t require Nitinol wire on both sides of the glove.

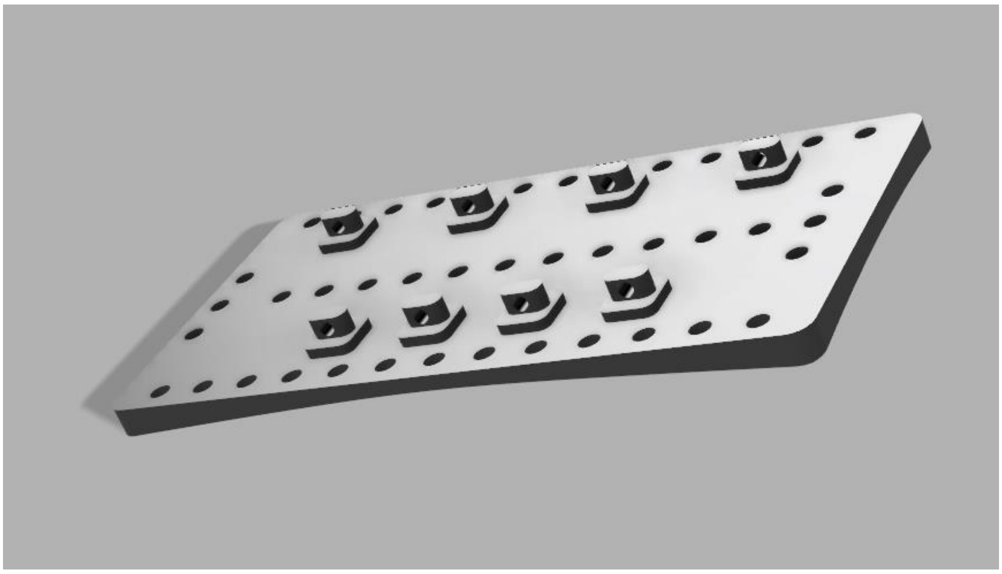


Figure 12: Topside hand piece

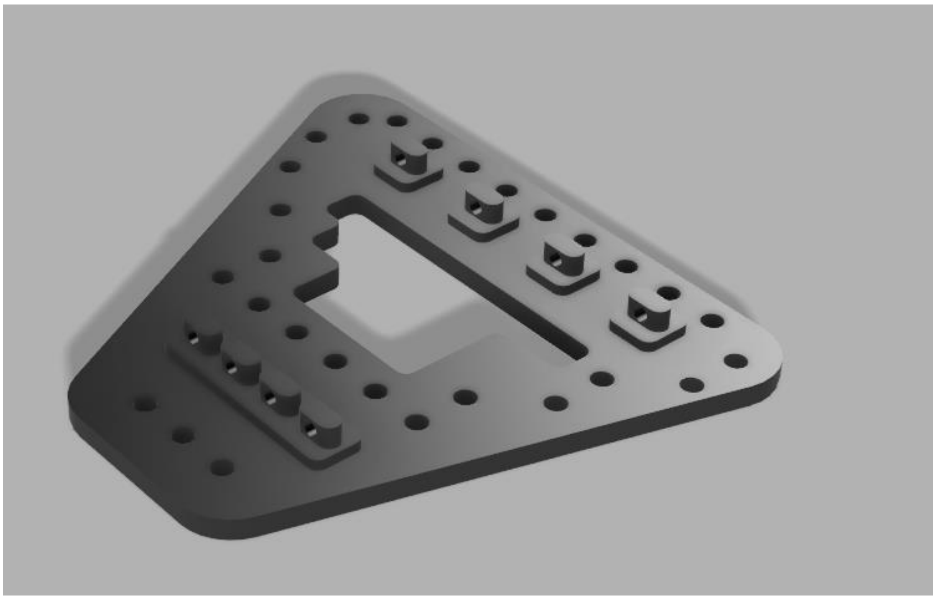


Figure 13: Underside hand piece

9.2 The Nitinol

For the Nitinol Wire I used a brand called Flexinol. I order five meters of Flexinol wire from robotshop.com. I chose the wire with the following properties: [Figure 14(Appendix)].

- High temperature
- Diameter: 150µm
- Activation temperature 90^{0}C
- Length 5m
- Resistance 55Ω/m
- Recommended operating current: 410mA
- Recommended pull force 321 grams
- Recommended Deformation 3-5%

The reason to why I chose this type of Flexinol was because it has an off time of 1.2 second [RobotShop (2019)], which made the wire relatively responsive whilst still providing a pull force large enough to be used to provide a Haptic Feedback response to the user. I chose to use 14cm of wire to which enabled me to have the wire running from the top of the fingers on the glove, to the middle of the palm of the hand. I found that this created a superior feedback response to the user than just having the Nitinol wire attached to both end of the finger. The Nitinol wire was threaded through the loops of the exoskeleton and attached to crimps on both sides, which were then connected to Molex's, with wires running back to the circuit. Originally, I wanted to attach the Nitinol Wire to both sides of the hand so then the user would have a force applied to them to both extend and retract their fingers. I found that as only a suggestive force was being applied to the user, a second force acting on the user to return their fingers to the neutral position wasn't required in this case.

9.3 Motors

For the motors I considered two different types, that being between the choice of Rumble motors and Coin-cell vibration motors.

The Rumble motors are the industry go to for providing Haptic Feedback in the current mainstream gaming industry with their use being seen in both the Xbox [Xbox (2019)] and PlayStation controllers [PlayStation (2019)]. They do provide a large force; however, they do also take over a second to wind up to their full speed to provide the force. This makes them not very responsive as a means of providing force feedback with another issue being their size, with each motor having the dimensions of 250mm x 252mm.

The coin cell vibration motors operate at 3V at 12000RPM, whilst being relativity small in comparison to the rumble motors, being only 10mm x 3.4 mm in size [Precision Microdrives (2016)]. The motors are also very responsive with generating their full force in under a second, however they do provide less of a force than the rumble motors.

I decided to use the coin cell vibration motors over the rumble motors as they have a slimmer profile which enabled me to attach them to the glove without making the glove clunky. If I had used the rumble motors, then I would have had to create housing for each one as the weight that spins around on the axel would need to be covered so then it wouldn't be in contact with the user or the environment around them. Whereas with the coin cell Vibration motor I was able to sew them into a pouch directly onto the glove.

9.4 Blender

I used blender to create the 3D models and the animations that would be used in the Unity simulation. My aim was to create a hand model that I would be able to animate so then when a user presses a key in Unity the model hand would move the same finger as the glove. At first, I wanted to create a realistic looking hand model, however I came across several issues with the main being due to a large amount of clipping when trying to attach the fingers to a wrist, which then caused the hand to be lag in the simulation due to the model being constructed using a large number of polygons. The process that I went through to create the model was to first take a picture of my own hand that I set to be the back ground of Blender to use as a reference. I then created an elongated octagonal prism that would be used as the base for the finger, I was then able to create the fingernail by using a series of extrudes [Figure 15]. With a finger completed I was able to duplicate the model, scaling and aligning the finger to match the reference image [Figure 16]. Once of all the fingers where I place I then able to extrude and merge the fingers together to create a palm, however I then came across a lot of errors when I tried to join the palm to a wrist. [Figure 17] During the process of created the palm there were now too many surfaces which caused areas of the model to clip, and due to the clipping and large number of surfaces now being used the model was causes a noticeable amount of lagging once rendered.

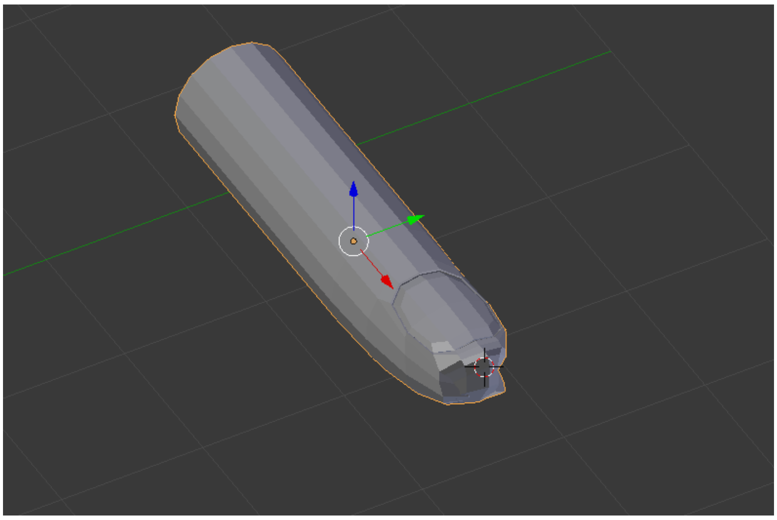


Figure 15: Finger model

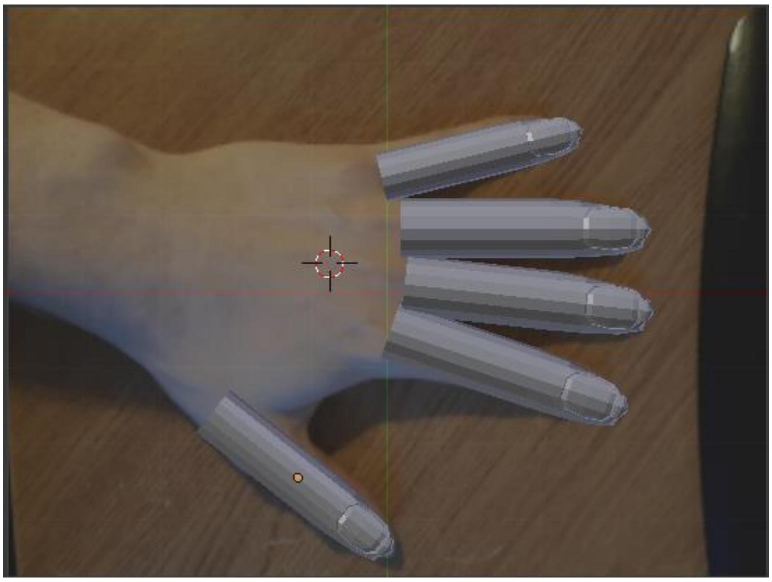


Figure 16: Hand with scaled fingers

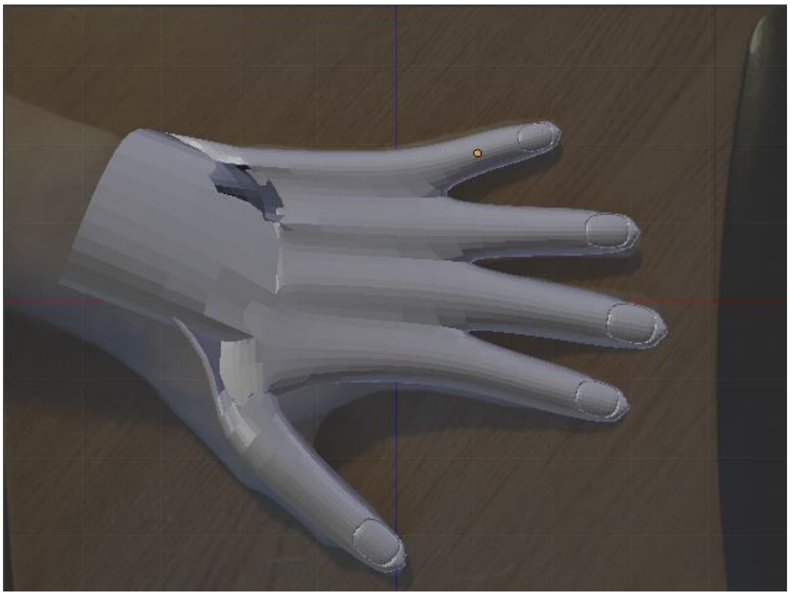


Figure 17: High poly hand with clipping

A solution to this problem was to opt for functionality over design, so I then created second version of the hand that took a minimalistic approach to the design aspect [Figure 18]. I created a new model using six cuboids in total with each finger and thumb being and the palm of the hand being made from a single cuboid. This then brought the number of faces from being 912 down to 27, which greatly improved performance. Now that I had a low poly model I then needed to add animations to it, before I could do this I needed to create a skeletal structure for the model which would allow the model to have the same movement as an ordinary hand [Figure 19]. With the skeletal structure in place I could then use a tool called key frames to create animations for each of the fingers and thumb moving. Keyframes work by capturing the models position at chosen frames and then calculates how fast the model needs to move to get to its next position in the given number of frames selected by the user. To do this I first captured the model at frame zero in the initial position I then set the animation to frame zero and moved the 3rd bone in the finger to Z position -0.2 and the 2nd bone to -0.7 also in the Z axi. Once both bones were in the current position I then captured the key frame. Then I went to frame 40 of the animation and set the model back to its initial position, and then captured the keyframe [Figure 20]. This then created a full animation of the finger being in the upright position to bending and then back to the upright position, which I replicated for the rest of the finger and thumb.

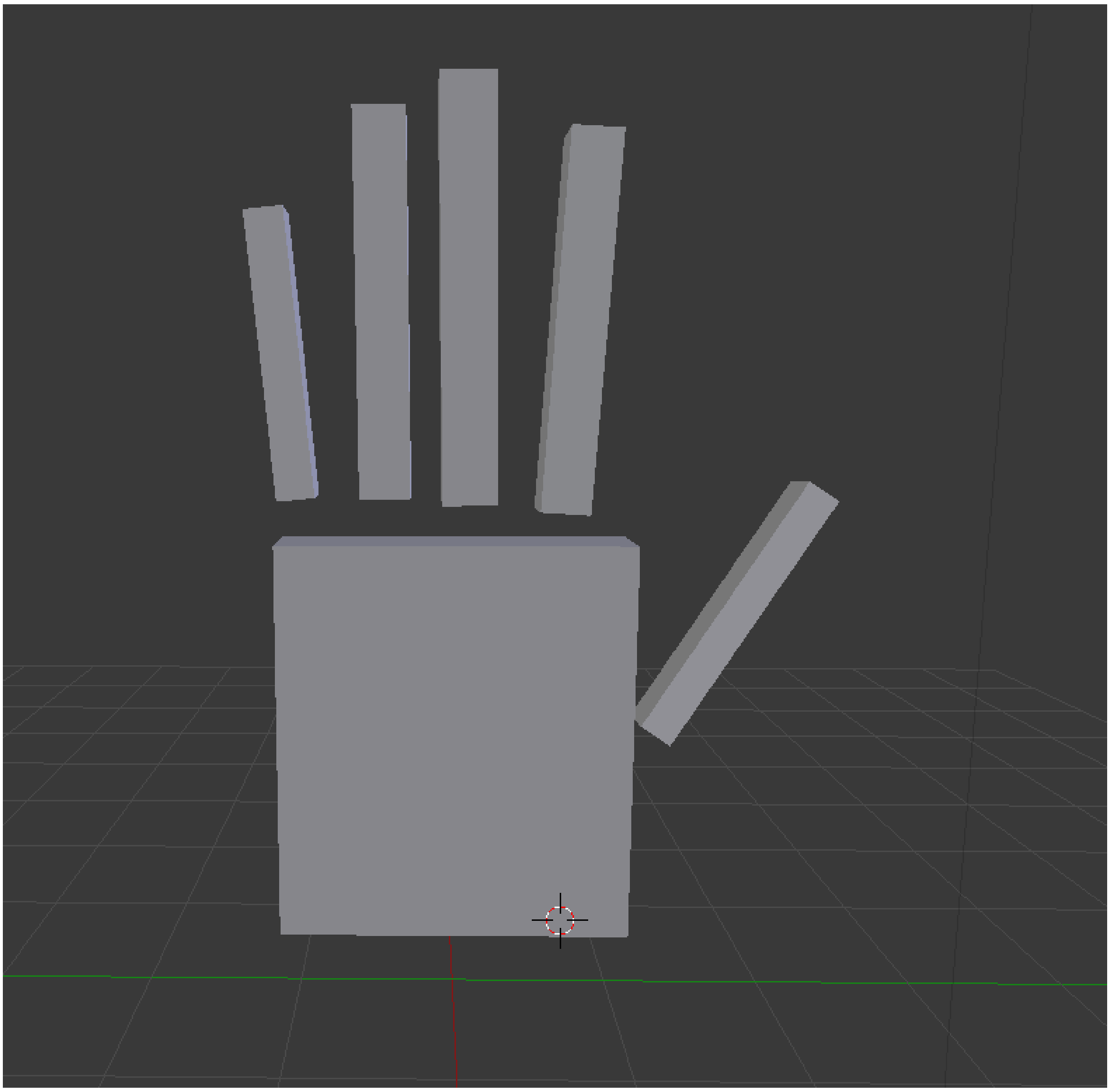


Figure 18: Hand model

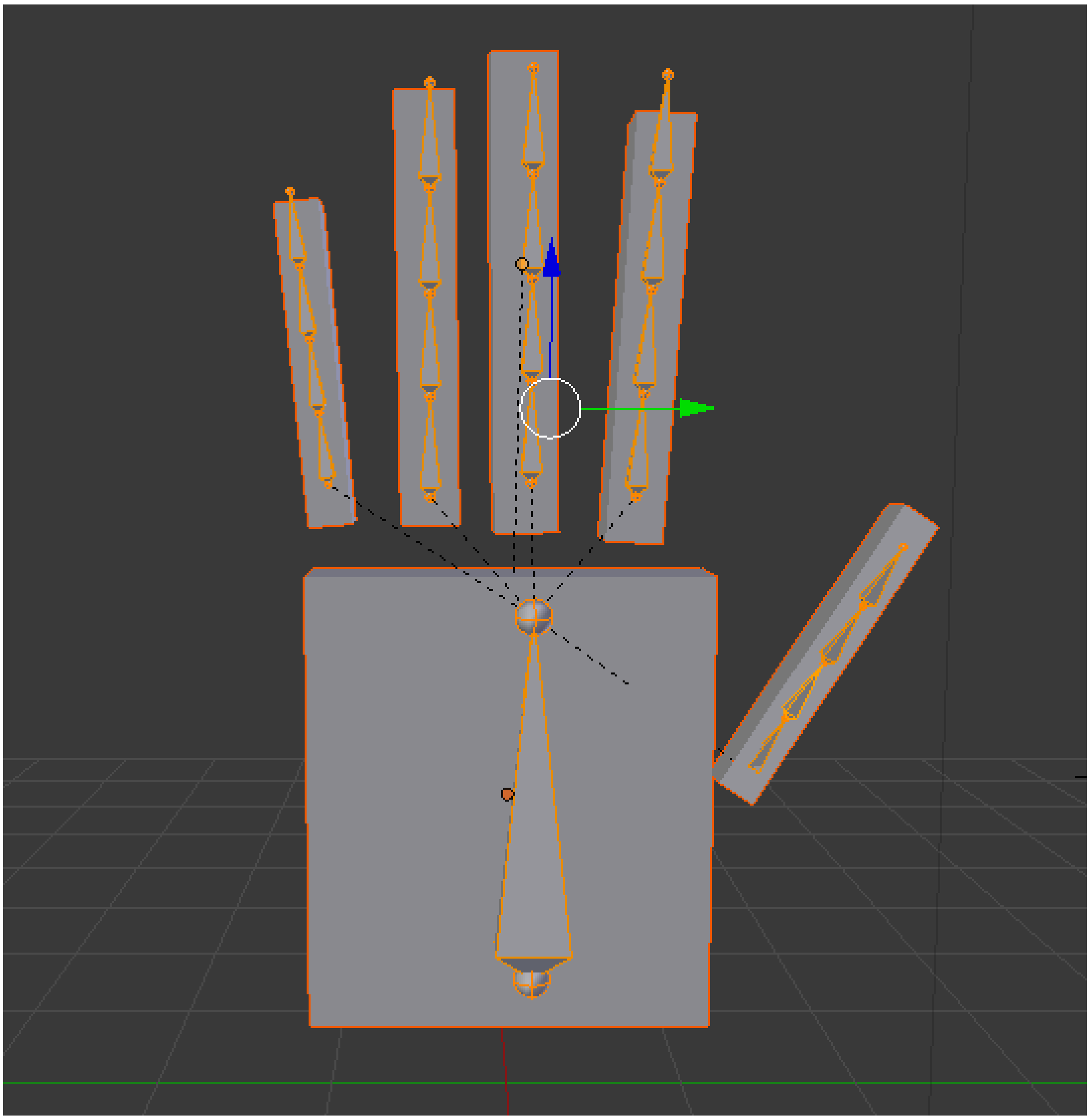


Figure 19: Hand model with Skeletal structure

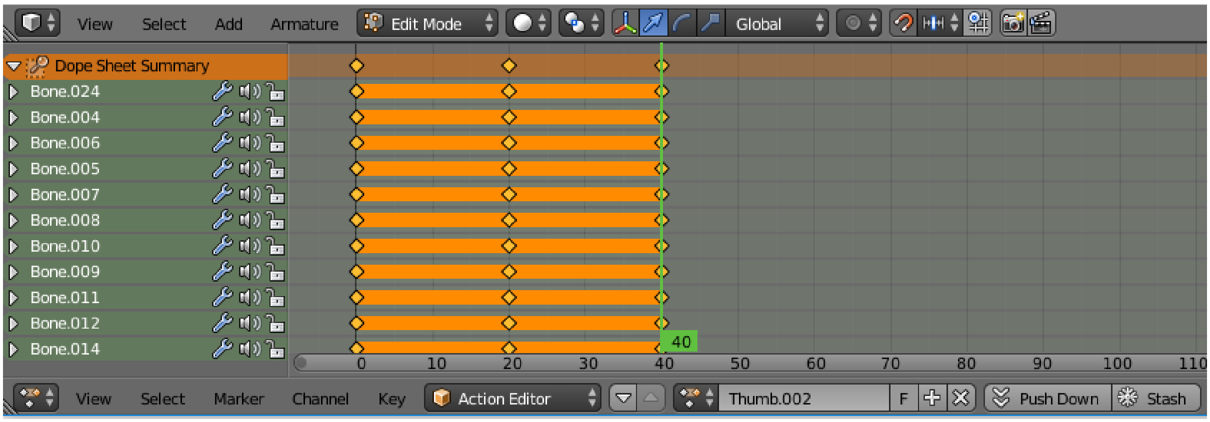


Figure 20: Key frames being captured at frame 0, 20 and 40

9.5 Unity

I used Unity with the model created in blender to create a simulation that would allow me to send signals to the Arduino and to trigger the animations from the Blender model. With the models being easily imported into Unity via Blender I had access to all the animations that I had created in Blender [Figure 21(Appendix)]. I then created an animation controller that allowed me to map the pathing for the animations and to select an idle animation that would play when no other animation was triggered [Figure 22].

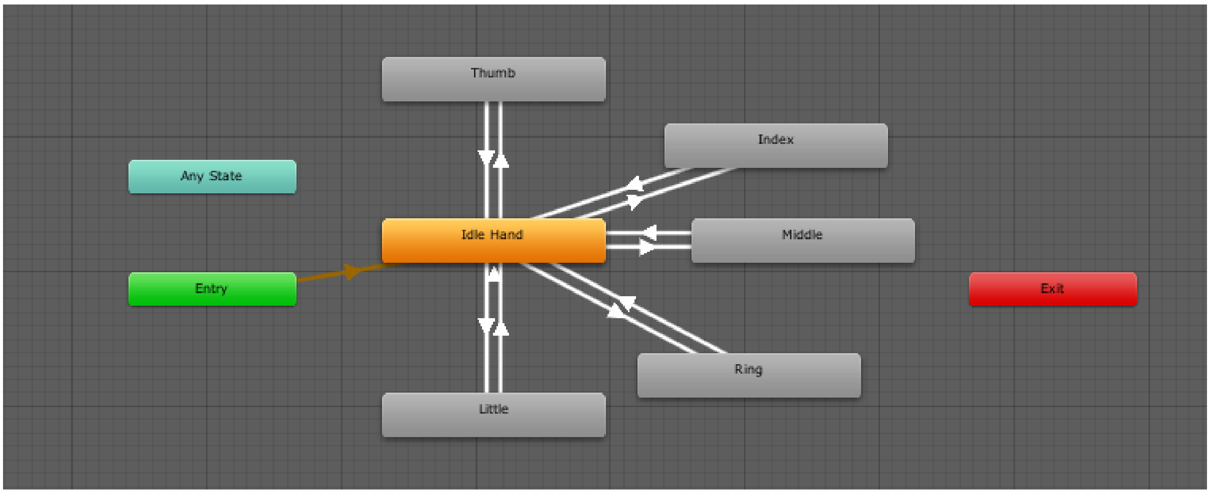


Figure 22: Map for animations

To control when an animation is played I wrote a script in C# called FingerAnimationScripts.cs that looks for a key press on the keyboard. In the example shown in figure 23, when the 'H' key is pressed it plays the Index finger animation and then sends a string to the Arduino containing the word "Index", which is then used to trigger an output pin that activates the Nitinol wire on the index finger of the Haptic glove.

```
if (Input.GetKeyDown(KeyCode.H)) // Press H to move Index Finger
{
    anim.Play("Index"); //Play Index Finger Animation
    Debug.Log("H key detected ");
    arduino.SendString("Index"); //Send string to Arduino Via serial Port
    Debug.Log("String sent ");
}
```

Figure 23: Code for control of animation and Haptic Glove

In order to send data to the Arduino I had to create another script called ArduinoComs.cs that would be used to open the serial port to the Arduino to send a string that could be read to decided which feedback was to be triggered. The first thing that this code does it to check to see what serial port is open [Figure 24]. It also sets the settings for the port to send data with a baud rate of 9600.

```
for (int k = 0; k < 50; k++) // Number of serial ports to be checked
{
    portNum = k; // Update Port number
    stream = new SerialPort("COM" + portNum, 9600); // COM port at 9600 baud rate

    if (!stream.IsOpen)
    {
        try
        {
            Debug.Log("Attempting to open serial port: " + "COM" + k); //Debug for open serial port
            stream.Open(); // Open serial port
            Debug.Log("Serial Port Open"); // Confirm port is open

            break;
        }
        catch (Exception ex)  // If serial port is not open send error message
        {
            Debug.Log("Failed to open port. Error: " + ex);
        }
    }
```

Figure 24: Code to check availability of serial port

Once the serial port that the Arduino was connected to has been identified the function SendStrings could then be called. This will then send the relevant string to the Arduino, dependent on what functional had been called in the FingerAnimationScripts.cs file [Figure 25].

```
public void SendString(string message) // Send message to Arduino
{
    try
    {
        stream.Open(); // Open stream
    }
    catch (Exception ex)
    {
        Debug.Log("Error: " + ex); // Error if steam not open
    }

    stream.Write(message);   //Send selected message to Arduino
    Debug.Log("Message was sent. Message was: " + message); // Erros message if string not sent
    stream.Close(); // Close stream
}
}
```

Figure 25: Code to send string

The final scripts that I implemented was two text scripts that I used to display the simulation controls to the user on the main scene [Figure 26 (Appendix)], (with the main scene being what is displayed on the screen). This then completed my simulation and was the final piece. With all the scripts implemented, the simulation had a hand model in an empty scene, when a mapped key was pressed [Table 1] the correct animation will play, and the relevant feedback will be impacted onto the user [Figure 27].

Controlls	
Key:	**Action:**
G	Moves Thumb (Nitinol)
H	Moves Index Finger (Nitinol)
J	Moves Middle Finger (Nitinol)
K	Moves Ring Finger (Nitinol)
L	Moves Little Finger (Nitinol)
Y	Thumb Motor
U	Index Finger Motor
I	Middle Finger Motor
O	Ring Finger Motor
P	Little Finger Motor
F	Stop Action

Table 1: Controls for simulation

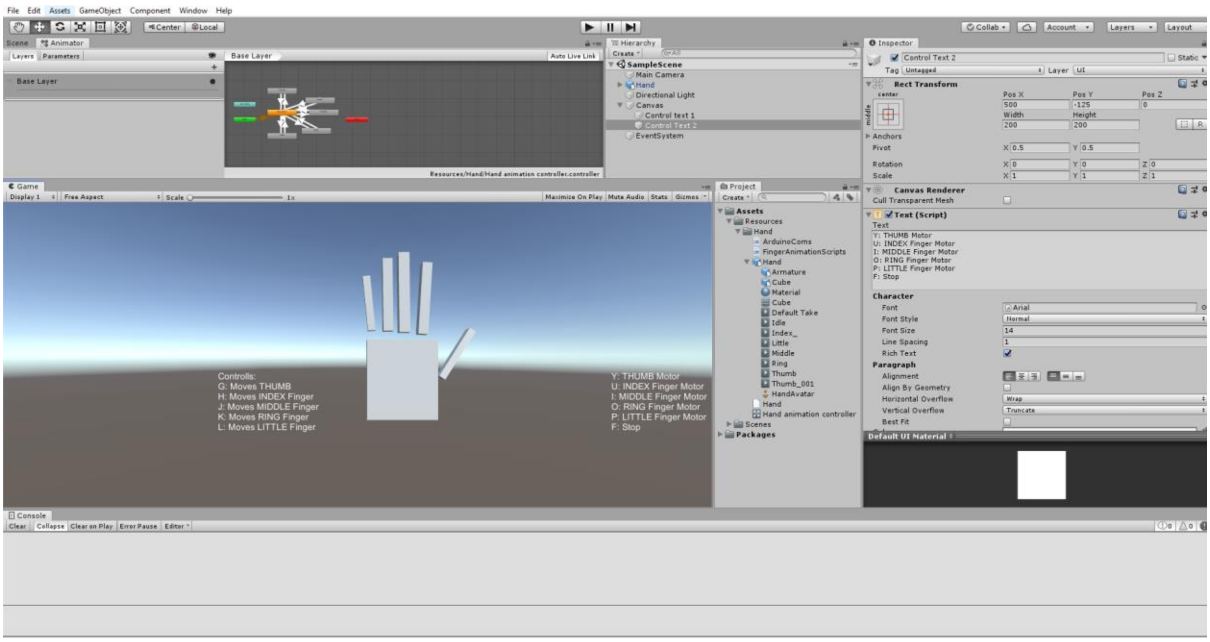


Figure 27: Unity simulation with all components

The last step was then to create a build of the simulation which means that it can be transferred to another computer without needed all the assets and software to run it [Figure 28].

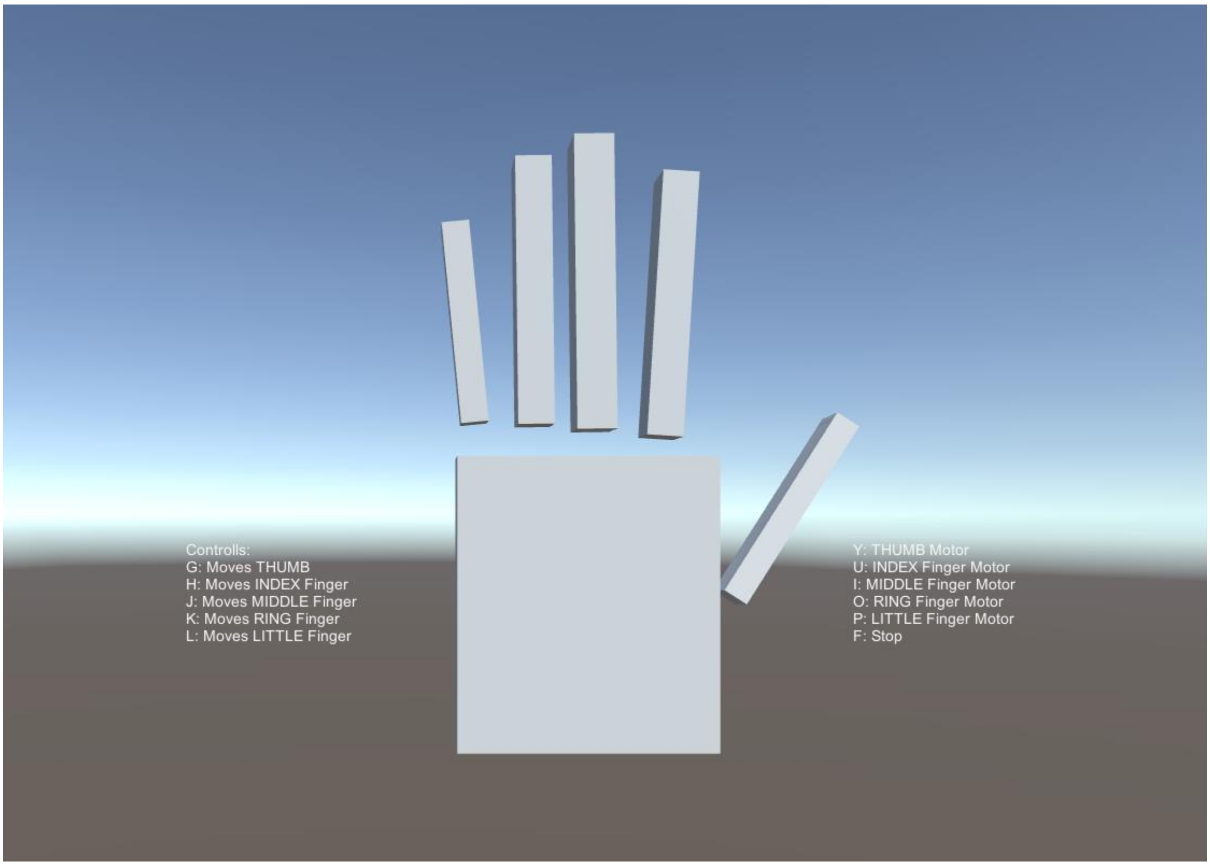


Figure 28: Final Build of simulation

9.5 Arduino

The Arduino was used to send signals to each one of the Haptic Feedback modules. Each device was connected to one of the output pins as seen in Table 2. The code worked by having the Arduino reading data from the serial port at a baud rate of 9600. When a message was received from the serial port the code then looks at a series of if statements which if the message is matching the condition for the if statement it then fires the relevant feedback [Figure 29].

Arduino Pin Map	
Device	**Pin Number**
Little Finger Nitinol	Pin 13
Ring Finger Nitinol	Pin 12
Middle Finger Nitinol	Pin 11
Index Finger Nitinol	Pin 10
Thumb Motor	Pin 7
Index Finger Motor	Pin 6
Middle Finger Motor	Pin 5
Ring Finger Motor	Pin 4
Little Finger Motor	Pin 3

Table 2: Arduino Pin mapping

```
if(content.substring(0,5) == "Index") {
String Hand =  content.substring(5, (content.length()));


  digitalWrite(10, HIGH); // Index Finger ON
  delay(3000); //Wait 3s
  digitalWrite(10, LOW); // Index Finger OFF

  }
```

Figure 29: Arduino Code reading data from serial port and Activating the Nitinol wire for the Index finger

9.6 Circuitry

To able to cause the Nitinol wire to transform in between its Martensite and Austenite state, I needed to heat the wire using ~400mA. In order to provide a relatively high current without bring the costs of the project up, I've used a LM317 Voltage regulator to create a current limiting circuit that will keep the current at 400mA for a given resistance. In the case of my final design the resistance of the wire was 7.7Ω. which required a 3 Ω for R2 [Equation 2] to keep the current limited to 400mA. In order to turn the Nitinol wire on and off I used a ZTX651 transistor that allowed me to pulse a signal from the Arduino output pin to the transistor which then activated the circuit, causing the Nitinol to change state [Figure 30].

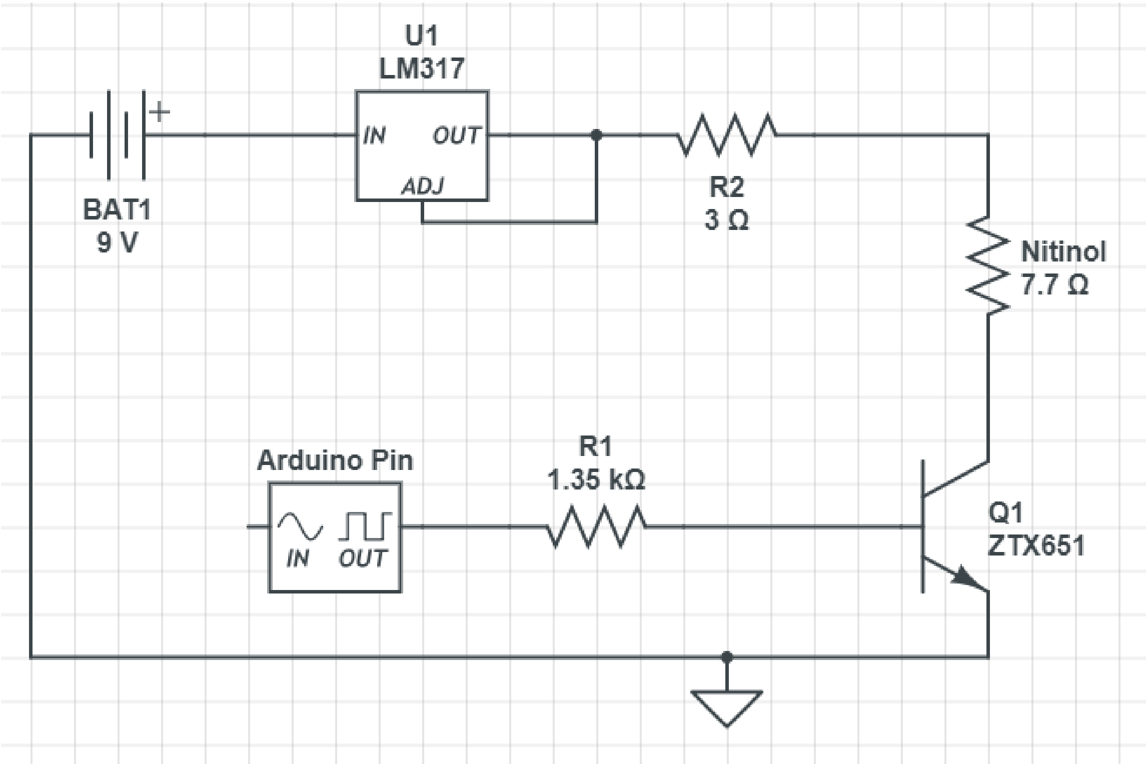


Figure 30: Circuit diagram for Current Regulator and switch

Originally, I wanted to use the 5V supply from the Arduino, however through testing I found that due to the back voltage of the LM317, which was 3v. This then limited the resistance that of the Nitinol to around 4.4Ω which then restricted the length of the wire to ~8cm. To improve this, I decided to use a 9V battery instead of the 5V supply from the Arduino. This then gave me more leeway with the length of the Nitinol wire and I was able to use the 14cm of wire that I required, which had a resistance of 7.7Ω, without the current dropping off early.

Once the circuit was working correctly I transferred the circuit to a piece of Veroboard. The circuit was comprised of five LM317 circuits and five transistor switch circuits, which is one circuit for each finger. I decided not to have the thumb provide a force feedback as the angle was difficult to incorporate into my design and it also provided the user with something to hold onto when taking off the glove, without adding tension to the Nitinol wires. I soldered two 4 pin Molex connectors onto the board that allowed me to connect the Nitinol wire between the LM317 and transistor circuit. Each strand of Nitinol was connected to the glove via crimps, as Nitinol is not solderable without an iodine flux. The wires on each end of the Nitinol are colour [Table 3]. Each one of the coin cell vibration motors were also soldered to the Veroboard, with them being directly connected to the Arduino, as due to their low operating Current and Voltage they could all be driven by the Arduino's PWM output pins. Lastly, I soldered the connectors for the battery to the board as well as an extra wire for the ground and for the supply voltage, in case I needed to power the circuit using a power supply for testing purposes. With two smaller pieces of Veroboard I soldered a single inline pin for each one of the transistor input signals and motors, which allowed me to plug the wires into the Arduino securely. [Figure 31].

Wire colour coding			
Type	**Colour**	**Pin**	**Location**
Nitinol	Grey	Pin 13	Little Finger
Nitinol	Organge	Pin 12	Ring Finger
Nitinol	Yellow	Pin 11	Middle Finger
Nitinol	Purple	Pin 10	Index Finger
Motor	white	Pin 7	Thumb
Motor	white	Pin 6	Index Finger
Motor	white	Pin 5	Middle Finger
Motor	white	Pin 4	Ring Finger
Motor	white	Pin 3	Little Finger

Table 3: Wire colour coding table

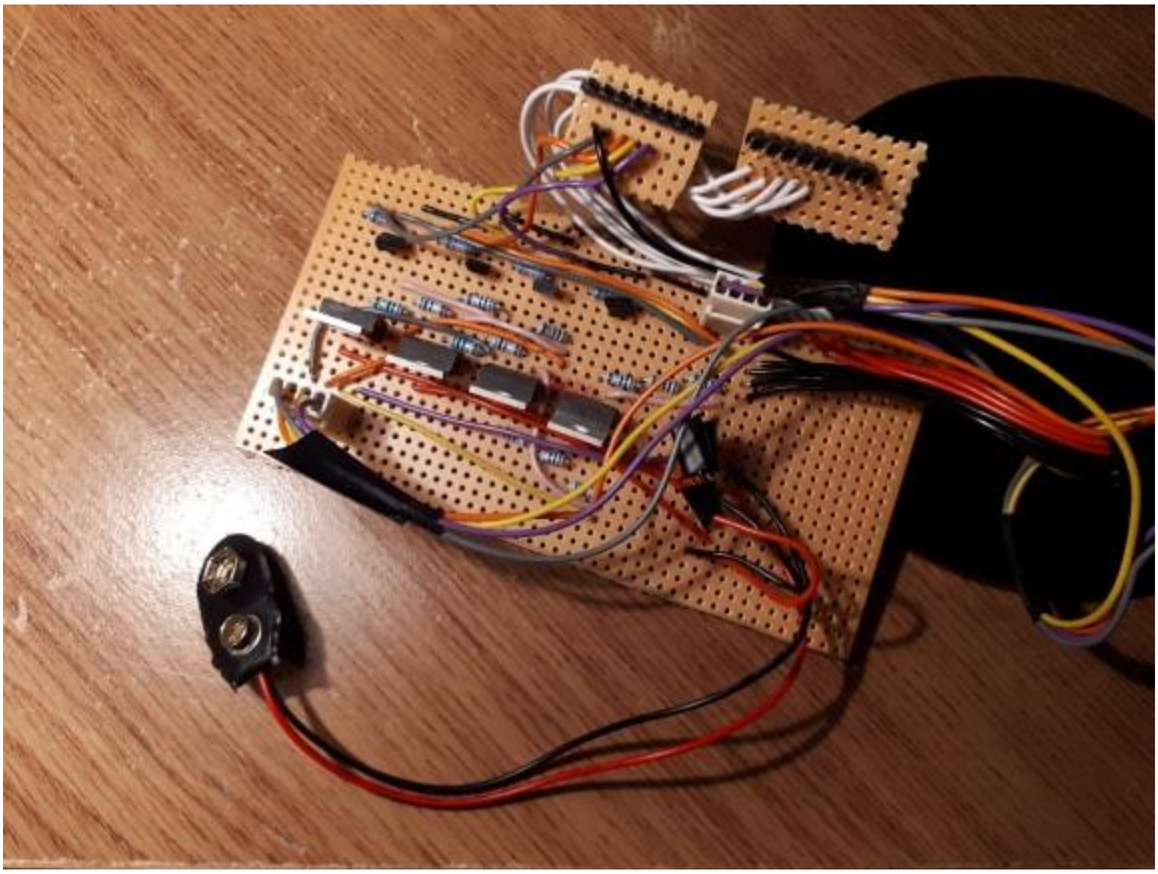


Figure 31: Veroboard Circuit

10 Testing

Throughout the process of this project each component was tested to ensure that everything was working as should be expected and when not working correctly, the design was adapted to meet the requirements.

10.1 Software

In order to test the software, I wrote a series of Debug commands into the code, so then if an error would appear such as the serial port not being open, or a string hadn’t been sent to the Arduino. To test that I could successfully send a string from Unity to the Arduino I wrote some test code that simply toggled an LED when a key was pressed. This gave me a visual representation that it was successfully sending a signal. Once I was able to signal for an LED to be turned on and off I could then apply similar code to signal for the motors and Nitinol to be toggled.

10.2 Hardware

After I had finished soldering all the components onto the Veroboard I then checked all the connections with a multimeter. I then made sure that all the points on the Veroboard that I had drilled through to break the connections between two components no longer conducted a current.

After I had assembled the glove with all the Nitinol wire and motors working, I then had six people of different builds try on the glove to see if they were able to fit their hand into the device comfortably. I found that the glove fit all of them, however I received feedback that the Ring finger exoskeleton was tighter than the rest. To solve this issue, I loosened the stitching on the 3D printed components. Whilst the glove did fit everyone, it did take a while for each user to remove the glove due to needed to pull the glove of in increments due to the Nitinol wire attached to the inside of the fingers. With the project showcase in mind, I then decided to create a second glove with Nitinol on only one of the fingers which allowed me to demo the force applied to the user but making it easier for the user to take the glove off with a time constraint kept in mind.

Another test was to make sure that the wires had been all connected correctly, to test this I loaded the build of the Unity simulation and pressed all the mapped keys to double check that the correct feedback was being applied to the user in relation to its key press.

From my testing I found that the loops on the 3D printed palm piece were too thin and thus the wire cut through the loops. To solve this issue, I changed the original model on Auto Desk Fusion 360 and created thicker walls for the loops. This was then implemented into my final design and solved the issue [Figure 32].

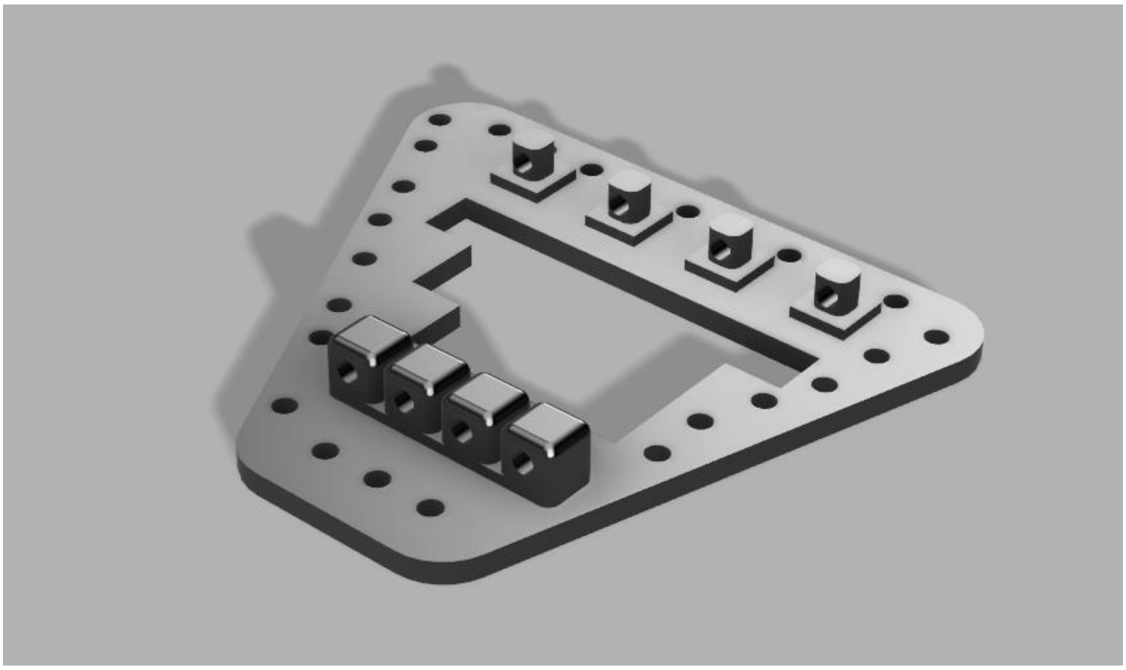


Figure 32: Larger loops on the palm piece

10.2.1 Nitinol Characterising

Nitinol was a material that I hadn't previously worked with before and so I carried out a series of tests to see what the wire was capable of. My first test was to determine at what current the wire would work at. I used a 10.5cm length of wire and connected it via crocodile clips to a power supply. I then started off a 200mA and incremented in intervals of 10mA. I found that the wire wouldn't do anything until 230mA had been applied and even then, you could only see a slight tension in the wire. At 240mA the wire then contracted by 2mm and then as the current increase the length of movement increased [Table 6].

Minimum Current testing				
Current (mA)	Original length	Final Length (mm)	Change in length (mm)	Notes
200	105	105	-	No Change
210	105	105	-	No Change
230	105	105	-	Tension in wire
240	105	103	2	First movement
250	105	102	3	
260	105	101	4	
270	105	100	5	
280	105	100	5	

Table 6: Minimum Current to activate the Nitinol

After this initial testing I decided that it was a poor way of characterising the Nitinol wire. Rather than just having the wire on the desk and measuring the change in length in comparison to a change in current. I decided to measure the wire whilst it had a weight attached to it. I attached the Nitinol wire to a clamp in front of a ruler and used a set of penny weights to increment the weight on the wire to perform stress testing. This time I had used a fixed current of 400mA and a length of Nitinol of 118mm and the starting weight at 50g. I then increased the weight in increments of 50g up until 400g where the wire snapped and calculated the stress and strain of the wire. [Equation 5 & 6].

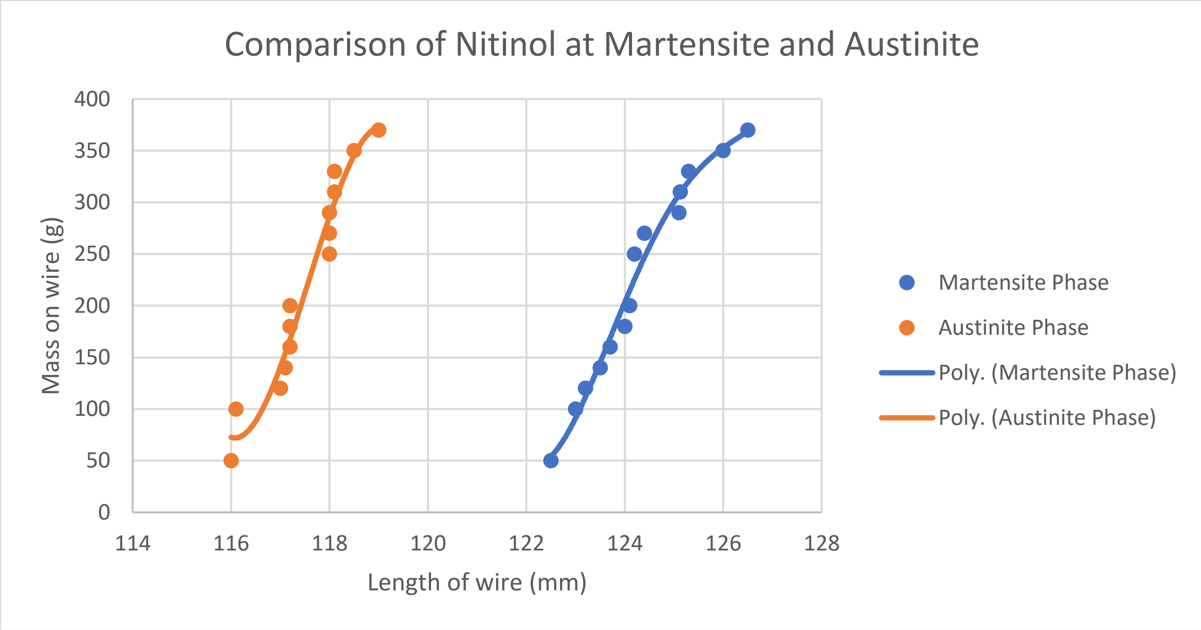
Nitinol under increasing weight					
Mass (g)	Start length (mm)	Finish Length (mm)	Change in length (mm)	Stress (Pa)	Strain
50	118	113	5	27711864	0.0424
100	117	113	4	55423728	0.0513
150	118	112	6	83135593	0.0508
200	118	113	5	110847457	0.0423
250	118	111	7	138559322	0.0593
300	117	110	7	166271186	0.0598
350	118	109	11	193983050	0.094
400	Break	Break	Break	Break	Break

Table 7: Nitinol Under increasing stress until breaking point

I then carried out a third round of testing where I measured the length extension of the Nitinol wire with increasing mass for both the Martensite and Austinite phases. This then demonstrated that the Nitinol wire is stronger when in its Austinite phase as the extension of the wire as the wire extended less comparison to the Nitinol in its Martensite phase [Table 8] [Graph 1].

Comparison of Nitinol at Martensite and Austinite					
Room temp			~90 temp		
Mass (g)	Length of wire (mm)	Extension (mm)	Mass (g)	Length of wire (mm)	Extension (mm)
50	122.5	0	50	116	0
100	123	0.5	100	116.1	0.1
120	123.2	0.7	120	117	0.9
140	123.5	0.5	140	117.1	0.1
160	123.7	0.5	160	117.2	0.1
180	124	0.5	180	117.2	0
200	124.1	0.4	200	117.2	0
250	124.2	0.2	250	118	0.8
270	124.4	0.3	270	118	0
290	125.1	0.9	290	118	0
310	125.13	0.73	310	118.1	0.1
330	125.3	0.2	330	118.1	0
350	126	0.87	350	118.5	0.4
370	126.5	1.2	370	119	0.5

Table 8: Comparison of the length of Nitinol wire in both the phases which with increasing mass



Graph 1: Relationship between Mass and Length of wire is both phases

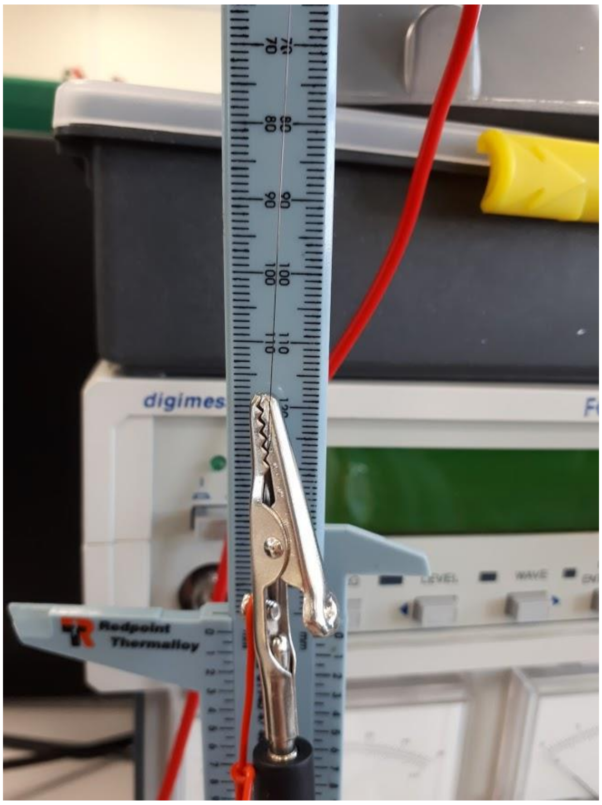


Figure 33: Testing Nitinol

10.2.2 Circuit Calibration

Before I soldered my circuit to the Veroboard I first tested the components to make sure that each one was working as expected and also so then each module of the circuit was calibrated correctly to provide the correct outputs. The first set of testing that took place was to make sure that the current supply circuit was providing the Nitinol wire with 400mA. To do this I created a series of circuits comprised of 20W resistors with the resistance of 22Ω in parallel [Figure 34] which gave me a range of resistances between 11 Ω and 3.14 Ω. The reason why I used 20W resistors is because of the high power over the load, which for 7.7 Ω would be 1.232W [Equation 3]

$$Power \ = R \ \times I^2$$

Equation 3: Power for R = 7.7Ω and I = 400mA

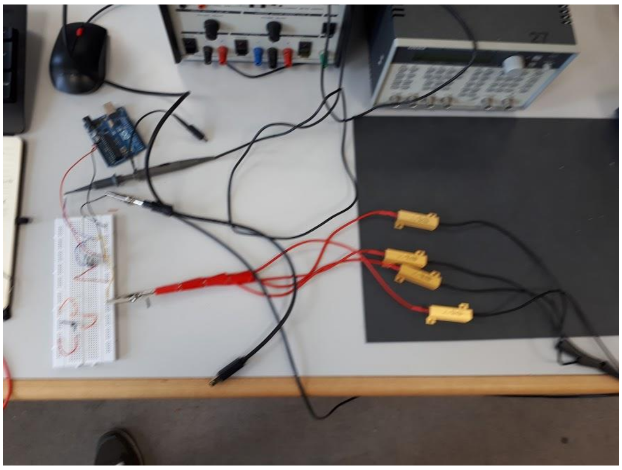


Figure 34: Parallel resistor circuit to test current limit

In total I created six different circuits by placing the resistors that were available to me in parallel and calculated the current over the load [Equation 1]. From my initial testing I found that the current was dropping below 400 mA when the resistance was greater than 4.4 Ω [Table 4]. Whilst the Nitinol would still function at a lower current, I wanted to be able to use a wire that was 14cm long which would have a resistance of 7.7 Ω and would therefore need to have higher current in order to keep the operational speed fast.

$$I = \frac{V}{R}$$

Equation 1: Derived from Ohms Law to find current

Current Limiting ciruit With 5V supply		
Resistance Ω	Volage across Load (V)	Current Across Load (mA)
11	2.5	227
7.3	2.288	313
5.5	2.016	366
4.4	1.798	408
3.6	1.541	428
3.1	1.32	425

Table 4: Relationship of resistance and current over Load (Nitinol wire), 5V supply

After probing the circuit with a multimetre I found that not enough Volts were able to make it the Load when using the 5V supply from the Arduino. In an attempt to solve this issue, I decided to redo the experiment with a 9V battery instead of using the Arduino as the power supply. I found that the drop off resistance was increase to greater than 5.5 Ω [Table 5]. Whilst the data does show only a small improvement to the current being supplied for each resistance, there was a noticeable difference and I think that my data provided for the 9V supply may have errors as the data collection was rushed. If time had allowed I would have liked to have redone this round of testing, but as the Nitinol wire was being activated sufficiently fast I continued to solder the final circuits onto the Veroboard.

Current Limiting ciruit With 9V supply		
Resistance Ω	Volage across Load (V)	Current Across Load (mA)
11	2.9	263
7.3	2.5	342
5.5	2.3	418
4.4	1.833	417
3.6	1.345	376
3.1	1.137	355

Table 5: Relationship of resistance and current over Load (Nitinol wire) at 9V supply

Inconclusion to my testing of Nitinol I was able to characterise the wire. For 14cm of Nitinol wire at 230mA with no mass acting on the wire, tension was first seen in the wire. At 280mA the wire first starts to contract but only by 2mm. At 280mA the wire has an increased contraction of 5mm. I found that the Nitinol wire works best when the current over the wire is between 380mA and 420mA, so I selected 400mA to be my fixed current as that then provided a 20mA leeway. The Nitinol wire can operate with a weight attached up to 400g where at that point the wire becomes damaged and can potentially break. As long as the deformation of the wire is kept between 3-5% then the wire can operate over a long period of time, where as if the deformation goes above 3-5% then the wire becomes permanently deformed and weakened.

11 Results and Discussion

I found that the 3D printed finger pieces worked well, with them being the correct size to fit on most people's hand comfortably. However, I would make a few changes them. The finger pieces would work better if they had holes to sew them onto the glove on both the top and bottom of the main body instead of just on the bottom. This would have to be tested as although the it would help to keep tension in the wire, it may also make the glove more restricted for the user and could mean that users with larger hands may have difficulties using the glove. The printed component that was attached to the centre of the palm functioned as I wanted it to, however the 3D print did fail and one of the loops broke off due to a miss print. I think that if the same piece was to be printed again then there wouldn't be an issue, however under a time constraint I wasn't able to print another part in time to fully test this.

The implementation of the Haptic Feedback also went well, the users was able to feel the suggestive force on their fingers and the vibration from the motors on the back of their hand. To improve this, I would like to implement a multicore strand of Nitinol wire that would allow me to apply a larger feedback force to the user. With a stronger force being applied I would then create a way of providing a force in the opposite direction, this could be in the force of another Nitinol wire, or the use of elastic to create a resistance to the user's finger movements. In addition, I would also use more have more motors on the glove to provide specific force to the hand/fingers. For example, one motor per degree of freedom.

The Unity simulation and Arduino code achieve full functionality; however, both could be improved. The Unity simulation does successfully find what COM port the Arduino is on and sends the correct string, however the Arduino is only reading the sting that has been sent via the serial port and then comparing the string received to the criteria to enter a series of if loops. I would have liked to have used threads in my code to have improved the functionally of the Arduino code as in its current state the Arduino can only apply one type of feedback at a time. Instead of using an Arduino Uno I would use a STM Microelectronics board, with Mbed libraries, which would have enabled me to write more complex code. To improve the simulation, I could have implemented a more complex and realist hand, and I could create an interaction with the model that would trigger the Haptic Feedback instead of the user having control over the feedback directly.

The first thing that I would do to improve the circuit would be to design a PCB instead of having my hardware soldered onto a piece of Veroboard. I would also find a better way to provide the current required to activate the Nitinol wire. Originally, I was just going to have the wire use the lab power supplies, however I wanted to have the Whole Haptic device portable, which is why I then tried to find a way to produce 400mA to supply the Nitinol with, whilst only using the Arduino and later a 9V battery. In hindsight I should have stuck to using the power supply as by using the currently limiting circuit I became very restricted on the length of Nitinol wire that I could use for each finger. I would also like to find a better way of connecting the Nitinol to the rest of the circuit, where my method of using crimps and Molex's to connect the Nitinol wire did work, however it wasn’t a very secure way to connect the wire, and the Molex boxes had to be glued in place to keep the tension in the wire. A clamp system may have worked better, where the wire could be placed in between two small plates of a conductible material that could have the power/ground wires soldered onto. I would also like to extend the length of the wire going from the circuit to the Glove [Figure 35]. Originally, I wanted to have the circuit mounted to the wrist of the user, however with seeing how bulky the box needs to be to house the circuitry I think it would work a lot better if you could have the box mounted to a belt. This would also allow the user to use the device in a VR environment without restricting their movement.

Figure 35: Finished Haptic Feedback Glove

12 Conclusion

In conclusion believe that I’ve achieve the objectives that I’ve set out to complete. All of them can be improved in various different ways, but a project never truly comes to an end. I’ve created a simulation that can output data to an Arduino to control both the motor and Nitinol feedback, I’ve designed and implemented a glove that the user can wear to experience the Haptic Feedback and all of the individual systems work together. Overall the Haptic Feedback glove works and does provide the user with two types of Haptic Feedback, with the Nitinol providing a suggestive contraction force on the fingers and the motors providing a vibration force.

13 Future Work

Whilst my main inspiration has come from the gaming industry, along the journey of the project I have thought have how the device/technology could be applied to other situations. One of the ways that the device could be used is in rehabilitation of people who have lost the ability to use their fingers after having a stroke. Stroke patients who experience a motor defect in relation to their fingers have to carry out hours of repetition tasks in order to retrain the muscle memory in their fingers to regain functionality. Several studies have taken place with use of robotic assisted rehabilitation being used to help people to retrain their hands. One trial saw the use of a device called the MusicGlove [(Nizan FriedmanEmail author, Vicky Chan, Andrea N Reinkensmeyer, Ariel Beroukhim, Gregory J Zambrano, Mark Bachman and David J Reinkensmeyer). (2014)] is a controller that used to get the user to apply pressure on both sides of a device to play a guitar hero style game. In that study they found that having the participants using their device, their recovery was improved. Another study that I’ve read implements an exoskeleton style support that help the participant to retrain their fingers by assisting their movement. They also had the user controlling a game/simulation whilst wearing the device. This study had mixed results, the participants had an increase in motivation to carry out the activities, but only the participants that had a server defect saw a significant improvement by taking part in the study opposed to standard rehabilitation [(Justin B. Rowe, Vicky Chan, Morgan L. Ingemanson, Steven C. Cramer, Eric T. Wolbrecht, David J. Reinkensmeyer) (2017)]. I think that with some adaptation to my Haptic glove, the Nitinol exoskeleton could be used in the same way to assist the user’s movement in finger related exercise and activities.

After learning a lot about Nitinol through my research and implementation of the material into my own project, I would like to investigate the possibility of using Nitinol to create a prosthetic hand. The wire is very light weight and with a single strand can lift over 300g, with four wires in parallel you could lift over a 1kg. In the case of using Nitinol as an actuator in a robotic hand, it would require some form of Nitinol mesh to be implemented that would give the user the ability to move the fingers to a variable position and to allow 3 axes of movement. Nitinol could be a light weight solution to using traditional servo motors that are currently used in prosthetic hands and would reduce the production cost as for purchasing 5m of Nitinol cost me £25.98 retail, where as a single servo motor could cost hundreds of pounds. The University of Saarland are currently developing a Robotic hand using shape memory alloys to mimic muscle fibres in fingers [University Saarland (2015)].

In conclusion I think that the implementation of Nitinol into both of these applications would be very interesting to research and further the develop the available technology that’s available. Overall, I have enjoyed my work with Haptic Feedback, but I think going forwards I would like to develop a Robotic hand that uses shape memory alloys as an actuator instead of motors as I believe that it will help to bring the cost and weight of prosthetic hand down, which would especially benefit children who require a prosthetic.

14 References

BIOFLIGHTVR (2019) *Website with company details.* Available at: https://www.bioflightvr.com/ Accessed: [1 May 2019].

Chemistry Learner (2019) *Nitinol* Available at: https://www.chemistrylearner.com/nitinol.html Accessed: [19 May 2019].

Confluent Medical (2019) *Nitinol Facts.* Available at: https://confluentmedical.com/tech-center/nitinol-facts/ Accessed [19 May 2019].

(Evan Strasnick, Christian Holz, Eyal Ofek, Mike Sinclair, Hrvoje Benko). (2018*)* *'Haptic Links: Bimanual Haptics for Virtual Reality Using Variable Stiffness Actuation'* CHI 2018, April 21–26, 2018, Montreal, QC, Canada. Available at: https://www.microsoft.com/en-us/research/uploads/prod/2018/02/HapticLinks_Strasnick_et_al_CHI2018-5a9063428e701.pdf Accessed: [1 May 2019].

Farnell Arduino (2019) *Arduino Uno* Available at: https://www.farnell.com/datasheets/1682209.pdf Accessed [19 May 2019].

Farnell PLA (2017) *Technical data sheet PLA* Available at: http://www.farnell.com/datasheets/2310522.pdf Accessed [19 May 2019].

Haptx (2019) *HaptX Gloves Development Kit* Available at: https://haptx.com/ Accessed: [1 May 2019].

(H. Fujita). (1989) *Studies of micro actuators in Japan.* Proceedings, 1989 International Conference on Robotics and Automation DOI: 10.1109/ROBOT.1989.100200

(Jaronie Mohd Jani, Martin Leary, Aleksander Subic, Mark A. Gibson). (2013) *A review of shape memory alloy research, applications and opportunities* Materials & Design (1980-2015) Volume 56, April 2014, Pages 1078-1113 Available at: https://doi.org/10.1016/j.matdes.2013.11.084 Access [19 Many 2019].

(Justin B. Rowe, Vicky Chan, Morgan L. Ingemanson, Steven C. Cramer, Eric T. Wolbrecht, David J. Reinkensmeyer) (2017) *Robotic Assistance for Training Finger Movement Using a Hebbian Model: A Randomized Controlled Trial* Availble at: https://doi.org/10.1177/1545968317721975 Accessed [19 May 2019].

New ATLAS (2017) *Haptic gloves bring long-distance couples a touch closer*. Available at: https://newatlas.com/flex-n-feel-glove-long-distance-relationships/47900/ [Accessed: 1 May 2019].

Nintendolife (2013) Taking a look back at the nintendo 64 rumblepak. Available at: http://www.nintendolife.com/news/2013/07/feature_taking_a_look_back_at_the_nintendo_64_rumble_pak [Accessed: 1 May 2019].

(Nizan FriedmanEmail author, Vicky Chan, Andrea N Reinkensmeyer, Ariel Beroukhim, Gregory J Zambrano, Mark Bachman and David J Reinkensmeyer). (2014) *Retraining and assessing hand movement after stroke using the MusicGlove: comparison with conventional hand therapy and isometric grip training* Journal of NeuroEngineering and Rehabilitation201411:76 Available online: https://doi.org/10.1186/1743-0003-11-76

PlayStation (2019) DualShock 4 wireless controller Available at: https://www.playstation.com/en-gb/explore/accessories/dualshock-4-wireless-controller/ Accessed:[1 May 2019].

Precision Microdrives (2016) *Product Data Sheet 10mm Vibration Motor*. Available at: https://www.precisionmicrodrives.com/wp-content/uploads/2016/04/310-101-datasheet.pdf Accessed: [19 May 2019].

PROVR (2019) *Website with summary*. Available at: https://provr.io/professional-training-vr/ Accessed: [1 May 2019].

Push Square (2019) *Five things you may not know about the PS4's Controller.* Available at: http://www.pushsquare.com/news/2013/03/five_things_you_may_not_know_about_the_ps4s_controller [Accessed: 1 May 2019].

RobotShop (2019) *Flexinol® Technical Data.* Available at: https://www.robotshop.com/media/files/PDF/flexinol-technical-data.pdf Accessed [19 May 2019].

Simon Fraser University (2017) *Technology puts 'touch' into long-distance relationships*. Available at: http://www.sfu.ca/university-communications/media-releases/2017/technology-puts-touch-into-long-distance-relationships.html [Accessed: 1 May 2019].

TeslaSuit (2019) *TeslaSuits suit information* Available at: https://teslasuit.io/ Accessed [1 May 2019].

TeslaSuit business (2019) *TeslaSuits market ideas* Available at: https://teslasuit.io/business-solutions/ Accessed [1 May 2019].

Texas Instruments (2016) *LM317 3-Terminal Adjustable Regulator* Available at: http://www.ti.com/lit/ds/symlink/lm317.pdf Accessed [19 May 2019].

TN Games (2019) *TN Games Haptic vest* Available at: http://tngames.com/ Accessed: [1 May 2019].

University Saarland (2015). *"Artificial hand able to respond sensitively thanks to muscles made from smart metal wires."* ScienceDaily. ScienceDaily, 24 March 2015. Available at: https://www.sciencedaily.com/releases/2015/03/150324084716.htm Accessed [19 May 2019].

Xbox (2019) *Xbox Elite Wireless Controller.* Available at: https://www.xbox.com/en-GB/xbox-one/accessories/controllers/elite-wireless-controller Accessed: [1 May 2019].

15 Appendix

15.1 Code

All of the code that was used in this project is available from:
https://github.com/bwickenden/PROJ325

The final code for both the Arduino and Unity is in the Project Zip Files folder. In the same folder is the Unity Assets

With the original Gantt chart and project documents in the Final Docs folder

15.2 Videos of Work

Nitinol Lifting 300g: https://www.youtube.com/watch?v=t1pBcG1ICdk

Nitinol Activation test: https://www.youtube.com/watch?v=EPpVOeJtwWA

Nitinol connected to LM317 with 9V supply test:
https://www.youtube.com/watch?v=4yOBEJD1VjM

Nitinol Glove working with 1 finger connected to final circuit:
https://www.youtube.com/watch?v=B5BW7MMWhCI

Nitinol Glove working with all fingers whilst connected to final circuit:
https://www.youtube.com/watch?v=DS6wZaGPQJo

Prototype animation of moving hand:
https://www.youtube.com/watch?v=pUMAnP8jDx8

Testing Animation manager in Unity:
https://www.youtube.com/watch?v=XCxy7JP3XBU

Testing Unity and Arduino coms with LED:
https://www.youtube.com/watch?v=XKsxaWLxkYQ

15.3 Tables

Table 1:

Controlls	
Key:	**Action:**
G	Moves Thumb (Nitinol)
H	Moves Index Finger (Nitinol)
J	Moves Middle Finger (Nitinol)
K	Moves Ring Finger (Nitinol)
L	Moves Little Finger (Nitinol)
Y	Thumb Motor
U	Index Finger Motor
I	Middle Finger Motor
O	Ring Finger Motor
P	Little Finger Motor
F	Stop Action

Table 2:

Arduino Pin Map	
Device	**Pin Number**
Little Finger Nitinol	Pin 13
Ring Finger Nitinol	Pin 12
Middle Finger Nitinol	Pin 11
Index Finger Nitinol	Pin 10
Thumb Motor	Pin 7
Index Finger Motor	Pin 6
Middle Finger Motor	Pin 5
Ring Finger Motor	Pin 4
Little Finger Motor	Pin 3

Table 3:

Wire colour coding			
Type	**Colour**	**Pin**	**Location**
Nitinol	Grey	Pin 13	Little Finger
Nitinol	Organge	Pin 12	Ring Finger
Nitinol	Yellow	Pin 11	Middle Finger
Nitinol	Purple	Pin 10	Index Finger
Motor	white	Pin 7	Thumb
Motor	white	Pin 6	Index Finger
Motor	white	Pin 5	Middle Finger
Motor	white	Pin 4	Ring Finger
Motor	white	Pin 3	Little Finger

Table 4:

Current Limiting ciruit With 5V supply		
Resistance Ω	Volage across Load (V)	Current Across Load (mA)
11	2.5	227
7.3	2.288	313
5.5	2.016	366
4.4	1.798	408
3.6	1.541	428
3.1	1.32	425

Table 5:

Current Limiting ciruit With 9V supply		
Resistance Ω	Volage across Load (V)	Current Across Load (mA)
11	2.9	263
7.3	2.5	342
5.5	2.3	418
4.4	1.833	417
3.6	1.345	376
3.1	1.137	355

Table 6:

Minimum Current testing				
Current (mA)	Original length	Final Length (mm)	Change in length (mm)	Notes
200	105	105	-	No Change
210	105	105	-	No Change
230	105	105	-	Tension in wire
240	105	103	2	First movement
250	105	102	3	
260	105	101	4	
270	105	100	5	
280	105	100	5	

Table 7:

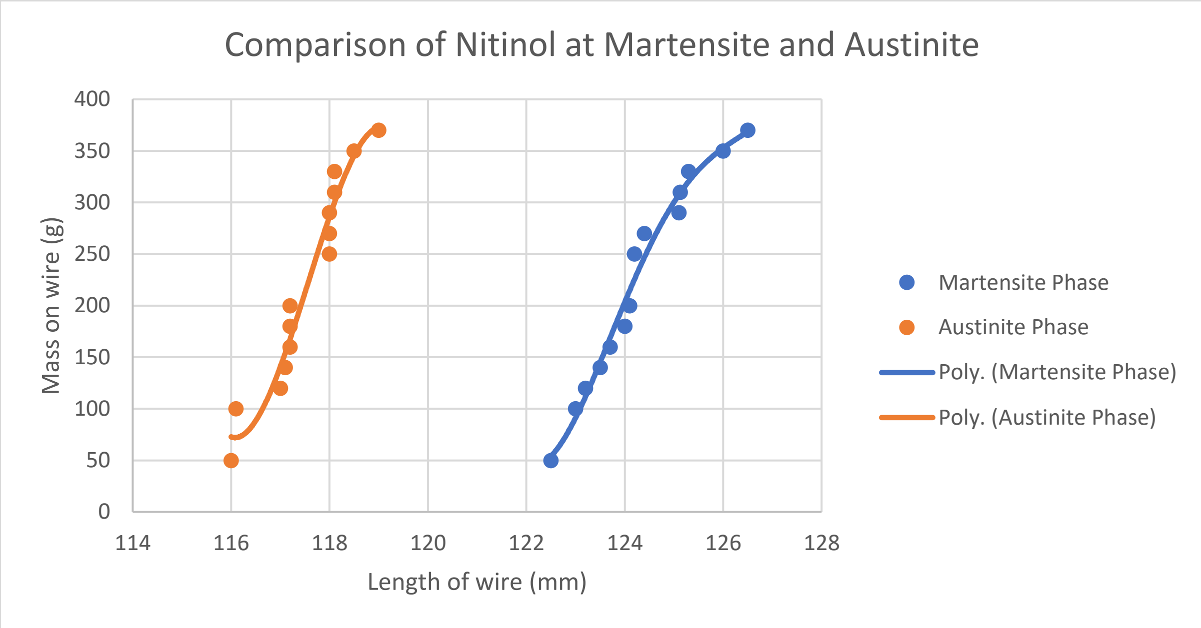
Nitinol under increasing weight					
Mass (g)	**Start length (mm)**	**Finish Length (mm)**	**Change in length (mm)**	**Stress (Pa)**	**Strain**
50	118	113	5	27711864	0.0424
100	117	113	4	55423728	0.0513
150	118	112	6	83135593	0.0508
200	118	113	5	110847457	0.0423
250	118	111	7	138559322	0.0593
300	117	110	7	166271186	0.0598
350	118	109	11	193983050	0.094
400	Break	Break	Break	Break	Break

Table 8:

Comparison of Nitinol at Martensite and Austinite					
Room temp			~90 temp		
Mass (g)	**Length of wire (mm)**	**Extension (mm)**	**Mass (g)**	**Length of wire (mm)**	**Extension (mm)**
50	122.5	0	50	116	0
100	123	0.5	100	116.1	0.1
120	123.2	0.7	120	117	0.9
140	123.5	0.5	140	117.1	0.1
160	123.7	0.5	160	117.2	0.1
180	124	0.5	180	117.2	0
200	124.1	0.4	200	117.2	0
250	124.2	0.2	250	118	0.8
270	124.4	0.3	270	118	0
290	125.1	0.9	290	118	0
310	125.13	0.73	310	118.1	0.1
330	125.3	0.2	330	118.1	0
350	126	0.87	350	118.5	0.4
370	126.5	1.2	370	119	0.5

15.4 Graphs

Graph 1:



15.5 Figures

Figure 1:

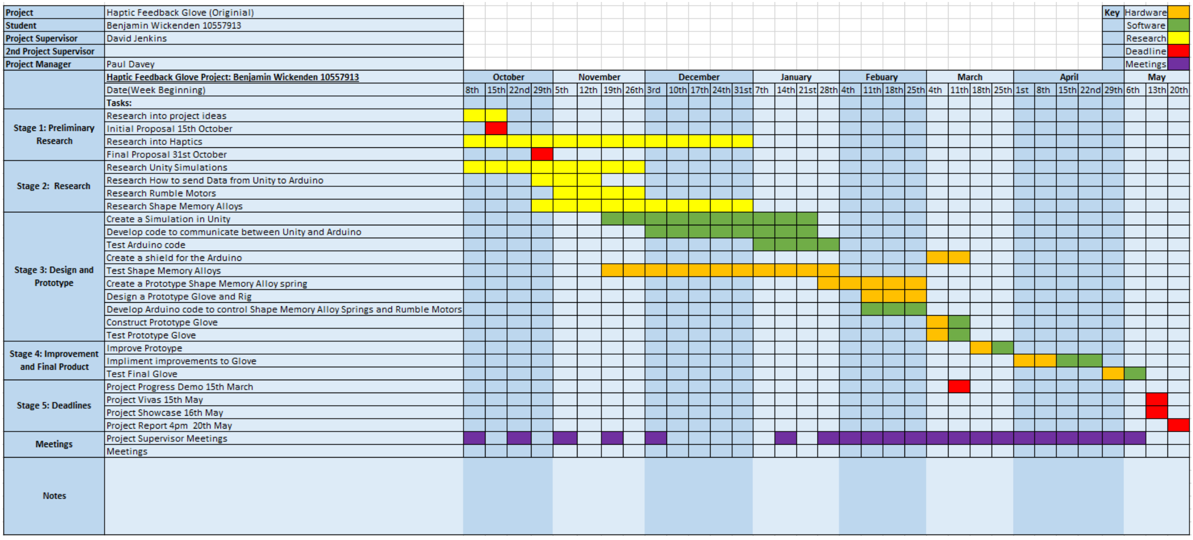


Figure 2:

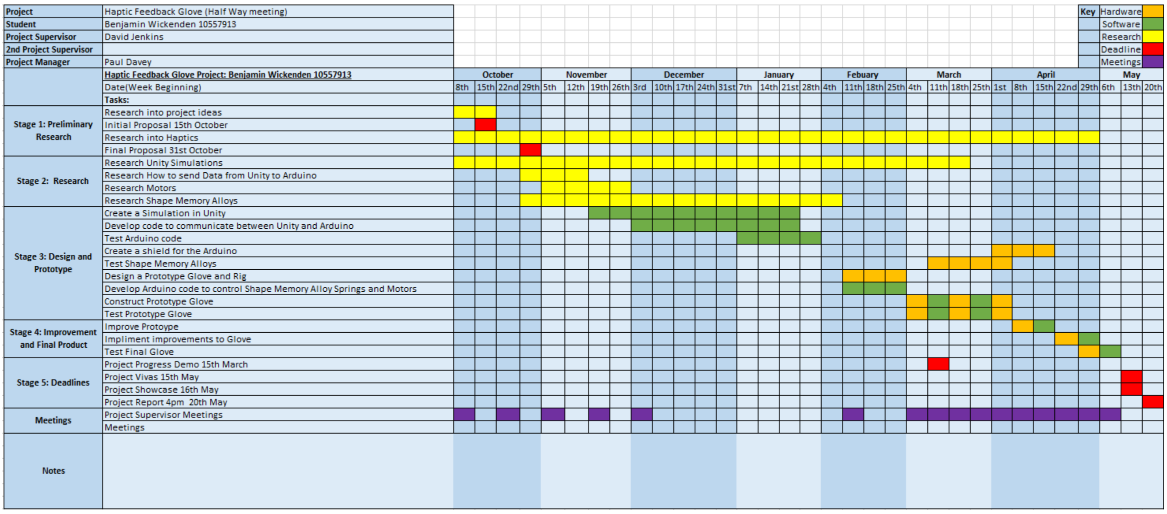


Figure 3:

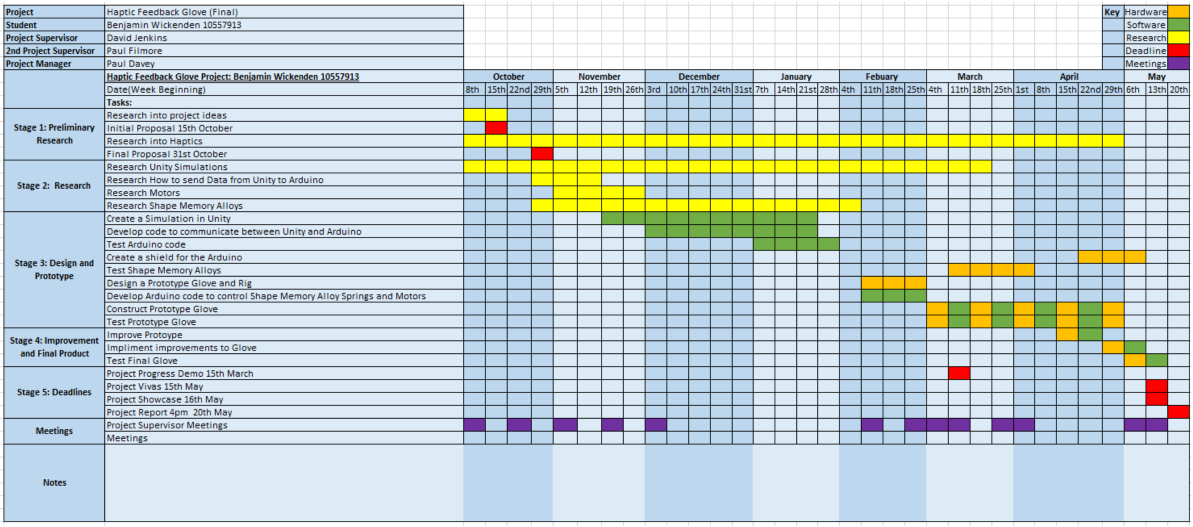


Figure 4:

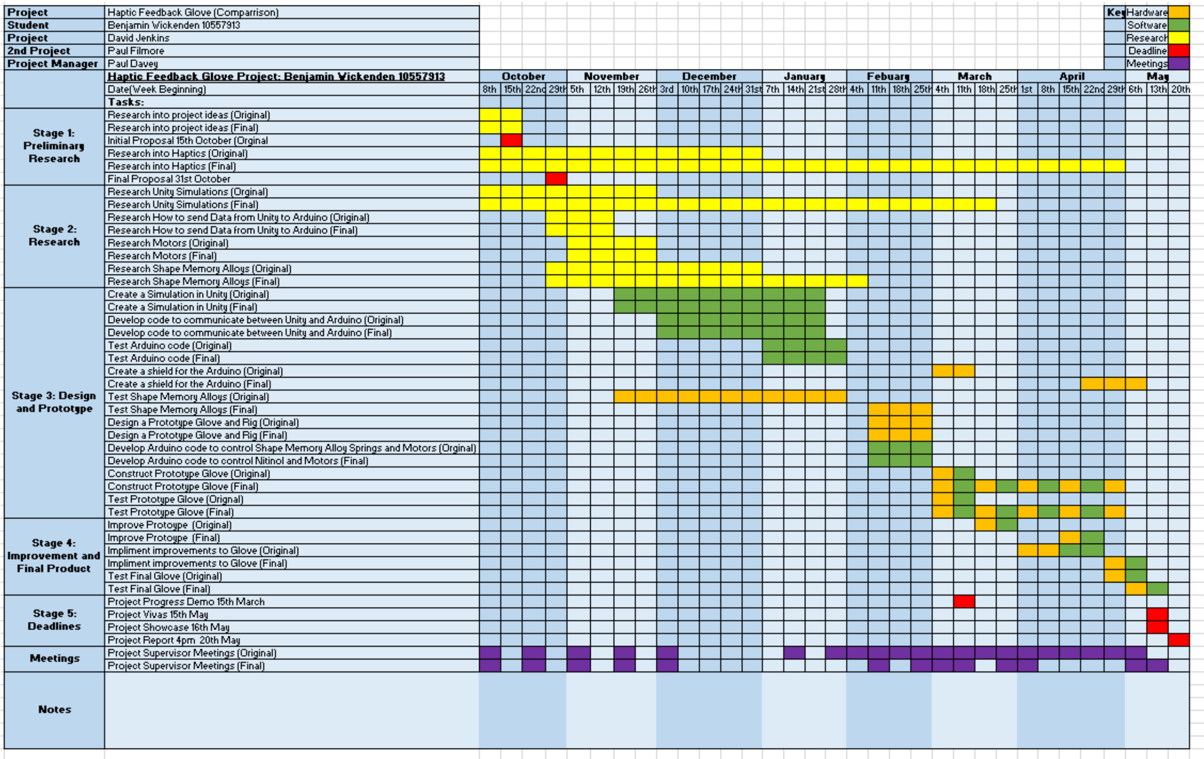


Figure 5:

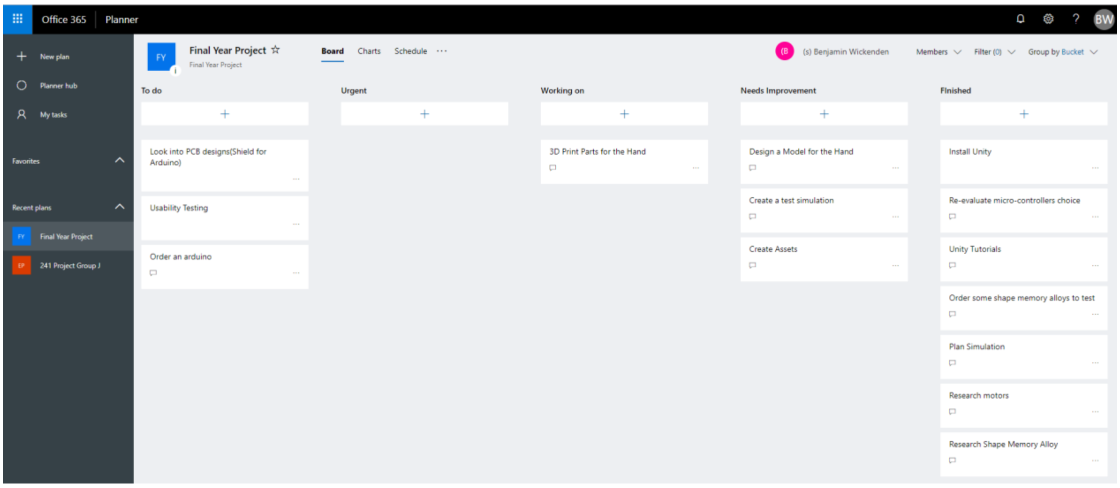


Figure 6:

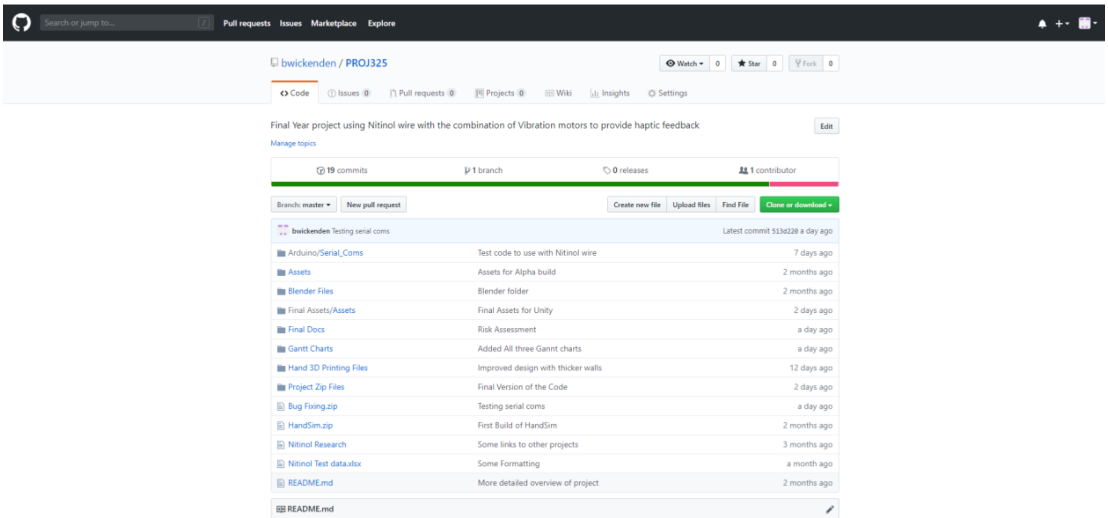


Figure 14:

Figure 21:

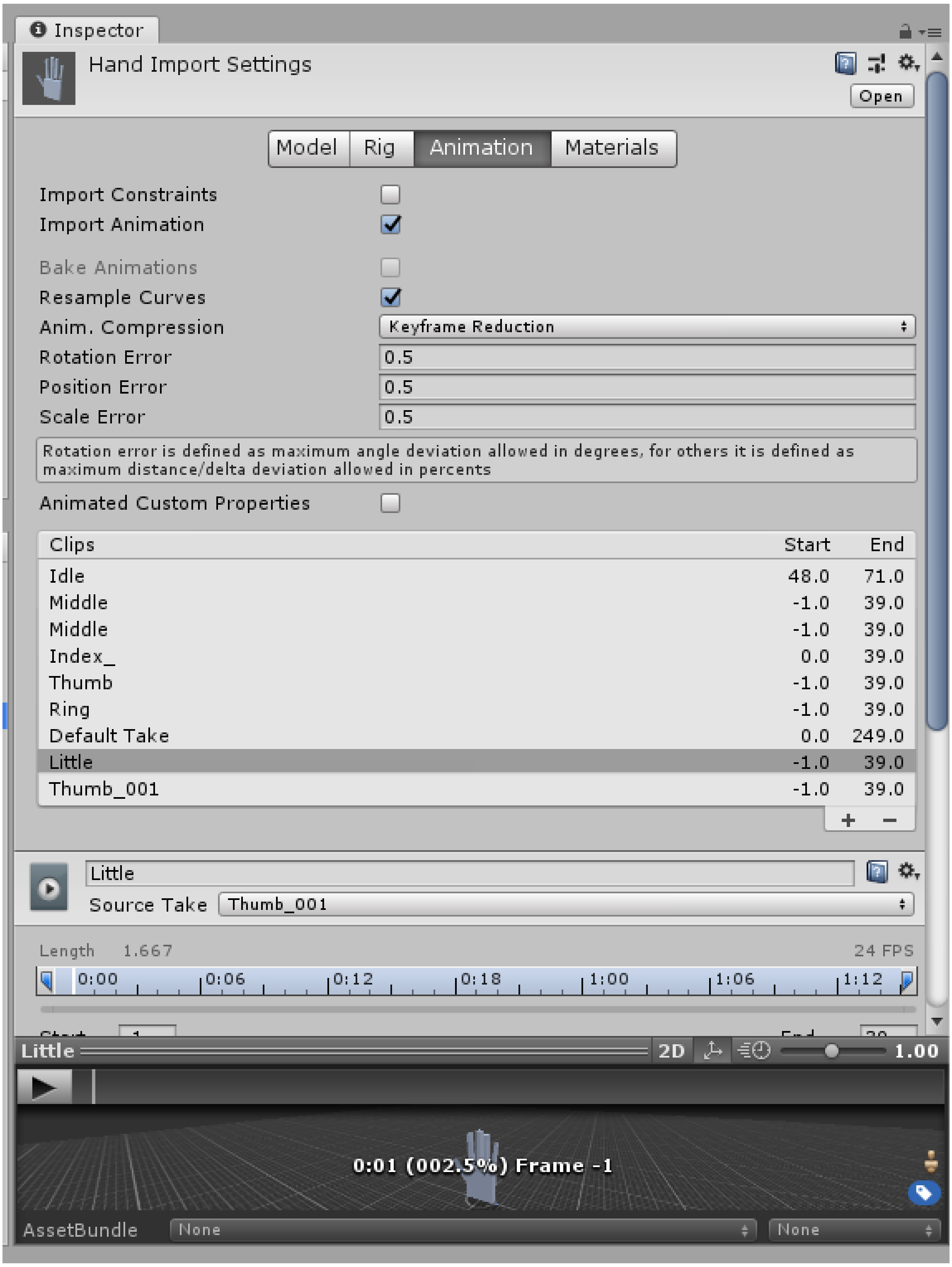
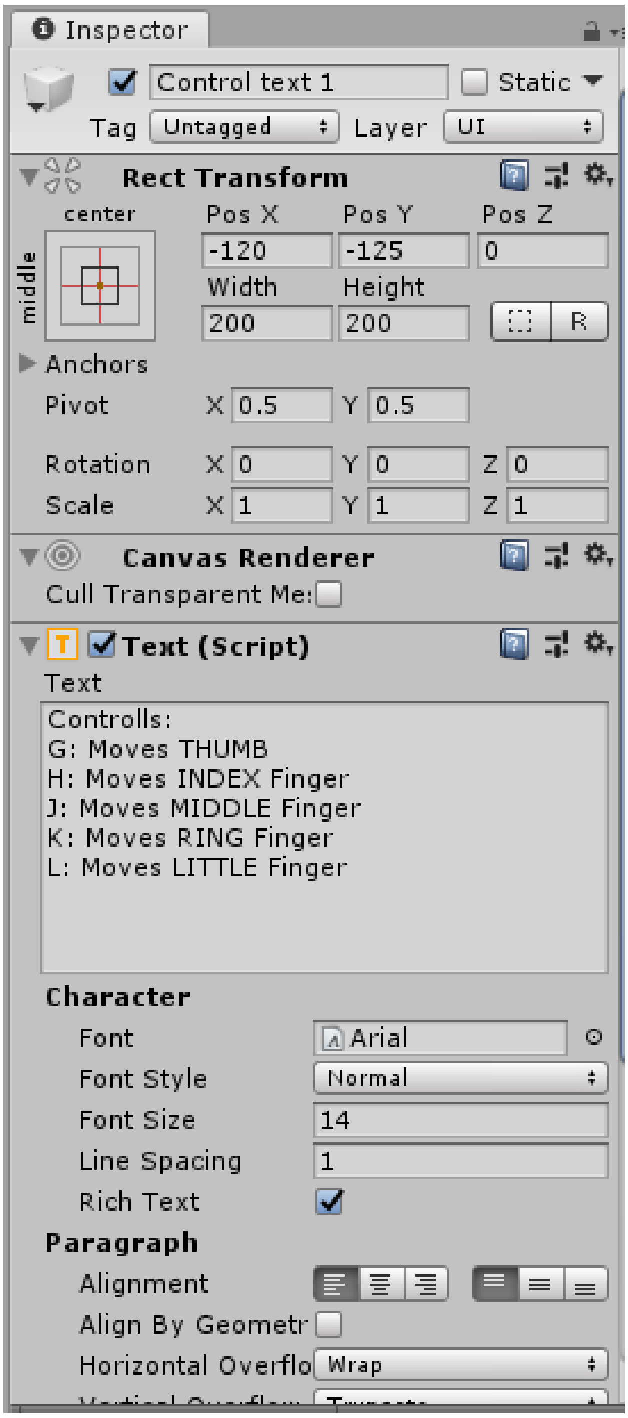
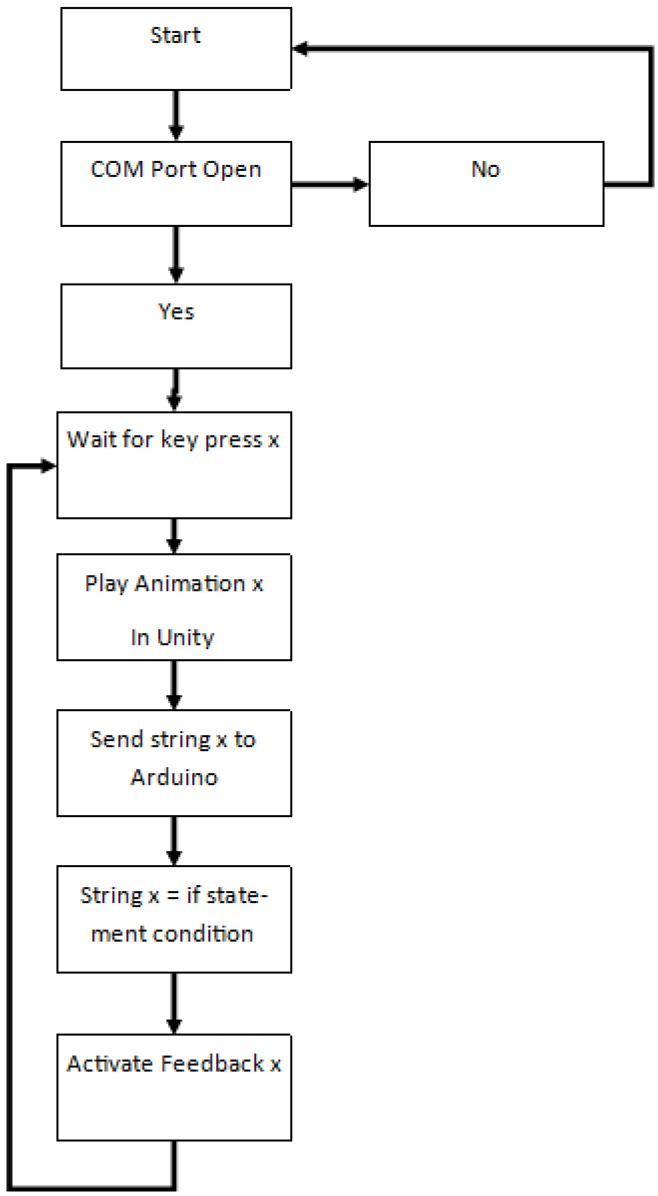

Inspector
Hand Import Settings
Open
Model
Rig
Animation
Materials
Import Constraints
Import Animation
Bake Animations
Resample Curves
Anim. Compression
Keyframe Reduction
Rotation Error
0.5
Position Error
0.5
Scale Error
0.5
Rotation error is defined as maximum angle deviation allowed in degrees, for others it is defined as maximum distance/delta deviation allowed in percents
Animated Custom Properties
Clips
Start
End
Idle 48.0 71.0
Middle -1.0 39.0
Middle -1.0 39.0
Index_ 0.0 39.0
Thumb -1.0 39.0
Ring -1.0 39.0
Default Take 0.0 249.0
Little -1.0 39.0
Thumb_001 -1.0 39.0
Little
Source Take
Thumb_001
Length 1.667
24 FPS
0:00
0:06
0:12
0:18
1:00
1:06
1:12
Little
2D
1.00
0:01 (002.5%) Frame -1
AssetBundle
None
None

Figure 26:


Inspector
Control text 1
Static
Tag Untagged
Layer UI
Rect Transform
center
middle
Pos X
Pos Y
Pos Z
-120
-125
0
Width
Height
200
200
R
Anchors
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Canvas Renderer
Cull Transparent Me:
Text (Script)
Text
Controlls:
G: Moves THUMB
H: Moves INDEX Finger
J: Moves MIDDLE Finger
K: Moves RING Finger
L: Moves LITTLE Finger
Character
Font Arial
Font Style Normal
Font Size 14
Line Spacing 1
Rich Text
Paragraph
Alignment
Align By Geometr
Horizontal Overflo Wrap

15.6 Flow Chart



15.7 Project Proposal

Key Information

Student: Benjamin Wickenden

ID:10557913

Module: PROJ325

Project Title: Haptic Feedback Glove

Project Supervisor: David Jenkins

Project Manager: Paul Davey

Background

I have decided to base my project around Haptic feedback. The largest market for haptic feedback is currently the games industry where Haptic feedback is being used to increase the users immersion to provide a better gaming experience. Haptics have been around in the games industry since the late 1990's with the introduction of Nintendo's Rumble Pak for the N64, however with the recent breakthroughs and rapid development of VR there has been a demand for more complex types of Haptic Feedback. Haptics have also been implemented into a variety of different industries in recent years, one example is that companies are now investing in VR and Haptics to aid the training of new staff, this method has proven to both cost effective and provides a safe learning environment. In addition, Haptics have also been implemented for Medical use in a variety of different ways with one being with the use of a master and slave rig for certain surgeries. Lastly communication companies such as Skype are looking into ways to make their users feel more connected to each other when in a video call, this has the potential to implement a form of Haptic feedback that would allow the users to hold hands.

Scope

For my project I've decided that I want to focus on providing compressive feedback onto a hand using several shape memory springs. I also plan to use rumble motors alongside the springs to provide two different types of Haptic Feedback, and I plan to control both systems using an Arduino Uno. Alongside creating the hardware for the project, I will also be creating a simulation using the Unity Engine to provide a simulation for the user. The idea is that the Haptic feedback will make the user feel more immersed in the simulation as they will be able to connect with the digital environment physically. In order to do this, I will export data from Unity to Arduino using the USB serial port. This will then allow me to write to variables in the Arduino code, which values can be used to drive the motors and springs. The simulation will be relatively simple and will include an object stimulating your hand. As of now I haven't used Unity before and it will depend on how my skills increase over the next couple of months to whether I can create a more advanced simulation, but my main focus and interest of this project is the shape memory alloy springs that are providing the Haptic feedback. The project will also require me to 3D print a series of objects such as components to make up the glove and housing for the microcontroller.

Objectives

- Create a Simulation in Unity that can output Data to Arduino
- Design the Physical glove that the user will wear
- Control Rumble motors using Data from the Simulation
- Create a Shape Memory Alloy that can be used to provide a compression force onto a hand
- Control the Shape Memory Alloy using the data provided to the Arduino from the Unity simulation and to have all systems working together

Timeframe / Milestones

I've broken down each task that I need to complete with their allotted time frame in my Gantt Chart, however bellow I've listed each stage and what that consists of.

	Description of Work	Start and End Dates
Stage One	Preliminary Research	September to October
Stage Two	Research	October to January
Stage Three	Design and Prototype	November to March
Stage Four	Improvement and Finial Product	March to May

Project Budget

At the current moment of time I don't have the exact figure of each component, so I've provided an estimate for each area of the project and have allocated a £20 buffer incase I've underestimated any costs.

	Description of Work	Anticipated Costs
Phase One	NA	NA
Phase Two	NA	NA
Phase Three/Four	Shape Memory Alloy	£20
	Arduino Uno	£30
	Rumble Motors	£10
	Misc. Components	£20
	Total	**£ 80.00**

Monitoring and Evaluation

During the first semester I've planned to have fortnightly contact with my project supervisor David Jenkins however once the second semester begins I plan to up that to weekly contact to ensure that all deadlines are meet and achieved. In addition, I will also be keeping a log book and writing a report.