UNSW

ENGG1000
Engineering Design & Innovation

Final Design Report

Semester 1 2018

Course Coordinator - Chris Menictus

Team 15

El-Kamand, Zane	z5204828	z5204828@ad.unsw.edu.au
Ling, Harvey	z5205919	z5205919@ad.unsw.edu.au
Meade, Oliver	z5208226	z5208226@ad.unsw.edu.au
Nguyen, Dan	z5206032	z5206032@ad.unsw.edu.au
Oo, Ryan	z5206676	z5206676@ad.unsw.edu.au
Sriram, Roshan	z5206049	z5206049@ad.unsw.edu.au
Wang, Wilson	z5209164	z5209164@ad.unsw.edu.au

Executive Summary

This document is the final report for Team 15's *Project Bionic Hand* which:

- discusses and justifies the design process of the bionic hand from the prototype;
- discusses the specifications of the bionic hand in a system overview and component level design;
- analyses the implementation of the bionic hand;
- analyses the results of the final testing; and
- discusses Team 15's recommendations for *Project Bionic Hand.*

The bionic hand encountered many issues such as the ineffectiveness of the motor, the gripping mechanism and weight of the hand. These were addressed after the acceptance testing and changes were implemented before final testing.

However, during final testing the bionic performed below average with a score of 58.5. The final bionic hand lacked aesthetics and had limited functionality, changes to the bionic hand during implementation were considered over the choice of materials due to increasing weight. The bionic hand was unable to pick up the water bottle and efficiently pick up household objects due to the limited functionality may be due to the overuse of the servo. The keypad input system was able to receive a bluetooth signal however the mechanism of this system was nonfunctional due to time and budget constraints.

Team 15 recommends the extensive use of PCBs over breadboards to increase robustness, increased use of 3D Modelling software to plan out the positions and configurations of all subsystems and wiring, as well as a more robust servo and lighter materials of a similar strength.

Team 15 recommends greater research into the servo and how to avoid wearing it out as well as reducing the weight of the claws and reconsidering the housing of the servo as the final designs' housing increased vulnerability to damage.

Table of Contents

1. Introduction

With reference to *ENGG1000 S1 2018 Detailed Project Description*, the specified objectives of Project Bionic Hand is to *"design and construct a gripper that is able to pour water into a cup and allow the user to drink, pick up objects, and press keys autonomously via bluetooth" (see appendix A)*. The *Projection Description* also outlined restrictions of the bionic hand design to be *"affixed to the forearm, activated by electromyography, cost no more than $150, and any electrical components are battery-powered."* The qualitative marking criteria also requires that the bionic hand be *"aesthetic, robust, simple and innovative."*

The bionic hand is a prosthesis that mimics the function of the human hand and aims to improve the quality of life of individuals amputated at the hand. This document is the final report for Team 15's Project Bionic Hand which will feature:

- the design process of the final bionic hand which consists of the system overview of the bionic hand prototype, how it was implemented and its performance in the acceptance testing.
- key problems which arose during the design process and solutions proposed by Team 15;
- the final design in its full specifications, system level design which shows how components are interfaced to achieve its specific function and component level design which analyses the purpose and function of the components; and
- the results of the final testing of Team 15's final bionic hand and Team 15's recommendations.

2. Prototype

The bionic hand prototype was the implementation of pre-acceptance testing systems and components. After acceptance testing, Team 15 assessed the performance and quality of the bionic hand for improvements before the final testing. This section of the report consists of system overview of the bionic hand prototype, its implementation, and the results of the acceptance testing and how it affected the design of the final bionic hand.

2.1 Prototype System Overview

The prototype was split into two separate systems to achieve the function of gripping and the function of a keypad input *(appendix D)*. This decision by Team 15 was made to simplify the solution to the problems outlined in the problem statement *(appendix A)*.

The gripping system by acceptance testing did not have EMG circuitry ready and only consisted of a three-pronged gripper; a ‘2+2’ arrangement with two pairs of opposing prongs *(see appendix E)*. The mechanism of the gripper relied on a DC motor rotating a worm screw which in turn rotated two worm gears that controlled the rotation of the prongs. Foam pads were used as a gripping material which aligned the inner surface of the fingers. The purpose of the foam pad was to allow the contouring of the gripper to mould to the shape of the object thus allowing a more fixed grip by balancing the moments across the object. The prongs had a concave shape to achieve the same effect by increasing contact points between the object and gripper for a more fixed grip. The dual prongs had an aluminium foil that acted as a “webbing”. This structural feature had a function in collecting small sized objects by using the prongs to slide these objects into the webbing.

The keypad input system design by acceptance testing consisted of a working bluetooth receiver and linear actuator prototype with bluetooth reception and processing code. *(see appendix F)*.

2.2 Implementation of Prototype

The prototype has been constructed from pine wood due to its low cost, availability and easy machinability *(see appendix B)*. Four prongs were cut using the bandsaw into the shape shown (see *appendix* E). Two of these were then hot glued together with a block of 2cm x 2cm separating them at each end of the prong to create a space between them as shown by the side view. This was done twice so that there would be 2 claws. After sanding correct any cutting errors each claw was then mounted onto a 3D printed gear as shown. They were then placed on either side of a worm gear which was mounted to a DC motor. The screw gear turns the gears attached and hence rotate each claw in opposite directions (allowing for the motion of opening and closing). This, along with the electronics, was all contained in a plywood housing with dimensions: 20x13x10(cm) (l x w x h) to keep the claws upright. This was mounted to the arm using velcro straps the chassis, attaching the arm to the underside of the bionic arm.

2.3 Acceptance Testing

The task of the acceptance testing required Team 15 to:

- ➢ pick up and turn an empty water bottle *(3 marks)*;
- ➢ pick up the smallest and largest object *(3 marks)*;
- ➢ receive a code via a bluetooth signal *(2 marks)*; and
- ➢ hand in a completed risk assessment for the device *(2 marks)*.

The acceptance test provided an opportunity for Team 15 to test the capabilities and limits of the gripping system and bluetooth system of the bionic hand. Observations of the bionic hand during testing were made and tabulated in appendix C.

Team 15 received 10/10 for the acceptance test as the prototype was able to pick up and turn a water bottle, pick up the largest and smallest object due to the correct dimensions of the gripper, and able to receive a bluetooth signal. This overall showed that the general prototype design was appropriate for *Project Bionic Hand*.

Despite passing the acceptance testing, there was observed limitations to the function of the prototype's gripper. Very small objects such as the 50 cent coin was difficult for the gripper to pick up (using the "sliding" technique) and thus consumed time. The proposed solution to this was to change the inherent structure. The bottom claw was to be longer than the top claw, therefore the top claw can be used to slide small objects off the table (where objects sat) into the palm of the longer claw thus increasing allowing the bionic hand user to pick up small objects easily.

Prototype was not able to effectively grip the rockmelon. This was due to the weak grip force applied onto the rockmelon due to insufficient gearing and/ or torque supplied by the motor. The proposed solution was to replace the motor with a servo to generate greater torque which is necessary for the gripping of heavier objects.

The rotation of the claws was not smooth and claws were not properly fixed to the chassis as the claws were glued directly atop their respective worm gears and worm gears screwed to the chassis. Thus prototype was unreliable in gripping heavier objects as claws were not robust enough. This can be solved through the use of an axle to allow the rotation of the claws about a fixed point whilst having the claws fixed to the chassis to allow a more robust rotation.

The gears used in the prototype were susceptible to wear as these gears were 3D printed from polylactic acid (PLA). The proposed solution was to replace the gear system with a much simpler gripping mechanism which involved the direct attachment of a moving claw to the servo and fixing the other claw to the chassis. This design limits wear on the mechanism and is easier to manufacture due to its simplicity.

Aluminium foil which was used as webbing to collect small objects was prone to ripping and was unaesthetic to the wooden chassis of the bionic hand. Therefore this material was to be replaced with a more robust material such as a nylon net.

These changes would be implemented into the final design which aims to improve the performance of the bionic hand for final testing.

3. System Overview of Final Design

Based off the prototype, the final bionic hand remained resolved into two systems: gripping and keypad input. This section of the report will provide an overview of the final bionic hand in its two systems; outlining the purpose and structure of the systems.

3.1 Gripper System

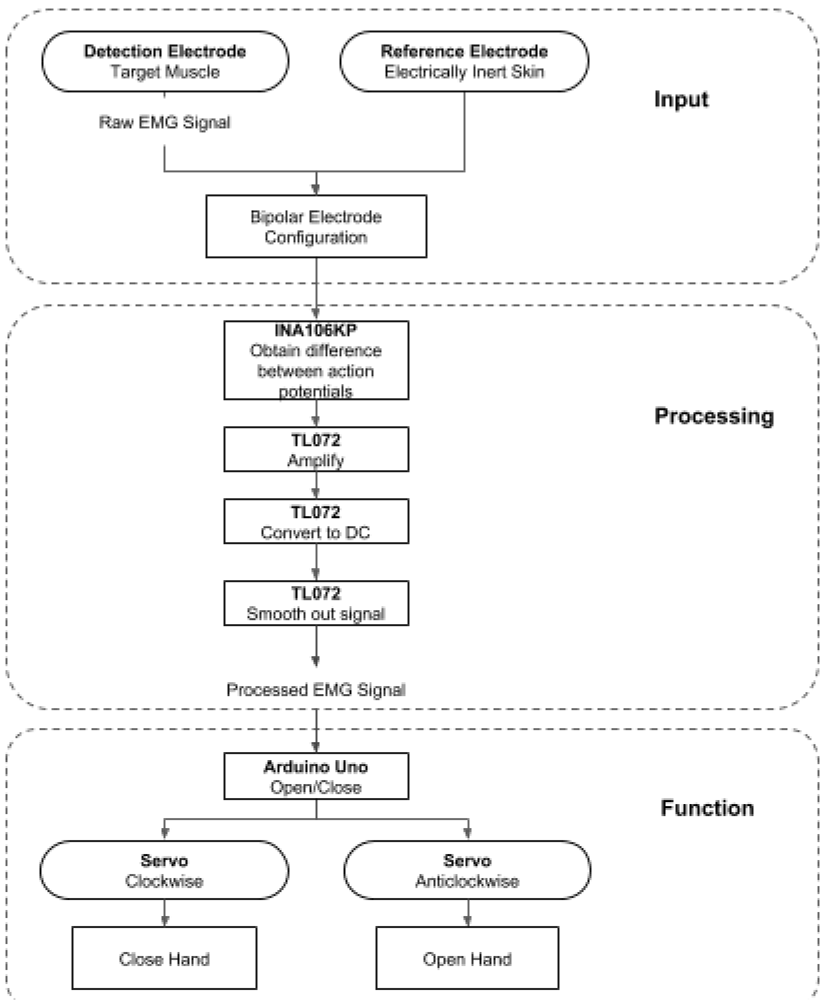


Figure 1 - Gripping Block Chart

Figure 1 shows the components of the gripping system and the interactions between each component.

Input Subsystem

A bipolar configuration of electrodes (two detection electrodes on the bicep and one reference electrode on the olecranon) was used as a bipolar configuration facilitates the removal of common noise between the two detection electrodes (using the INA106KP chip), providing a cleaner signal as opposed to the use of a monopolar configuration.

Processing Subsystem

This signal needs to be amplified (using a TL072 IC Chip) as the difference in electrical potentials between the two detection electrodes is too small in amplitude to be passed directly to the Arduino Uno. The signal is now an amplified AC signal and needs to be converted into a DC signal and smoothed out, so a DC signal of constant amplitude is sent into the Arduino. This task is carried out using the last two TL072 IC Chips.

Function Subsystem

A DC signal needs to be read by the Arduino to process this data and execute relevant actions. It reads DC voltages through one of its analog pins and then is able to calculate and record its. The Arduino needs to act as a switch for the gripping mechanism depending on if the user has their muscle tensed or not. Thus if the voltage received is higher than a preset threshold voltage, it moves the gripping mechanism by rotating the servo. The gripping system requires two degrees of rotation hence two EMG signals need to be recorded simultaneously. Depending on which EMG signal is above the threshold, the gripping system will rotate one way. Likewise, the gripping mechanism also needs to be stopped on command to increase accuracy of gripping. This stopping action is executed when neither of the EMG signals reach over the threshold.

3.1 Keypad Input System

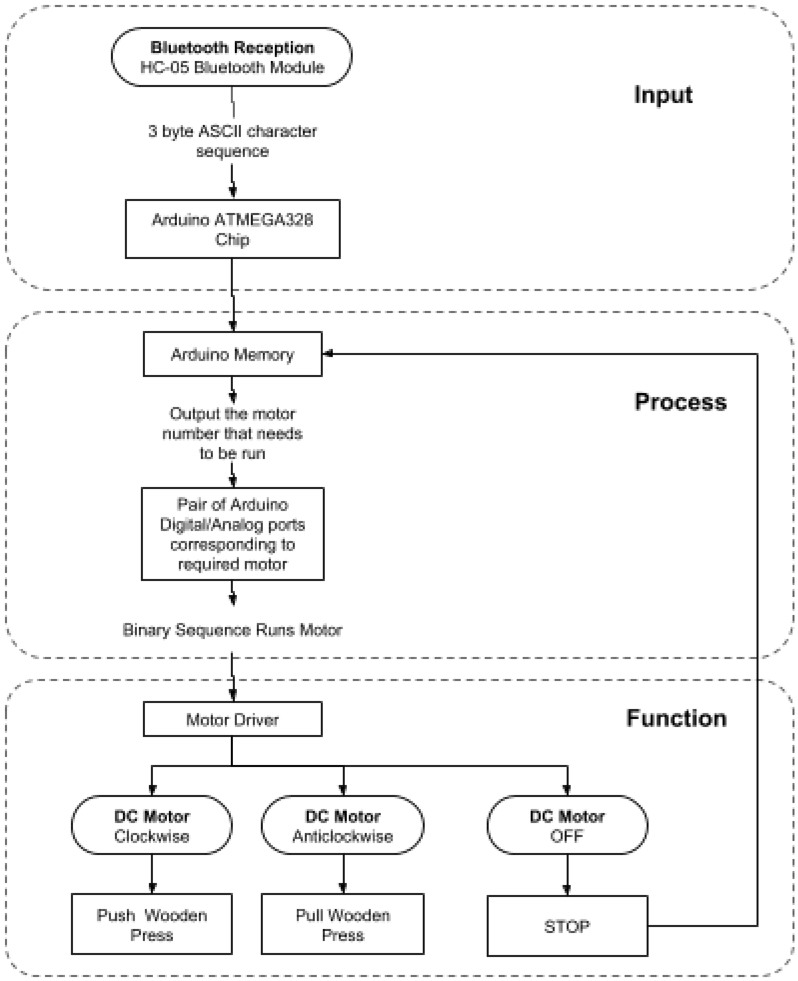


Figure 2 - Keypad Block Chart

Figure 2 shows the components of the keypad input system and the interactions between each component.

<u>Input Subsystem</u>

The HC-05 Bluetooth module is connected and paired to an external Bluetooth transmission source like a mobile phone. This allows a stable connection medium for a 3 byte data package to be sent through. This 3 byte data package contains a 3 ASCII character sequence that needs to be read by the keypad input system. The bluetooth module is connected to the Arduino's serial pins to allow communication between the bluetooth module and the Arduino and thus allow the Arduino to also receive and record the sequence.

Processing Subsystem

The sequence received by the Arduino needs to be then processed to execute the correct keypad input. It first needs to record the sequence and the Arduino does this by recording the sequence in a 3 element array which allows it to be accessed later. The Arduino then reads through the array to determine which buttons it needs to press and in what order. This allows an exact translation of the sequence sent by the Bluetooth source.

Function Subsystem

The pressing of the keypad buttons is done through a mini linear actuation device. This linear actuation device is made up of a wooden press joined with a threaded nut with a bolt joined to the axle of a motor. As the motor rotates, the wooden press and the nut extends outwards to press the keypad. As the motor rotates in the opposite direction, the wooden press retracts back to its neutral position. 10 separate motors are used to allow each button press to be controlled separate from the other buttons. This stock movement of the wooden press extending in and retracting back is controlled by signals sent to respective motor drivers. The motor drivers then rotate the motor with the direction of rotation dependant on what part of the pressing sequence it is up to.

4. Final Design at Component Level

This section of the report breaks down the system into its components and explains the function of each component and justifies the purpose of the components.

4.1 Gripper Component Level Design

4.2.1 Emg Detection

EMG signals are used to activate the bionic hand, hence EMG signals must be obtained from the targeted muscle using surface electrodes on the skin. An INA106KP IC Chip and a bipolar configuration of electrodes of the targeted muscle is used to achieve this.

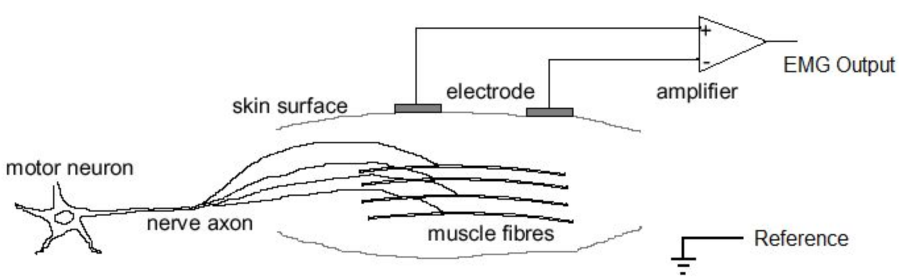


Figure 3 - Bipolar Configuration

Source: https://www.nrsign.com/monopolar-vs-bipolar-emg-readings/

Bipolar configuration requires the selection of a target muscle and a point of reference on an electrically neutral surface of the skin. Two surface electrodes must be placed in alignment of the muscle fibre of the target muscle on the surface of the skin approximately 1 cm apart so the EMG signal detected is the same and is assumed that the noise affecting this surface of skin will be common. These electrodes are then connected to the INA106KP chip. Figure 3 *(above)* shows an EMG signal from an electrode sent into the non-inverting input of the INA106KP chip and the other into the inverting input of the INA106KP. The INA106KP is a differential op-amp with a fixed gain of 10, thus it takes the difference between the voltages of the two signals and amplifies the difference. This removes the common noise between the two signals, allowing us to obtain a clean EMG signal.

4.2.2 Amplification

The EMG signals obtained are very small in amplitude (1-10 mV) and must be amplified to a suitable amplitude so it can be passed into the Arduino Uno. This amplification was achieved using a TL072 IC chip *(see figure 4, below)*, one 10k Ohm resistor and one 150k Ohm resistor. The EMG signal is connected to the inverting input of the TL072 chip via the 10k Ohm resistor. A feedback loop between the inverting input and the output of the TL072 chip was then created using a 150k Ohm resistor giving a gain of -15.

$$Gain = -\frac{R_2}{R_1} = -\frac{150\ kOhm}{10\ kOhm} = -15$$

The signal has now been inverted and amplified by a factor of 15. Next, the DC component of the signal is removed as this acts as a voltage offset. A 0.01uF capacitor is used to do this as it blocks the DC component of the signal, increasing the resolution of the signal. Post-amplification, EMG signal needs to be amplified again (as the removal of the voltage offset attenuates some of the signal), using a 453k Ohm resistor and a 150k Ohm resistor. The 453k Ohm resistor is connected to the inverting output of the TL072 chip, inverting and amplifying the signal by a factor of -3.

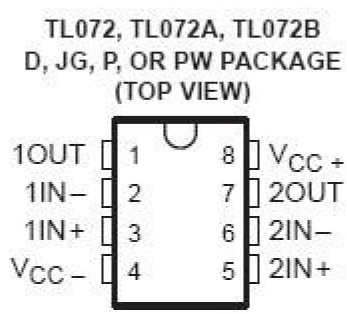


Figure 4 - TL072 Pinout Diagram

Source: Rory, **https://electronics.voltsandbits.com/projects/karaoke-circuit/**

4.2.3 Rectification

This stage of the EMG processing takes the absolute value of the EMG signal using a second TL072 IC chip, two 1N4148 diodes and jumper cables. Each wavelength consists of a positive voltage and a negative voltage *(see figure 5, below)*. During the positive cycle, the first 1N4148 diode becomes positively biased while the second 1N4148 diode become negatively biased. Hence, diode 1 is on and diode 2 is off. The current flowing through diode 1 is sent into the non-inverting input (pin 5 of the TL072 chip) and retains its positive voltage. During the negative cycle, diode 1 becomes negatively biased and diode 2 becomes positively biased. Hence, diode 1 is off and diode 2 is on. The current flowing through diode 2 is then sent into the inverting input (pin 2 of the TL072 chip) changing negative voltage into a positive voltage, thus the EMG signal is rectified.

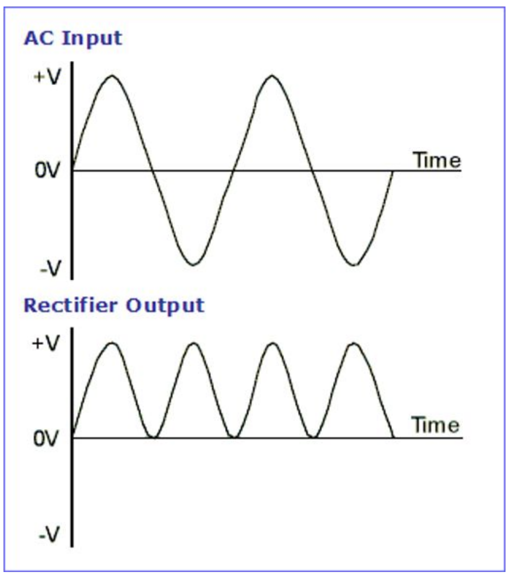


Figure 5 - Rectification of AC Input

Source: Surtell, **https://www.eleinmec.com/printerpage.asp?18**

The rectified EMG signal is converted into a DC signal for use in the Arduino Uno. This is achieved using a third TL072 IC Chip, a 80.6k Ohm resistor and a 1.0uF capacitor. The signal is passed into the inverting input of the TL072 chip. The 80.6k Ohm resistor and the 1.0uF capacitor then connect the inverting input and the output of the TL072 chip creating a low pass filter with a cut off frequency of 2Hz.

$$f = \frac{1}{2\pi RC} = \frac{1}{2\pi(1.0\ \mu F)(80.6\ kOhm)} = 2\ Hz$$

This cut-off frequency is so low, that the signal passing through the filter becomes a DC signal.

Team 15 confirmed that the circuit was working correctly using a Cathode Ray Oscilloscope.

4.2.4 EMG Interpretation

Post-rectification of the EMG signal, the signal needs to be read and interpreted to correctly run the gripping system. This interpretation is done through the Arduino Uno which acts as the communication device between the mechanical and electrical components of the gripping system. The Arduino first needs to appropriately read and process the DC voltage that comes from the EMG system. It does this by “reading” through one of the Arduino’s analog pins.

The inbuilt *analogRead()* function maps input voltages (received from the analog pin) into 10 bits of integers (1024 values). This means that it is able to read voltages between 0-5V (the operating voltage of Arduinos) and place an appropriate integer value to it. A simple calculation of 5V/1024 x (*returned integer*) will hence represent the exact voltage that was read by the analog pin *(see appendix K)*.

As there are two input voltages needed to be read (two EMG signals), both the A0 pin and A1 pin needed to be used. Likewise, they needed to also be connected to a common ground, which was any of the GND pins on the Arduino.

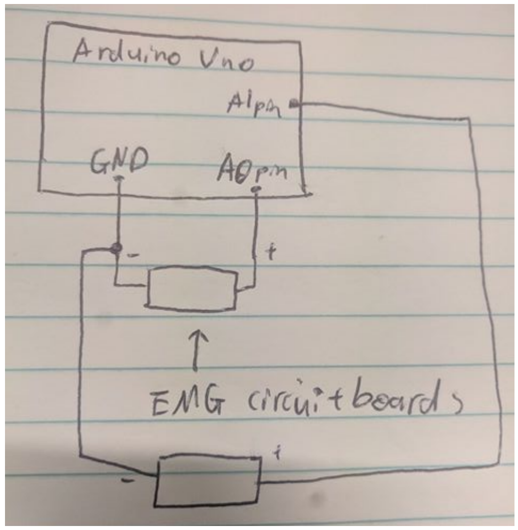


Figure 6 - Arduino, EMG circuit board interface (A0 and A1 pins as the analog pins)

Using the code *(appendix K)* and circuit above *(see figure 6, above)*, the EMG signal was able to be properly debugged and understood. The *Serial.println()* function that prints the voltages and a new line read every 100 microseconds allows the reading and comparison of the precise values of the received voltages from the data dump displayed on the Serial monitor.

From testing with the operator, the EMG signals were determined to be consistent about both EMG circuits and the bounds of the EMG signal were also defined *(appendix J)*. It was observed that a signal of 30 mV was continuously outputted, while a signal consistently above 600 mV was observed when the user tensed a targeted muscle. Therefore the appropriate threshold voltage of 600 mV was decided to act as the switch mechanism for the rest of the gripping system. From experimentation of different muscle groups, it was found that the chosen muscle will consistently output a signal above 600 mV. Therefore the detection muscle for the bionic hand is decided by the operator.

4.2.5 Controlling the Gripping System

The EMG signal acted as a switch to operate the mechanical gripping system. The entire mechanical gripping system is run by a high power HOBBYTech YM2763 servo. The servo is activated when the EMG signal is above the threshold voltage, thus the servo rotates when the operator tenses the appropriate muscle and stops rotating when there is no muscle tension. Each EMG circuit and its signal is assigned a specific direction of rotation and thus the direction the servo rotates is determined by which muscle the operator tenses.

The way servos are engineered are such that it will always try to correct the mechanical system to reach the servo's desired angle of rotation. A standard servo like the YM2763 is able to *write* (rotate to) a specific angle within 0-180 and continues to rotate to that angle until it is reached or a new command is given to it. The commands given to the servo is determined by the nature of the signal that passes through the *signal wire*. The Arduino *servo.h* library which contains the *servo.write()* function allows the Arduino to send an appropriate PWM (pulse width modulation) digital signal through the servo signal wire to write the desired angle. *Ie servo.write(180)* tells the servo to rotate to the position of 180 degrees. To achieve this, the servo needs to be powered by an external high current battery (the Arduino only operates at around 20mA), have the servo signal cable attached to an appropriate PWM digital pin and be grounded with the Arduino (to complete the circuit) *(see figure 7, below)*.

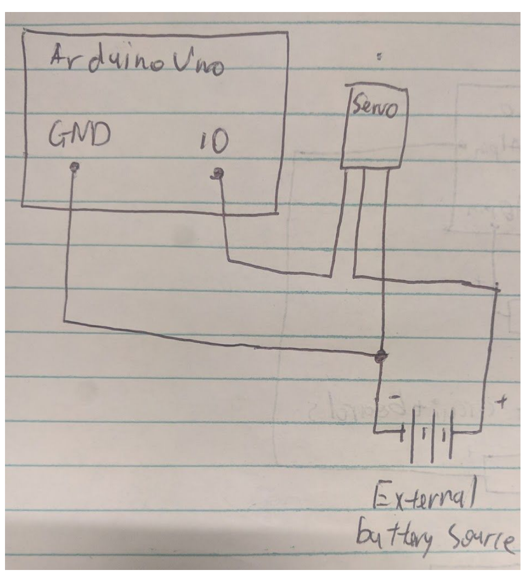


Figure 7 - Arduino, Battery Source Interface (use of digital pin 10)

In order to allow the servo to continuously rotate during tensing and stop on command when not tensing, the Arduino needs to steadily increment the angle it writes to the servo when threshold voltage is reached and stop the servo by continuously writing the current angle when threshold voltage isn't reached. By writing the current angle instead of stalling the servo to stop the gripping, the gripping system has a longer operating life and also allows us to be more delicate with our gripping mechanism. The incrementation of the angle needs to be controlled by the *delay()* function which pauses the operation of code as the servo needs to be given time to rotate 1 degree (about 10ms). This is as the high clock speed of the Arduino (around 8Mhz) doesn't allow enough time for the servo to rotate. It is also noted that a higher delay results in a slower rotation allowing the speed of the gripping system to be controlled. Limited by the geometry of the gripping system, the maximum opened state is at 90 degrees and maximum closed state is at 180 degrees. The default starting state is set at 90 degrees and the servo immediately writes this as the arduino is turned on so the gripping starts consistently every time.

4.2.6 Gripping Mechanism

The movable claw is affixed directly onto the rotating surface of the servo. The claw rotates about the servo's axis, while a supporting axis is placed on the opposite side of the claw to allow for a more controlled rotation of the claw. The rotation of the movable claw will follow the same motion as the servo within the range of 90 degrees to 180 degrees. The opposite claw is fixed to the chassis to decrease wear on the gripping mechanism and to simplify manufacturing and assembly processes.

The concavity of the gripper and the contouring of the foam pad will enable a more balanced grip on the object negating any sideways forces and moments acting on the object, as well as equally distributing the applied force along the full length of the pad. A friction pad was added onto the foam which provides greater static friction between the gripper and the object, thus less torque would be required to hold the object.

The gripping mechanism still utilises the sliding technique that was demonstrated in the acceptance testing prototype and included the changes discussed from the results of the acceptance testing. The webbing of choice was a nylon net however due to budget and time constraints, this was not implemented and will be discussed in the final testing section of the report.

4.3 Keypad Input Component Level Design

4.3.1 Bluetooth Module

The HC-05 bluetooth module is paired and connected to an external source *(refer to figure 8)*. This external source then sends a 3 byte data package which encodes the keypad instructions on it in the form of an ASCII sequence. The data received is then sent to the Arduino which processes the signal.

The Arduino outputs 5V in its transmitting port (TX port) however the Bluetooth module's receiver pin only supports 3.3V. Hence a voltage divider is constructed such that the Bluetooth module does not burn. This voltage divider is constructed through the use of 1 and 2 kOhm resistors.

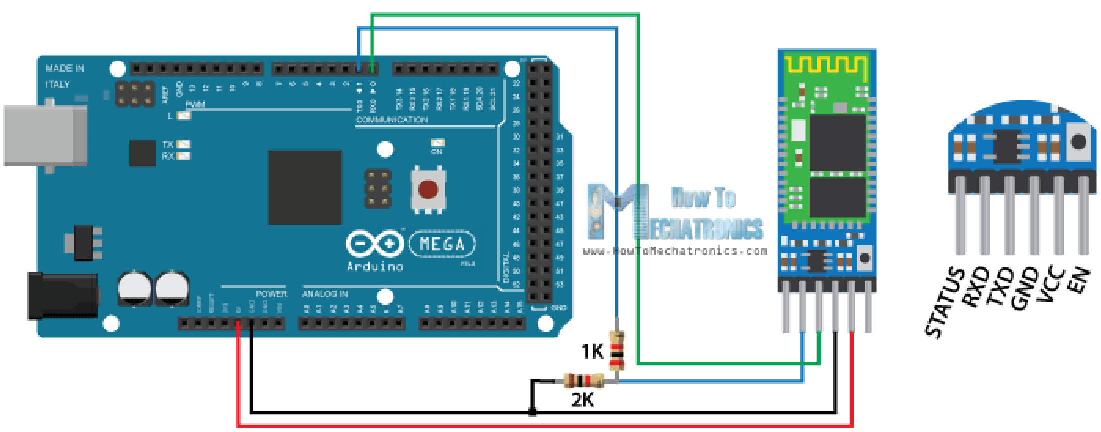


Figure 8 - Arduino and Bluetooth Module Connection

The *Serial.read()* function allows the Arduino to read the electrical signals sent to its serial port and interpret it as data. In this case, as an ASCII character is sent over, the ASCII character's code is returned. This allows the code to be stored in an array to be processed later *(see appendix M)*.

4.3.2 Sending Data to Motor Drivers

After recording 3 numbers, the first number in the array is read and depending on which ASCII character it represents, it activates a relevant pair of digital/analog output ports on the Arduino. These ports are connected to a H bridge circuit which acts as a motor driver to operate the mechanical pressing system. The L293D motor driver is able to power and operate two motors simultaneously directly from the Arduino *(refer to figure 9)*. Thus 5 motor drivers run 10 small motors to account for 10 different possible key presses. A total of 20 digital ports in the Arduino MEGA is allocated for this function.

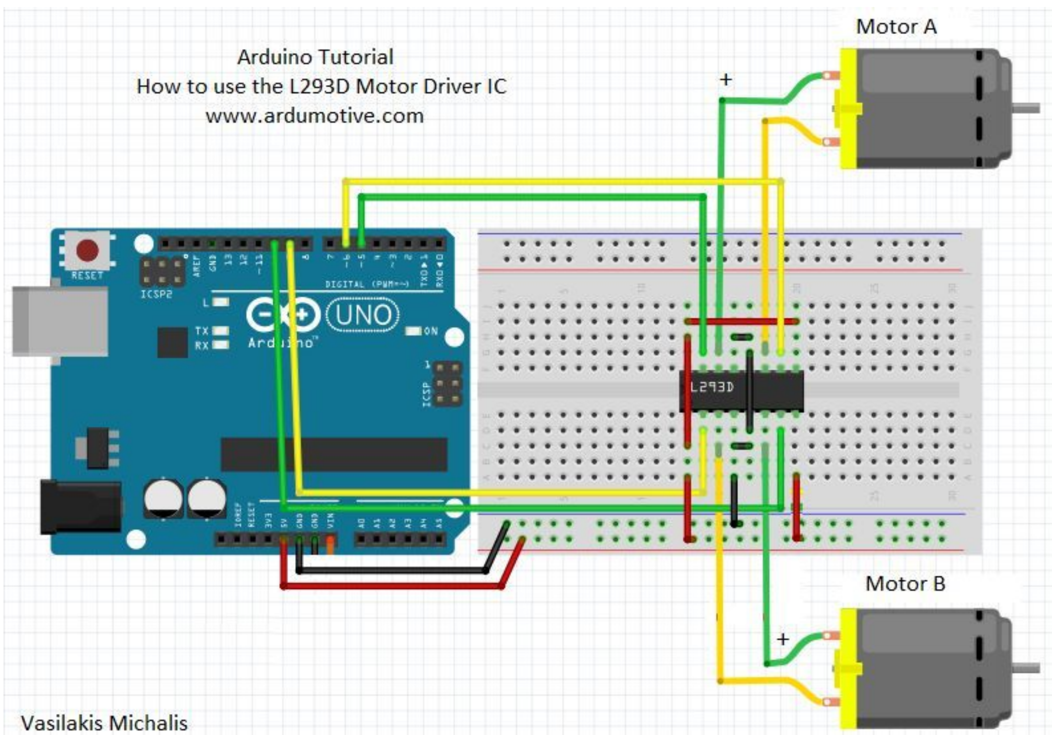


Figure 9 - Motor Driver Interface with Two Motors (keypad input system involves 5 motor drivers)

4.3.3 Keypad Pressing Mechanism

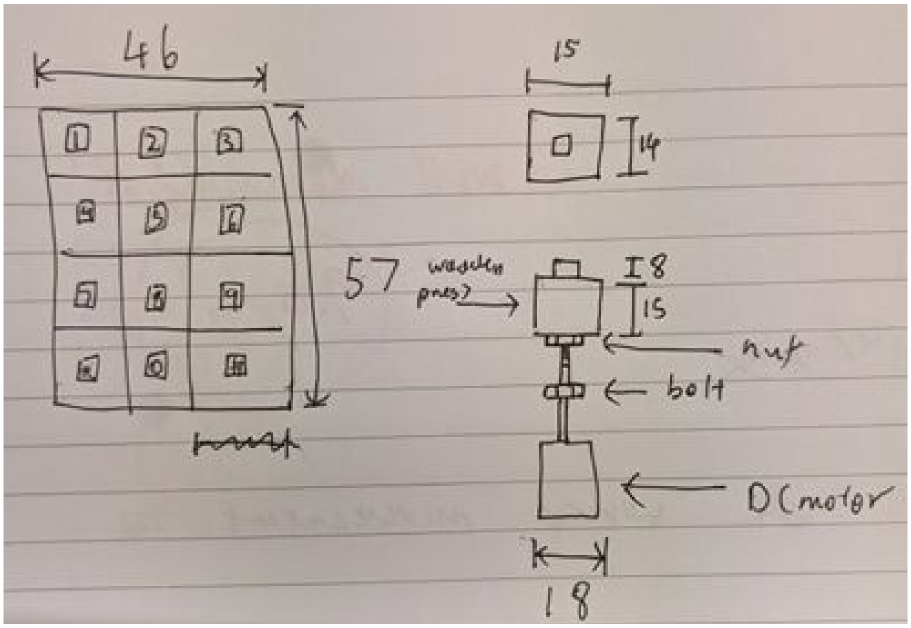


Figure 10 - Dimensions of Keypad Input System

From figure 10, the keypad pressing mechanism involves 10 wooden blocks operated by 10 DC motors that extend outwards to press the 10 keys on the keypad. A wooden press is glued to a steel nut which is then screwed into a bolt. The bolt is glued to the motor axle such that as the motor rotates, the bolt also rotates. If the wooden press is locked and unable to rotate as the DC motor runs, the nut and wooden press system is able to move up and down allowing the system to press a keypad button.

The dimensions of the keypad system is 46mm by 57mm with each key being equally spaced out. To be able to accurately press all ten buttons, each of the 10 wooden blocks need to be dimensionally accurate to satisfy the spacing of the 10 keys. The dimensions of the wooden press is 15mm by 14mm, with the pressing block having a height of 8mm and the wooden press with a total height of 23mm. The wooden presses are tessellated against each other such that they are unable to rotate and are only able to extend into and out of the housing.

The small low torque motors used to run the blocks are 18mm in diameter (greater than the dimension of the wooden press and greater than the spacing between blocks). This meant that the position of the axle, bolt and nut with respect to the base of the wooden press was not consistent across all 10 blocks (blocks situated near the edges such as 1, 2, 3 and 0 didn't have the motor axle centred on the block). This was accounted for when drilling in holes into the wooden press to allow for perfect tessellation.

4.3.4 Keypad Pressing Operation

The operation of the keypad pressing mechanism depends on the nature of the motors rotation. As the motors are powered off the Arduino MEGA directly, it's output current is naturally low and hence the motor is run at an appropriately low speed. The stock movement (same for all 10 motors) involves rotating the motor counterclockwise for 1 seconds, stopping rotation for 2 seconds and then rotating clockwise for 1 second. This stock movement allows the wooden press to extend out, press the button and then return back to its neutral position. This stock movement is repeated for all three keypad presses and is integrated into the bluetooth reception code. The motor rotating counterclockwise (extending the block) is defined by Pin 1 having HIGH logic and Pin 2 having LOW logic. The logic for the pins are reversed for clockwise rotation (retracting the block) and double LOW logic in the pins stops the motor *(see appendix N)*.

5. Implementation of Final Design

This section of the report consists of the implementation process of Team 15's bionic hand which concerns the manufacturing process and the budget allocated to the hand. Any flaws made during the manufacturing process will be reflected in the final testing section of the report.

5.1 Manufacturing Process

Construction of the bionic hand required the handling of various woods and materials. Depending on the task to be achieved, a specific type of wood was chosen. Similarly, specific materials were chosen for other aspects of the task such as 'fixing to forearm.' Prior to beginning construction, the types of wood and materials necessary for each aspect of construction were discussed and decided upon. These have been accounted for in the budget below *(see table 2)*. The various aspects of the bionic and their associated materials were as follows:

Table 2 - Final Materials in Manufacture of Bionic Hand

Aspect of Bionic	***Material Allocated***	***Justification***
Grippers/Claws	Pine Wood	Pine is soft and malleable. The curved and complex nature of claw shape meant that the material used had to be easy to work with and pine fit this criteria. Pine is also an aesthetically pleasing material which means that there is no trade-off between structural integrity and physical appeal.
Housing of Electronics	Plywood	Ply is light, cheap and easy to manufacture with. The bionic was a heavy device due to the weight and size of the claws. To reduce the weight load, ply was used to house the components resulting in less of a strain on the users hand.
Attach to Forearm	Velcro	Velcro is light, cheap and strong. This made it ideal for the attachment of the bionic to the forearm. Velcro is also thin which allowed for easier integration with the rest of the bionic.

From the designs in appendix I, Team 15 built the claw of the bionic hand from pine wood and the chassis from plywood. The claw was made from two separate pieces of pine which were cut using a bandsaw. They were glued together using a hot glue gun to create a rectangular hole in the center of the claw. For ease of manufacturing, we decided against attempting to do this with one piece of pine. A hole was then drilled into one side of the claw to be used to house an axle providing stability to the hand. Once the construction of the movable claw was complete, the fixed claw design was drawn and another piece of pine. Once again, due to the size of the claw, the pine was seperated in two pieces which were glued together again after they had been shaped by the bandsaw. Due to the servo being directly connected to the movable claw, a compartment for the servo was made in the ply base of the arm. The fixed claw was then glued onto this base and the servo glued to the movable claw. Housing for the arm was then created from ply. This was a hollow box placed behind the claws. Four holes were drilled into the box to allow for velcro to be threaded through to support the users hand. A box was then made to house all the electronics associated with the servo and EMG. This box had multiple holes outside its back and front wall to allow for alligator clips to be attached to the electrodes which would be placed on the user. This box was then glued onto the bionic.

5.2 Budget

Table 3 displays the individual component costs required for the construction of Team 15’s bionic hand. The system has been reduced into components for the clear allocation of Team 15’s monetary resources.

Table 3 - Final Design Budget

Component	***Material***	***Estimated Cost (AUD)***
EMG Signal Detection	Wiring	$3
	Surface Electrodes	*(provided by biomedical stream)*
	INA106KP	*$16.33*
Component Budget Cost	**$19.33**	
EMG Signal Processing	Wiring	$5
	Breadboard	$15
	3x TL072	*$4.95*
Component Budget Cost	**$24.95**	
Gripping Mechanism	Wiring	$2
	Structure (wood)	$9
	Screws	$2
	Motor	$35
	Foam Pad (friction)	$1
	Friction Material/Coarse Sandpaper	$2
	6V Lantern Battery	$10
Component Budget Cost	**$61**	
BlueTooth	Wiring	$3
	Arduino	*(provided by electrical stream)*
Component Budget Cost	**$3**	
Keypad	Wiring	$1
	Structure (wood)	$1
	Screws & Nuts	$4
	AA Batteries	$4
	Motors	$27
Component Budget Cost	**$37**	
TOTAL Budget Cost	**$145.28**	

6. Final Testing

The final testing required Team 15's bionic hand to perform several tasks and undergo a qualitative marking criteria. The results of the final test are in figure 11 and appendix O.

Team	P1	P2	P3	P4	Number of items	P5	Time taken
15	5	0	10	2.5	5	5	120

P6	P7	P8	P9	Robustne	Simplicity	Aesthetic	Innovation	Total
0	15	0	8	3	4	3	3	58.5

Figure 11 - Final Testing Results

The low scores achieved in tasks 1 and 2 due to the bionic hand not being able to properly pick up the water bottle and objects may be due to earlier overheating of the servo. Prior to the competition the bionic was able to grasp and lift a full 600mL water bottle and pour the water into a cup in 20 seconds. This pre-competition testing is suspected of contributing to the issues experienced during final testing.

As mentioned earlier, the nylon webbing was not used due to budget and time constraints. It was rendered redundant after the change of our design from having two moving prongs to a static lower prong paired with a moving prong,

For the task that required 10 different objects to be picked up, our bionic arm did pretty well considering the fact that our servo was not functioning at its full potential. This prevented us from picking up certain objects like the melon, bottle and jacket, as the servo did not have enough torque. However, with smaller objects, the bionic arm was successful in delivering them into the required bucket as gripping these objects did not require large quantities of force to be applied.

The above has been summarised in Appendix O.

Table 5 - Criteria Analysis

Criteria	*Final Testing Mark*	*Reflection*
Innovation	**3/5**	Appropriate mark given as the core concept and means by which the bionic hand gripped was common. The keypad was relatively unique in design with the 10 motors actuating 10 blocks a rare design
Simplicity	**4/5**	Similar mark from the expected value, however slightly lower possibly due to the slightly complex nature of the axle set up. The expected value for this criterium was high as there is limited movements and user interaction required for the functioning of the hand
Robustness	**3/5**	Higher marks than expected were given. The lower expected mark was a result of the limited time for completely sound structural housing of the components. The main reason for a lower score than the average was the burnt out servo which meant that the the impact from dropping easily reduced its already limited ability to function
Aesthetics	**3/5**	Difficulty in finding a way to aesthetically house the components required for the bionic hands function resulted in this average mark. Components were contained and hidden from sight sufficiently, however, they were contained in bulky boxes.
Integration	**8/10**	The great majority of the subsystems and components of the task were well integrated to form the final bionic hand, with the exception of a few aspects of the hand such as the bluetooth & keypad input device. This lack of integration resulted in reduced marks, however, the marks were reduced to a lower degree than expected.

7. Recommendations

From the performance and analysis of Team 15's bionic hand in the final testing, Team 15 recommends that the mechanical design of the bionic hand be largely followed. This is especially with regards to the servo configuration, which proved to be suitably simple to assemble and construct as well as providing sufficient torque to the claw. However, the servo was prone to overheating and subsequently could be used only in short bursts. This could be remedied by using a more robust servo, or by using a lighter, yet similarly strong material (such as acrylic) in claw construction. In addition, the design did not fully satisfy the 'aesthetically pleasing' criterion. This was because each subsystem required its own space, which then had to be constructed and linked to the other systems. Through an increased use of CAD software in the design process, this could be easily rectified. Consequently, weight had become a significant unforeseen issue towards the end of the assembly, as more and more wood was required for the casing to contain all the systems.

Team 15's keypad input device failed to complete its goal of pressing three buttons. This was due to a mechanical failure in the linear actuation system which was to extend the blocks to press the appropriate keys. To remedy this, Team 15 recommends the use of a solenoid system instead, as it would be less complex, be easier to assemble and would complete the task in a more efficient manner than the linear actuation would have achieved.

While the EMG System was highly functional and efficient, a few improvements can certainly be made in this area. For instance, the system did not have a switch to turn it on and off. As a result it would drain any battery connected to it even when not in use. The Client would have to manually connect and disconnect the battery in daily use. Hence a switch would greatly increase usability.

The EMG System was made with breadboards. Team 15 recommends the use of PCBs to increase robustness, as with breadboards, the wires and connectors were highly susceptible to falling out and tangling each other. The Team also recommends the increased use of soldering for further robustness.

8. Conclusion

The final design for Team 15's bionic arm involved a two pronged gripper system alongside a keypad input device comprised of 10 extending blocks. The gripping utilised a static claw and a moving claw together with a servo motor to grasp object. This gripping system was activated via EMG with all associated electronics housed inside the bionic.

The design failed to address the problem significantly. It was unable to efficiently pick up the water bottle, failed to press the keypad buttons and was not able to pick up all the objects on the table. However, it was able to be constructed under-budget with easily accessible materials and had a sound design overall, with its faults and failures lying primarily in the implementation. Despite this, Team 15 believes that with a few minor tweaks, our Bionic Hand could become a very efficient and functional device.

The outcome of this project was a satisfactory bionic arm device which simulated the function of the human hand. Team 15 has learnt to work as a cohesive, well organised group of individuals.

9. References

iWrite (2018). *Design Report Overview.* [online] Available at: http://iwrite.unsw.edu.au/iwrite/ENGINEERING/Reports/Design-Reports/Writing-Engineering-Design-Reports.html [Accessed 22/04/18]

Unknown (2018). *Monopolar vs. Bipolar EMG Readings.* [online] Available at: https://www.nrsign.com/monopolar-vs-bipolar-emg-readings/ [Accessed 25/04/18]

Gundanium. (2011). *DIY Muscle Sensor/ EMG Circuit for a Microcontroller.* [online] Available at: https://gypsyware.files.wordpress.com/2013/03/muscle-emg-sensor-for-a-microcontroller.pdf [Accessed 26/04/18]

Rory (2008). *Karaoke Circuit.* [online] Available at: https://electronics.voltsandbits.com/projects/karaoke-circuit/ [Accessed 27/4/18]

Surtell, T. (Modified 2007). *The Rectifier.* [online] Available at: https://www.eleinmec.com/printerpage.asp?18 [Accessed 29/04/18]

Unknown (updated 2017). *Arduino Reference.* [online] Available at: https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/ [Accessed 19/05/18]

Nedelkovski, D. (2016). *Arduino and HC-05 Bluetooth Module Tutorial.* [online] Available at: https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-05-bluetooth-module-tutorial/ [Accessed 18/04/18]

Servo Motor - Standard 6 Volt with Metal Gear. [online] Available at: https://www.jaycar.com.au/servo-motor-standard-6-volt-with-metal-gear-13kg/p/YM2763 [Accessed 20/05/18]

10. Appendix

Appendix A - Problem Statement:

The loss of a hand can be debilitating to an individual. Design and construct a gripper that is able to pour water into a cup and allow the user to drink, pick up objects, and press keys autonomously via bluetooth. It must be affixed to the forearm, activated by electromyography, cost no more than $150, and any electrical components are battery-powered. Must be aesthetic, robust, simple and innovative. Design and construction must be complete by Monday of Week 13 for testing.

Appendix B - Bionic Hand Material Decision Matrix

Table 1 shows Team 15's decision matrix for the material of the bionic hand. The decision matrix has a point ranking system where points are accumulated from each criteria to give an indication of the desirability of a material. Criterias which Team 15 placed more emphasis on have a scaled point system where the first ranked material receives a bonus point.

Table 1 - Bionic Hand Material Decision Matrix

Material	Robustness	Cost	Aesthetic	Light Weight	Machinability	Total
Wood	*	****	**	**	****	13
Metal	***	**	*	*	**	9
Acrylic	**	*	***	***	*	10

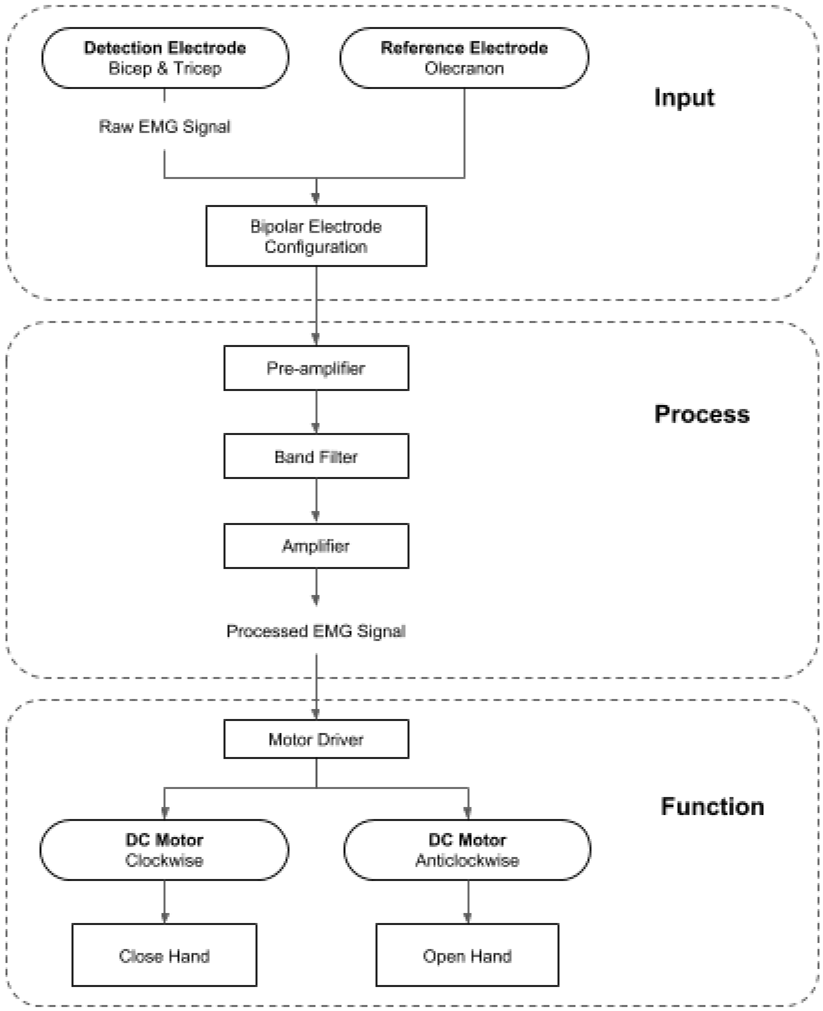
The emphasised criterias that Team 15 have chosen was cost and machinability. This was because they were budget and time constraints respectively and were identified as potential risks within the project implementation.

Appendix C - Observations of Acceptance Test

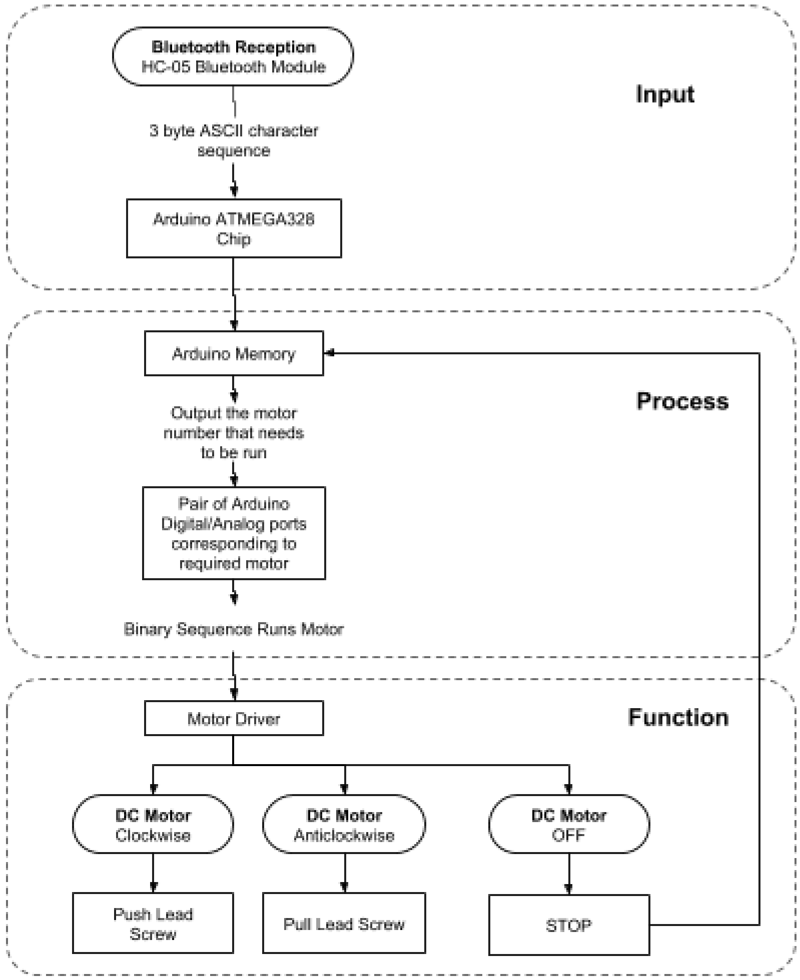
Advantages of Prototype Design
Right dimensions
Correct Bluetooth function

Disadvantages of Prototype Design	***Possible Improvements***
Slow pick up time of 50 cent coin due to size of object	Make bottom claw longer so top claw can slide coin into webbing
Not enough grip force to pick up heavy Rockmelon	Use a servo to generate the greater torque necessary for lifting/gripping of heavy objects
Rotation not smooth, claws were unstable	Use an axle to allow rotation about a point
Gears wore out with use	Use different gear material or mechanism
Use of aluminium foil as webbing	Use of a more durable webbing material

Appendix D - Design Proposal System Block Charts



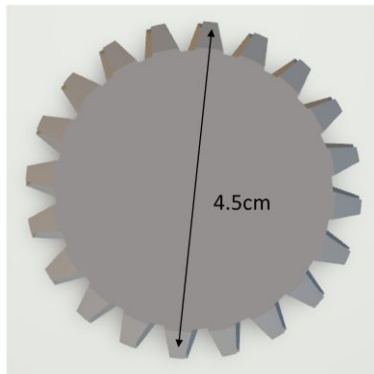
Gripping System Block Chart



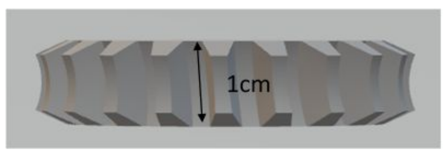
Keypad Input System Block Chart

Appendix E - Prototype Design

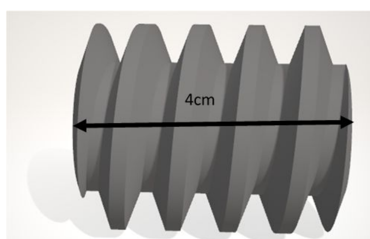
<u>Gears:</u>



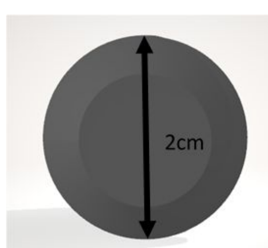
Top View of Worm Gear



Side View of Worm Gear

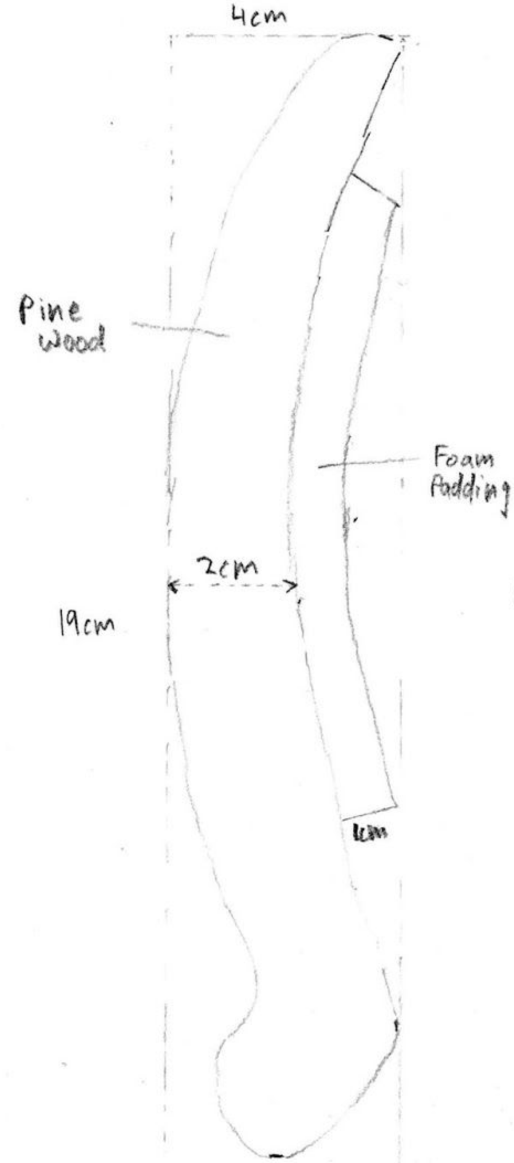


Top View of Worm Screw

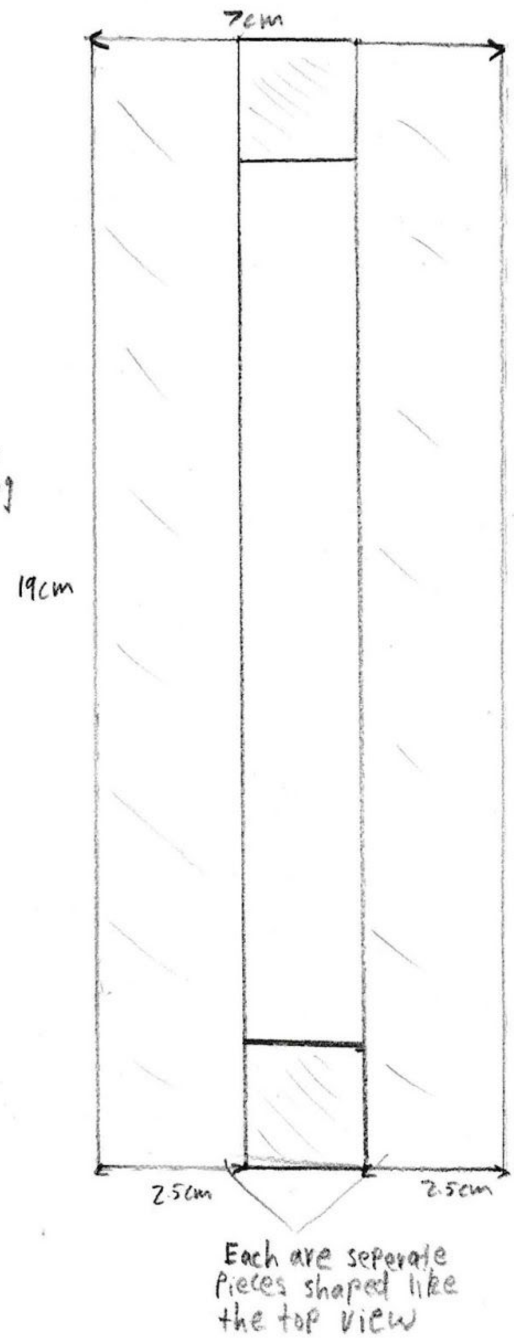


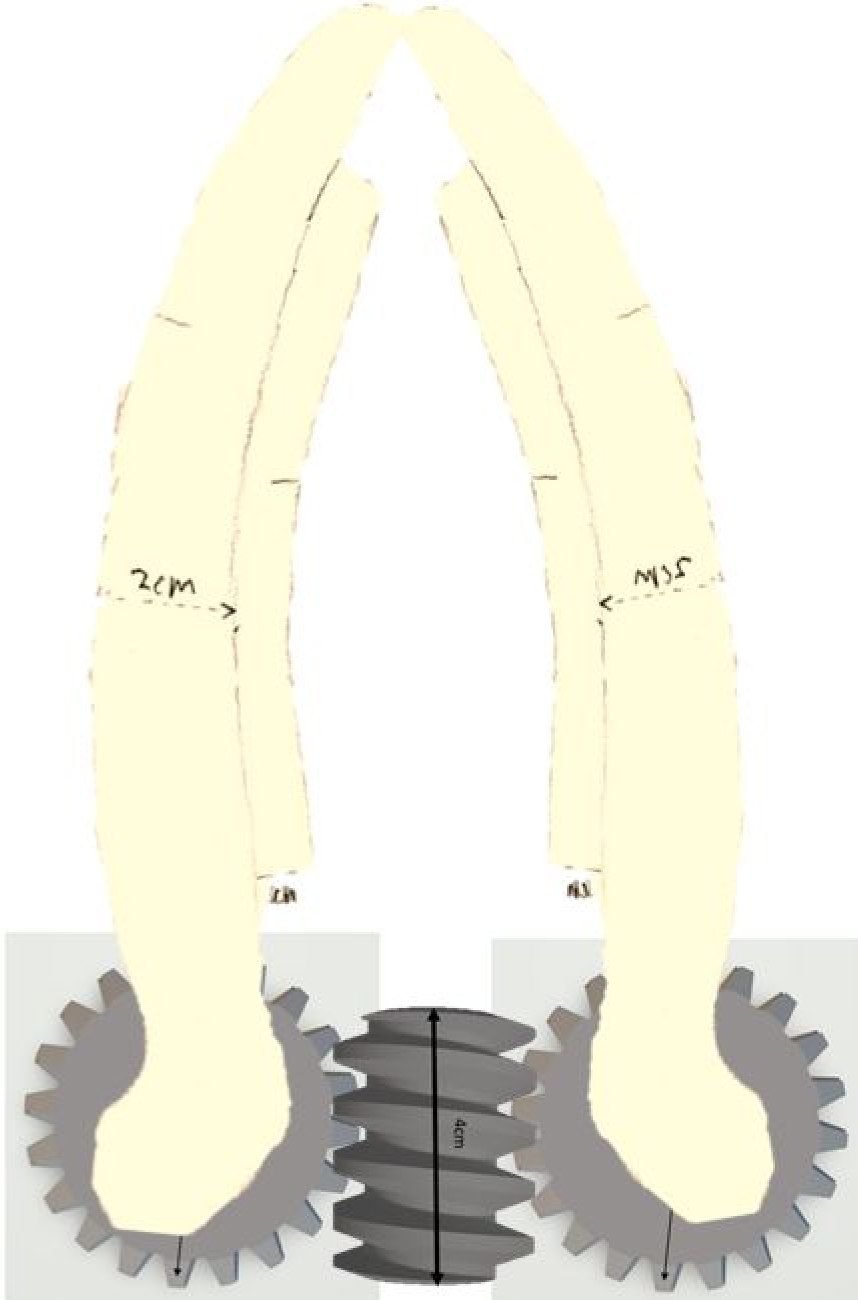
Side View of Worm Screw

Topview of claw



side view of claw

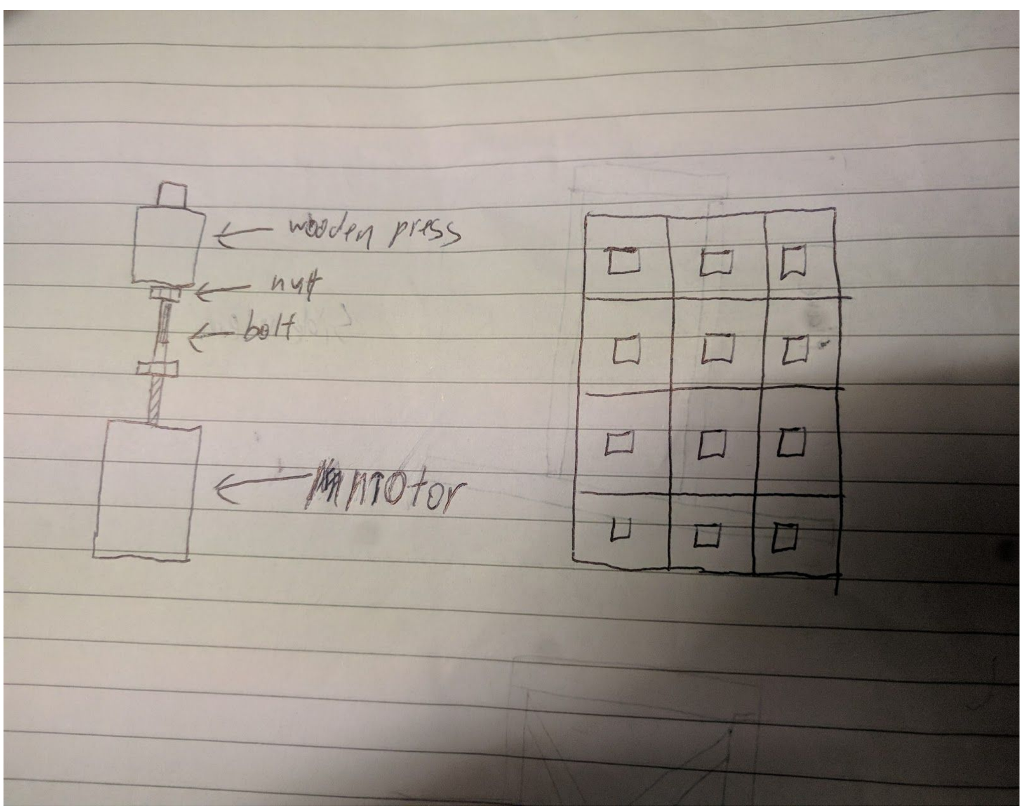




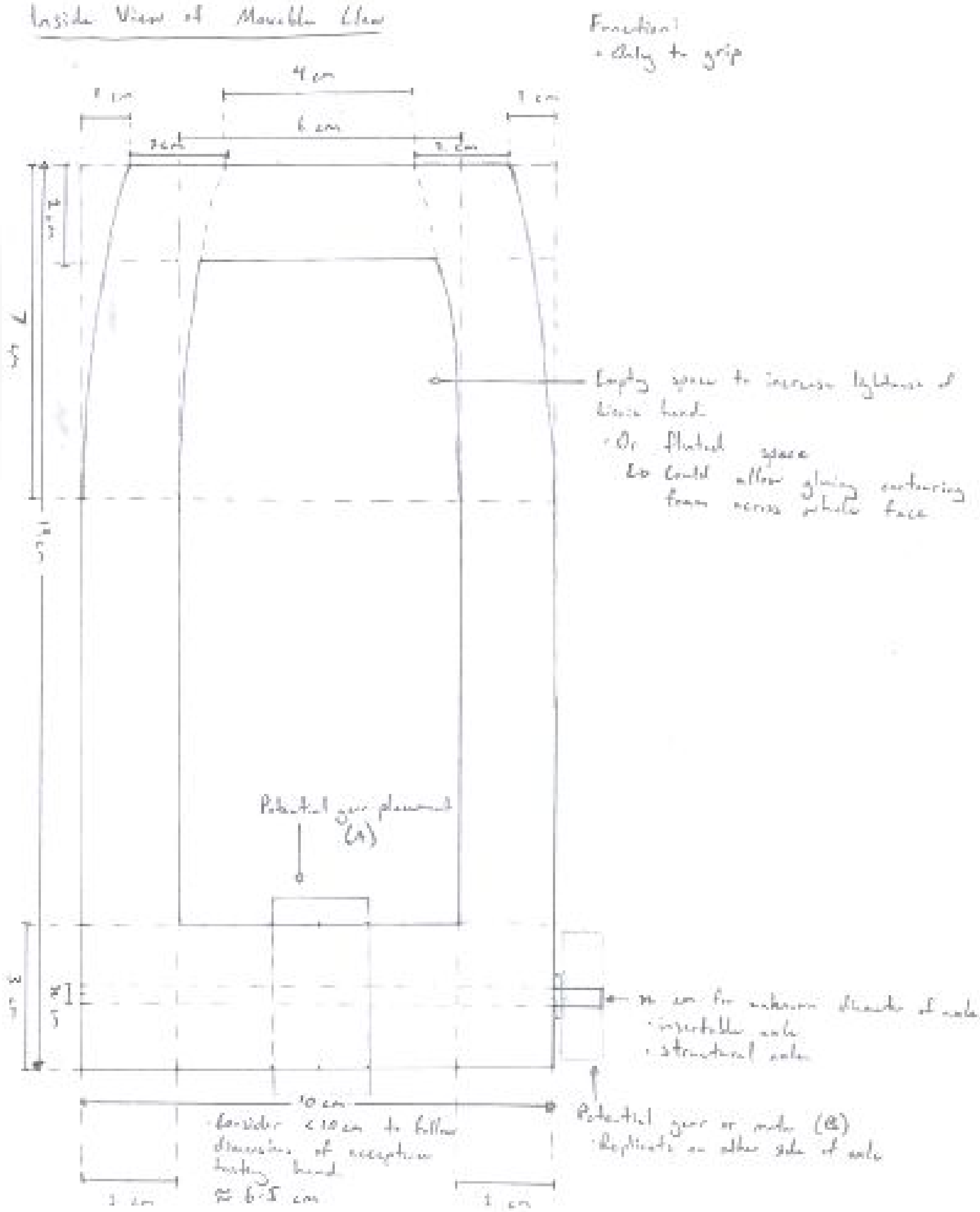
Top View of Prototype

Side View of Prototype

Appendix F - Design Proposal Keypad Input

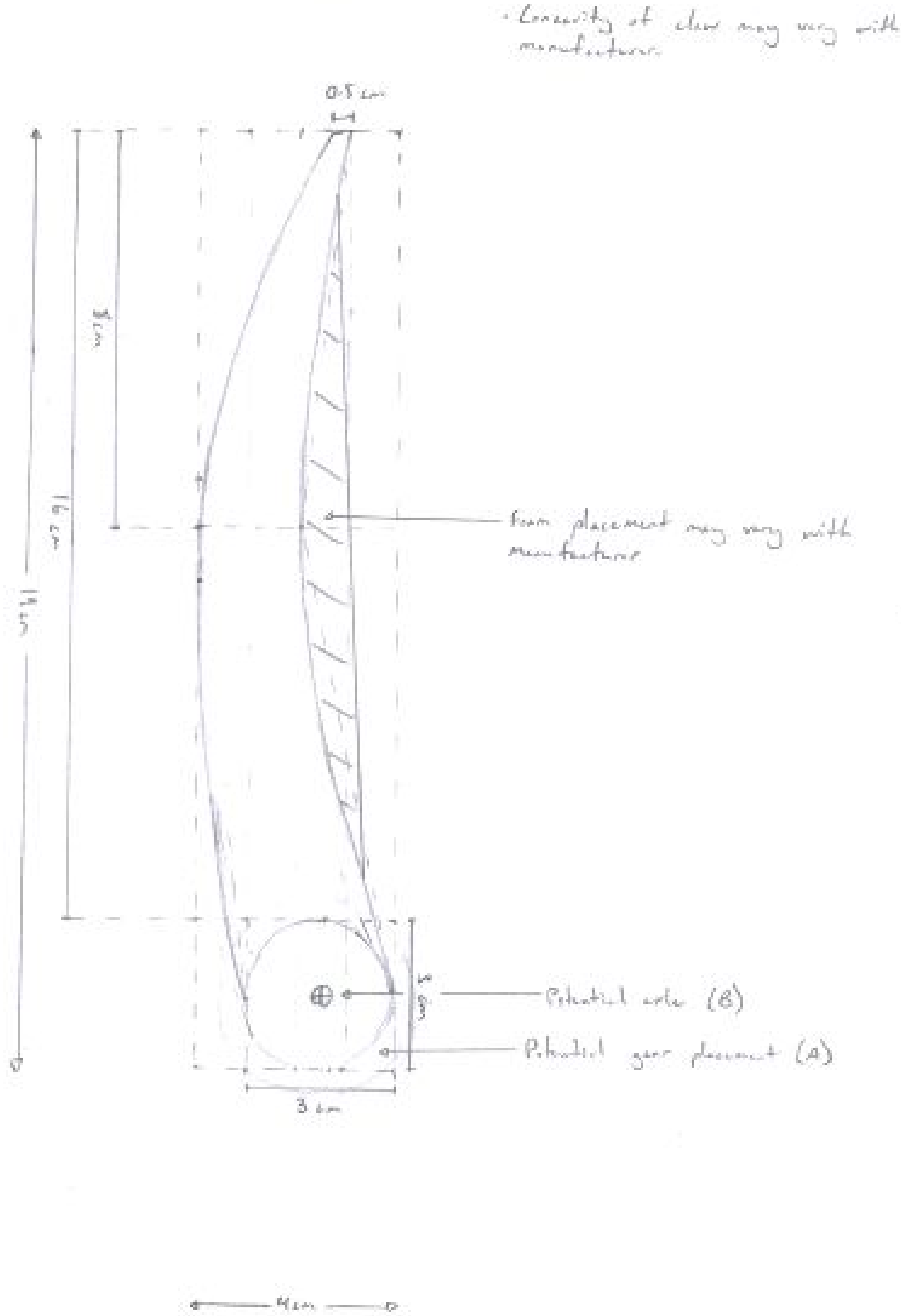


Appendix H - Movable Claw Diagrams

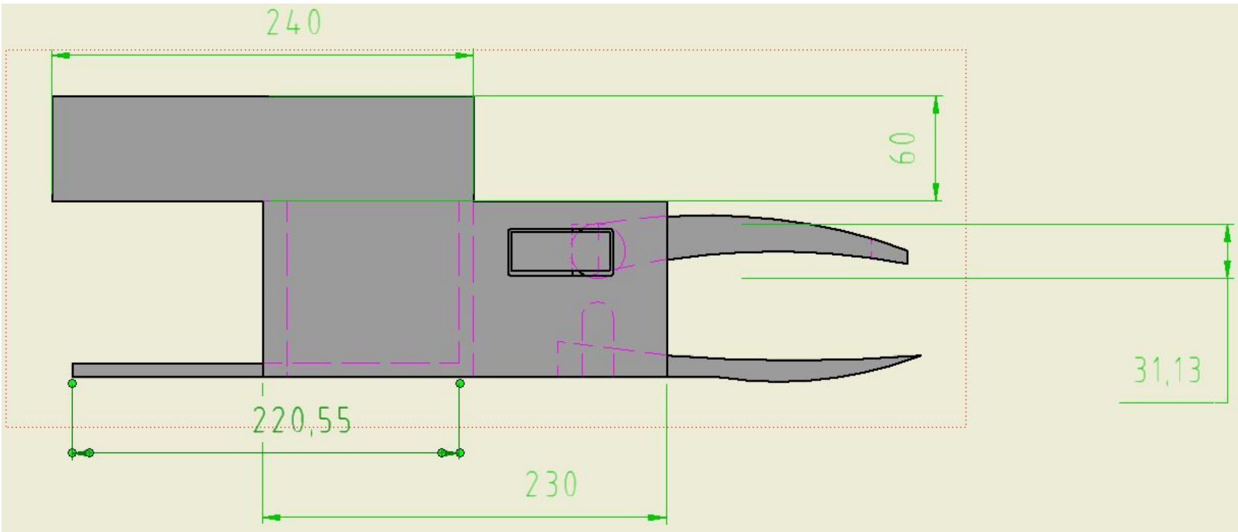


• Smooth edges to increase lightness and aesthetics. (sanding / filing)

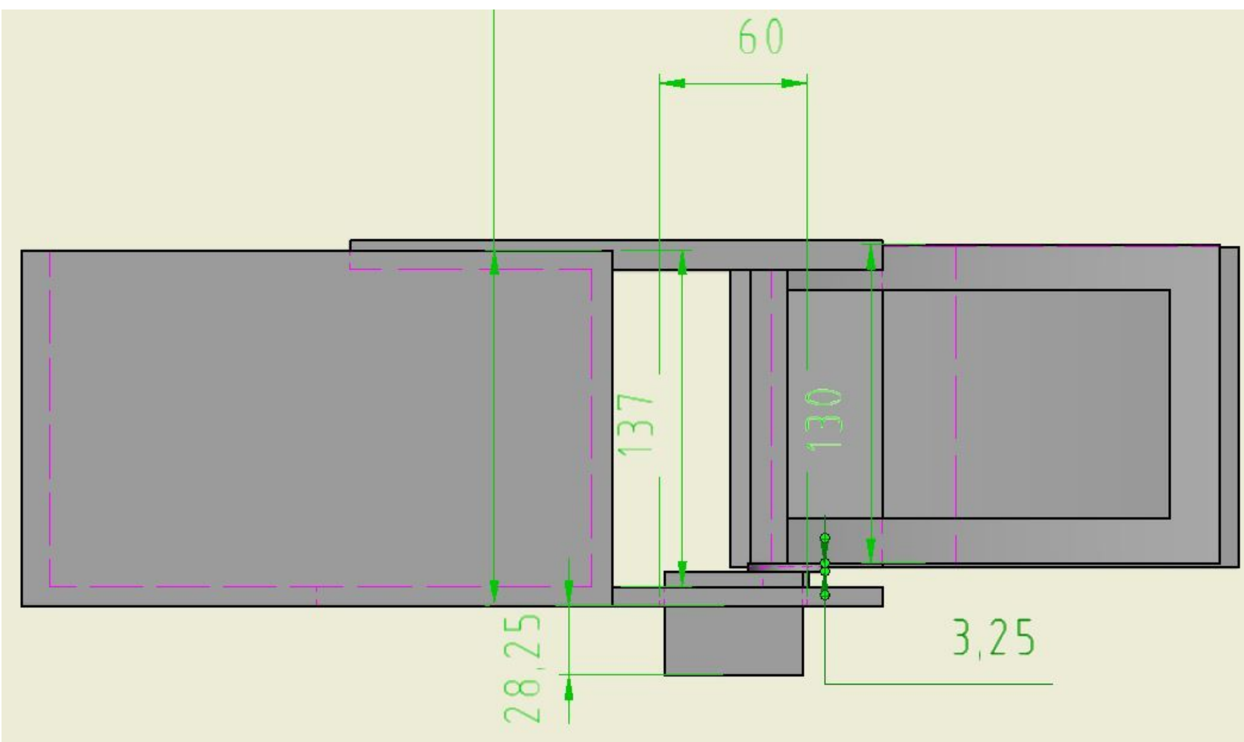
Side View of Movable Claw



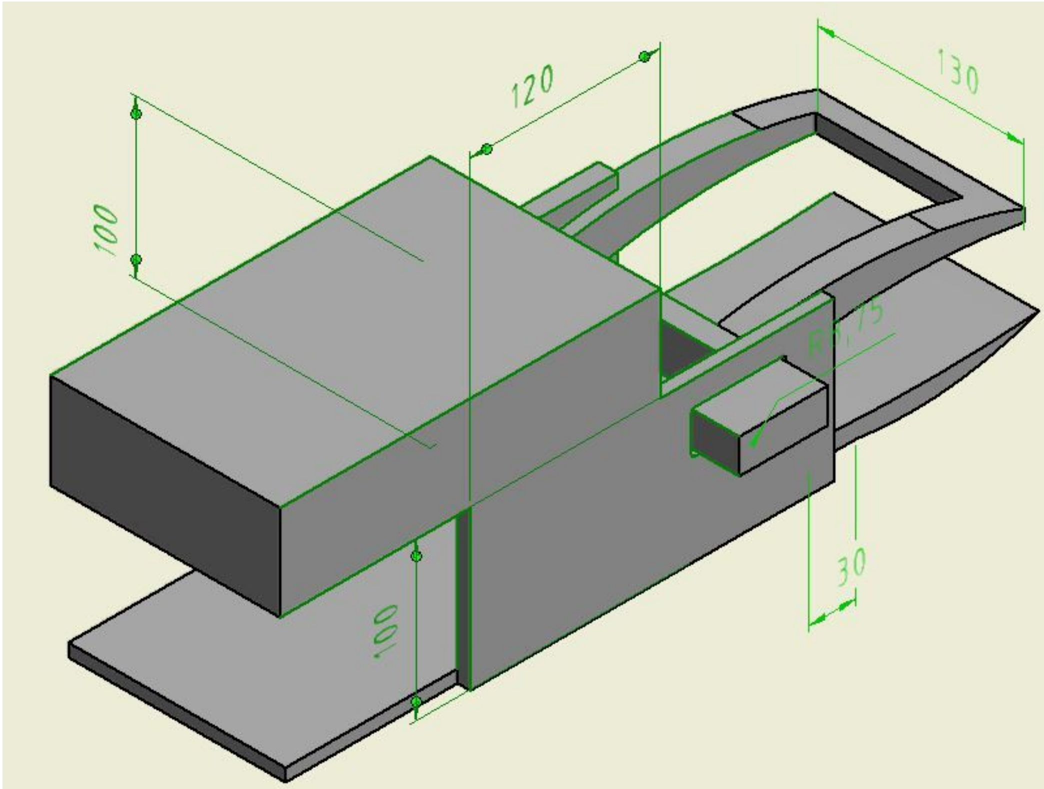
Appendix I - Dimensions of Final Bionic Hand



Side View of Bionic Hand



Top View of Bionic Hand



Appendix J - Observation of EMG Signal from Various Muscles

EMG Signal Output

Not tensing: 30mV

Tensing: 600 - 1000mV

Target Muscle	***Average EMG Signal Output (mV)***
Bicep	0.84
Lateral Deltoid	0.91
Trapezius	0.87

From experimentation of different muscles, the EMG output was consistent in amplitude. Therefore the target muscle should be chosen for the comfort of the operator of the bionic hand. The placement of the reference electrode did not matter as long as it was placed on an electrically inert surface of the skin (the acromion or olecranon is recommended).

Appendix K - Basic Code for Reading and Debugging EMG Voltage Input

```
int state = 0; //integer to store the value analogRead() returns
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); //default communication rate of serial communication with the computer
}
// put your main code here, to run repeatedly:
void loop() {
  state = analogRead(A0); //record what the analog A0 pin reads
  float voltage = state*(5.0/1024.0); //calculate the voltage from what analogRead() returns
  Serial.println(voltage); //print the voltage into the Serial monitor
}
```

Appendix L - Code for Operating Servo using EMG Signal as Switch

The rotation of the servo is limited within the range of 90-180 degrees and thus if the servo ever writes above 180 or below 90 degrees, it is reset back to the respective boundary (90 or 180).

```
#define closehand 180 //set the boundary angle for closed hand
#define openhand 90 //set the boundary angle for open hand
#define servoPin 10 //Set the servo pin as pin 10
#define servoDelay 40 //custom set delay by the handler
#define threshold 0.6 //threshold voltage of 0.6V
#include <Servo.h> //library for servo control functions
Servo myservo; //initiate the servo variable called myservo
int pos = openhand; //starting position at 90 degrees
int emg1 = 0; //initiate integers to store the EMG signal read by the analog pins
int emg2 = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); //begin the Serial communication with the computer
  myservo.attach(servoPin); //output servo commands to the pin that the servo is connected to
  myservo.write(pos); //start the hand at opened hand
  delay(2000); //allow time for the servo to reach that position
}

void loop() {
  //reset the position to the boundary if it ever goes past it
  if (pos<openhand){
    pos = openhand;
  }
  else if (pos>closehand){
    pos = closehand;
  }
```

```
 //read the two emg signals and calculate their respective voltages
 emg1 = analogRead(A0);
 emg2 = analogRead(A1);
 float voltage1 = emg1*(5.0/1024.0);
 float voltage2 = emg2*(5.0/1024.0);
 Serial.print(voltage1); //print the voltage of one of the EMG (used for debugging)
  //if the first emg is above the threshold voltage, increment up the position the servo writes (close the hand) and
delay the code to allow the servo to reach that position
 if (voltage1 >=threshold){
   pos++;
   myservo.write(pos);
   delay(servoDelay);
 }
   //if the second emg is above the threshold voltage, increment down the position the servo writes (open the hand)
and delay the code to allow the servo to reach that position

 else if (voltage2 >= 0.2){
   pos--;
   myservo.write(pos);
   delay(servoDelay);
 }
 //or else if there is no emg signal being received, continuously write the current position to stop and lock the servo
 else{
   myservo.write(pos);
 }
 //print the current position of the servo (used for debugging)
 Serial.print(" ");
 Serial.println(pos);
}
```

Appendix M - Code for Reading and Recording ASCII Sequence

```
#define numberOfCharactersSent 3 //each sequence has 3 characters
int state = 0; //initiate an integer to store the ascii character sent
int numbers[numberOfCharactersSent]; //initiate an array to store all the ascii characters
int arrayNumber = 0; //initiate a number that indicates what number the array is up to
int printed = 0; //integer that stores which sequence the code is up to
void setup() {
  Serial.begin(9600); // Default communication rate of the Bluetooth module
}
void loop() {
 // Checks whether data is coming from the serial port
 if(Serial.available() > 0){
   state = Serial.read(); // Reads the data from the serial port
   //only store the number if there are less than 3 numbers in the array and the character is a number
```

```
    if(arrayNumber < numberOfCharactersSent && state>='0' && state <='9'){
      numbers[arrayNumber] = state; //store the read number into the array
      Serial.println(numbers[arrayNumber]); //print it's ASCII code into the Serial monitor (for debugging)
      arrayNumber++; //increment the amount of numbers in the array
    }
  }
  //if the three characters haven't been printed simultaneously yet and there are three characters
recorded, print all three of them to indicate they have been recorded
  else if(printed == 0 && arrayNumber == numberOfCharactersSent){
    Serial.println(numbers[0]);
    Serial.println(numbers[1]);
    Serial.println(numbers[2]);
    //increment printed to 1 to indicate the three characters have been printed
    printed = 1;
  }
}
```

Appendix N - Code for Bluetooth Reception and Keypad Pressing

```
#define M1P1 19 //define the first pin for Motor 1 as 19
#define M1P2 18 //same for second pin
#define M2P1 15 //same for Motor 2
#define M2P2 14
…
…
#define M0P2 36 //repeated until all 20 pins are defined
#define onTime 1000 //the time the motors are on for
#define offTime 2000 //the time the motors are off for
#define numberOfCharactersSent 3 //each sequence has 3 characters
int state = 0; //initiate an integer to store the ascii character sent
int numbers[numberOfCharactersSent]; //initiate an array to store all the ascii characters
int arrayNumber = 0; //initiate a number that indicates what number the array is up to
int printed = 0; //integer that stores which sequence the code is up to
void setup() {
  Serial.begin(9600); // Default communication rate of the Bluetooth module
  pinMode(M1P1, OUTPUT); //establish all the digital pins that the motor controller is connected to as output pins
  …
  …
  pinMode(M0P2, OUTPUT);
}
void loop() {
  // Checks whether data is coming from the serial port
  if(Serial.available() > 0){
    state = Serial.read(); // Reads the data from the serial port
    //only store the number if there are less than 3 numbers in the array and the character is a number
    if(arrayNumber < numberOfCharactersSent && state>='0' && state <='9'){
      numbers[arrayNumber] = state; //store the read number into the array
      Serial.println(numbers[arrayNumber]); //print it's ASCII code into the Serial monitor (for debugging)
      arrayNumber++; //increment the amount of numbers in the array
    }
  }
  //if the three characters haven't been printed simultaneously yet and there are three characters
recorded, print all three of them to indicate they have been recorded
  else if(printed == 0 && arrayNumber == numberOfCharactersSent){
    Serial.println(numbers[0]);
```

```
    Serial.println(numbers[1]);
    Serial.println(numbers[2]);
    //increment printed to 1 to indicate the three characters have been printed
    printed = 1;
  }
  //if all three characters has been recorded, begin running the keypad pressing sequence
  If (printed ==1){
    //reset the array counter to reread the array
    arrayNumber = 0
    //keep pressing until three buttons have been pressed
    While (arrayNumber < numberOfCharactersSent){
      //record the ASCII code of the number you want to press
      state = numbers[arrayNumber];
      //if the ASCII code of the number is 49 (ie the number 1) then run the stock code for the first motor
      If (state == 49){
        digitalWrite(M1P1, HIGH);
        digitalWrite(M1P2, LOW);
        delay(onTime);
        digitalWrite(M1P1, LOW);
        digitalWrite(M1P2, LOW);
        delay(offTime);
        digitalWrite(M1P1, LOW);
        digitalWrite(M1P2, HIGH);
        delay(onTime);
        digitalWrite(M1P1, LOW);
        digitalWrite(M1P2, LOW);
      }
      …
      …
      //same for all motors up to motor 0 (ASCII code 48)
      Else if(state == 48){
        digitalWrite(M0P1, HIGH);
        digitalWrite(M0P2, LOW);
        delay(onTime);
        digitalWrite(M0P1, LOW);
        digitalWrite(M0P2, LOW);
        delay(offTime);
        digitalWrite(M0P1, LOW);
        digitalWrite(M0P2, HIGH);
        delay(onTime);
        digitalWrite(M0P1, LOW);
        digitalWrite(M0P2, LOW);
      }
      //increment the counter to read the next number
      arrayNumber++;
    }
    //once all numbers have been pressed, set printed as 2 to stop the loop of the code
    Printed = 2;
  }
}
```

Appendix O - Post Final Testing Subsystem Analysis

<table>
<tr><td colspan="2">Marks</td><td>P7
15/20</td><td>P8
0/10</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td colspan="7">Bluetooth Receiver</td></tr>
<tr><td colspan="2">Positive</td><td colspan="5">Signal received in manageable form
Signal processed and instruction sent to Keypad Input Device</td></tr>
<tr><td colspan="2">Negative</td><td colspan="5">Required reset for each new code sent
Means by which to reset was a hard to reach switch</td></tr>
<tr><td rowspan="2">Reasoning For:</td><td>Positive</td><td colspan="5">Thorough research into the reception, conversion, and transmission of a bluetooth signal. Carefully considered integration with Keypad Input device</td></tr>
<tr><td>Negative</td><td colspan="5">Lack of consideration to reset mechanism and poor ergonomic considerations</td></tr>
<tr><td colspan="7">Keypad Input Device</td></tr>
<tr><td colspan="2">Positive</td><td colspan="5">Innovative Design</td></tr>
<tr><td colspan="2">Negative</td><td colspan="5">Unable to activate keypad/actuate</td></tr>
<tr><td rowspan="2">Reasoning For:</td><td>Positive</td><td colspan="5">Well structured concept generation and evaluation phases along with thorough consideration and research of concept.</td></tr>
<tr><td>Negative</td><td colspan="5">Poor time management and inability to effectively integrate the keypad inputter with the rest of the bionic.</td></tr>
</table>

Marks		P1 5/5	P2 0/10	P3 10/10	P4 2.5/10	P5 5/10
EMG Detection						
Positive		Signal detected Placement of electrodes reduced noise/impedance				
Negative		Activation means was taxing on user Repetitive movement/uncomfortable				
Reasoning For:	***Positive***	Greatly considered choice of placement of electrodes. Configuration of electrodes for greatest efficiency researched				
	Negative	Trade-off required for acquiring the best/clearest pre-processed signal				
EMG Signal Processing						
Related Marks		P1 5/5	P2 0/10	P3 10/10	P4 2.5/10	P5 5/10
Positive		Clear signal Large range Well integrated with EMG detection				
Negative		Electronics associated with processing required great amounts of space/ housing				
Reasoning For:	***Positive***	Continuous trial and error until signal became clear Direct connection between processing unit and detection unit				
	Negative	Trade-off for a very clear and easy to work with EMG signal. Time for refinement and simplification of circuits not accounted for				
Gripping						
Positive		Support of objects of all sizes Smoothe, slow and precise gripping Friction Material helped with grasping objects Fixed claw provided support to objects				
Negative		Unable to support the weight of objects unless very light Accuracy at the cost of speed Unreliable servo				
Reasoning For:	***Positive***	Size of greatest object to be picked up was considered in detail. Careful testing and changing of code to result in slow gripping allowing for precision. Consideration of means to make pick up more efficient (leading to friction material). Reflection on results from acceptance testing used to improve on current design				
	Negative	Lack of research into the correct and efficient use of a servo motor. Insufficient testing prior to final testing. Change to core of gripping mechanism without sufficient research into effects of this change on entire bionic.				