

Universitat de Lleida

Escola Politècnica Superior

Grau d'Enginyeria Mecànica

Projecte Final de Grau

DESIGN AND IMPLEMENTATION OF A BIONIC ARM

Autor: Arnau Capell Gràcia

Director: Albert Saiz Vela

MAIG 2015



Agraïments / Acknowledgments

Primerament m'agradaria agrair aquest projecte al meu pare, Celestí Capell, sense ell no hauria estat possible la completa realització d'aquest ambiciós projecte, ell em va donar l'empenta necessària per continuar en els moments complicats i l'ajuda adequada en els moments de dubte.

També agrair el treball realitzat pel meu tutor del projecte, Albet Saiz, per el continu seguiment en l'estructuració del treball i els passos a seguir per una correcta metodologia. A més a més d'acceptar la tutela d'un projecte força complicat que havia decidit redactar en Anglès, un afegit considerable de dificultat.

Especial menció al Marc Corretgé per la paciència i facilitació de la seva impressora 3D personal per la impressió de pràcticament totes les peces, tant les fallides com les bones, d'aquest prototip. Sense ell tampoc res hauria estat possible.

Agrair la col·laboració de l'institut Caparrella per la facilitació de material per la elaboració de les plaques electròniques i els coneixements d'Arduino adquirits durant l'any.

Finalment agraïments a la Paula Mauro per el seu recolzament moral en els moments d'estrès o de no tenir les coses clares i donar-me la seva visió externa i personal del projecte per millora'l dia a dia. A més del seu consell per la realització d'una bona presentació oral.

A tots i totes involucrats en aquest treball per qualsevol motiu, Gràcies.



Contents

Agraïments / Acknowledgments	1
List of Figures	4
1. INTRODUCTION	7
1.1 Origins and motivation	7
1.2. Goals	8
1.2.1. Block Diagram	8
1.3. Outline	9
2. Orthopaedic hands: State-of-the-art	10
2.1. Aesthetic prostheses:	10
2.2. Pulling strings:	10
2.3. Electric prosthesis	11
2.4. Myoelectric Prosthesis	11
2.5. Mechanical prosthesis	12
2.6. Robotic hands	13
2.7. Michelangelo hand	14
3. Mechanisms	15
3.1. Bars mechanism	15
3.2. Mechanism with drive pulleys	16
3.3. Mechanism with motors in each joint	17
3.4. Servomotors	18
3.4.1. Pulse Width Modulation (PWM)	19
3.4.2. PID controller	20
3.4.3. The Servo used	21
3.4.3.1. Schematic sizes:	21
4. Development of the prototype	22
4.1. 3D Designs	22
4.1.1. The Hand	22
4.1.2. The Forehand	26
4.2. 3D Printing	29
4.2.1. The 3-Dimensional Printer	29
4.2.2. Specifications	30
4.2.3. Printing Material	31



4.3. Printing the 3D pieces	33
4.4. Assembly of the prototype.....	37
4.5. PCB design and manufacturing	44
4.5.1. The Muscle Sensor Board.....	44
4.5.1.1. The Electrodes.....	44
4.5.2. Arduino Shield	49
4.5.3. The Power Supply	52
4.5.4. Manufacturing Muscle Sensor PCB	54
4.5.5. Manufacturing of the Arduino's Shield	58
4.5.6. The Final Assembly	60
4.6. Experimental Result	62
4.6.1. Signal Tests	62
4.6.2. The Prototype performance.....	64
5. Costs and Economic Aspects	65
5.1. 3D Printer prices.....	65
5.2. Other elements	65
6. Conclusions	66
7. Annexes	67
7.1. Annex 1: Advancer Technologies Muscle Sensor.....	67
7.2. Annex 2: Arduino code.....	68
7.3. Annex 3: Gain estimation of the Muscle Sensor	72
7.4. Annex 4: Designed pieces drawings	73
8. Bibliography	76



List of Figures

Figure 1: Block diagram	8
Figure 2: Aesthetic prosthesis	10
Figure 3: 3D printed prostheses with strings	10
Figure 4: Greifer MyoHand VariPlus Speed by Ottobock.....	11
Figure 5: Myoelectric prosthesis scheme.....	11
Figure 6: Mechanical prosthesis.....	12
Figure 7: Robonaut [NASA]	13
Figure 8: Finger with helical gears and electric motors	13
Figure 9: Michelangelo hand.....	14
Figure 10: Toronto mechanism	15
Figure 11: Toronto mechanism scheme.....	15
Figure 12: Driving Pulleys System scheme	16
Figure 13: Adaptation of the finger with objects	16
Figure 14: Mechanism of driving pulleys with rubber return	17
Figure 15: Computer simulation of robot hand	17
Figure 16: Exploded servomotor	19
Figure 17: PWM graph of voltage with time	19
Figure 18: PID controller	20
Figure 19: Servo HS-422	21
Figure 20: Angle interval	21
Figure 21: Servo sizes side view	21
Figure 22: Servo sizes top view	21
Figure 23: Inmoov Robot.....	22
Figure 24: Inmoov right hand bottom.....	23
Figure 25: Inmoov right hand Top.....	23
Figure 26: Middle finger assembly	24
Figure 27: Thumb finger assembly	24
Figure 28: Wrist small with wrist large assembly.....	25
Figure 29: Wrist piece	26
Figure 30: Servos inside piece	26
Figure 31: Servo modelling.....	27
Figure 32: Servos virtually assembled	27
Figure 33: Servos virtually assembled and wrist piece	27
Figure 34: Holding piece of electronics	28
Figure 35: Hexagonal bar	28
Figure 36: Assembly of the forearm pieces.....	29
Figure 37: Assembly of the forearm pieces with hexagonal bar.....	29
Figure 38: Rep Rap Prusa Mendel i3	30
Figure 39: Repetier Host with wrist small and wrist large pieces.....	31
Figure 40: Pieces interpreted with Slic3r generating the G-code	32
Figure 41: Zoom of the G-Code	32
Figure 42: Printing wrist large piece.....	33
Figure 43: Failed piece, layers displaced	33



Figure 44: Inmoov Right hand pieces	34
Figure 45: Inmoov Right Hand printed	34
Figure 46: Wrist piece profile	35
Figure 47: Wrist piece top view	35
Figure 48: Servos piece just printed (18h 30 min)	35
Figure 49: Electronics piece while printing	36
Figure 50: Electronics piece printed (13h. 18min)	36
Figure 51: Drilling pieces with 3mm drill	37
Figure 52: Sanding inner parts	37
Figure 53: Sanding the piece	37
Figure 54: Pass the 2mm screw and test friction	38
Figure 55: Combination of two powerful glues	38
Figure 56: Holding the glued piece	38
Figure 57: Drilling the 8 mm holes	39
Figure 58: Passing and attaching the fishing lines	39
Figure 59: Allen screws	40
Figure 60: Female allen nuts position	40
Figure 61: Testing the fingers movement	41
Figure 62: Hand with wrist attached	41
Figure 63: Servo attach to servos piece	42
Figure 64: Fishing line attach to servo	42
Figure 65: Holding System of strings	42
Figure 66: All servos with each strings	43
Figure 67: Hand and wrist with servos piece	43
Figure 68: Assembly of all pieces	43
Figure 69: Scheme of connections between Arduino, electrodes, muscle sensor and batteries	44
Figure 70: Electrodes	44
Figure 71: Proteus scheme of muscle sensor	45
Figure 72: Layout scheme of final muscle sensor board	45
Figure 73: INA106 (Measurement)	46
Figure 74: Differential amplifier	46
Figure 75: TL072P (Amplify)	47
Figure 76: Inverting Operational Amplifier	47
Figure 77: Rectification stage	47
Figure 78: Output	48
Figure 79: Example of the board working with Sinusoidal wave	48
Figure 80: TFT screen back	49
Figure 81: TFT screen front	49
Figure 82: Shield scheme and configuration	50
Figure 83: Push-button	50
Figure 84: Servos wiring	50
Figure 85: Arduino's inner configuration	51
Figure 86: LM7805 configuration	51
Figure 87: Arduino Shield for TFT, servos and buttons	52



Figure 88: 7.4V LIPO battery	52
Figure 89: Scheme of the symmetric alimentation	53
Figure 90: Standard 9V battery	53
Figure 91: The obtained photolith	54
Figure 92: Copper plate in the insolation machine	54
Figure 93: Insolation machine	55
Figure 94: Revelation liquid.....	55
Figure 95: Board revealed	56
Figure 96: Revelator liquid	56
Figure 97: Acid product	56
Figure 98: Acid attack.....	56
Figure 99: Extractor	56
Figure 100: Copper remains	57
Figure 101: Pierce the board	57
Figure 102: Final muscle sensor board.....	57
Figure 103: Shield design from computer to special paper	58
Figure 104: After revealing the shield	58
Figure 105: Type of pins used	59
Figure 106: Using pliers to nail the pins	59
Figure 107: Welded pins top shield.....	59
Figure 108: Welded pins bottom shield	59
Figure 109: Side view of the shield with Arduino	59
Figure 110: Final shield with screen plugged in	59
Figure 111: Muscle Sensor attached to forearm.....	60
Figure 112: Arduino attached to forearm	60
Figure 113: Final Assembly.....	61
Figure 114: Prototype finally assembled.....	61
Figure 115: Signal output from the biceps.....	62
Figure 116: Output signal INA125P	63
Figure 117: Signal obtained after amplify	63
Figure 118: Signal with force.....	63
Figure 119: Signal without force	63
Figure 120: Prototype state without doing force.....	64
Figure 121: Prototype when force applied	64
Figure 122: Advancer Technologies Muscle Sensor v3	67
Figure 123: INA125P Datasheet scheme and gain equation.....	72



1. INTRODUCTION

1.1 Origins and motivation

The hands are the principal instrument for the physical manipulation of things, also being the part of the body with most of the nerve endings because the tactile information they give.

The main use of the hands is to take and hold objects, although exists a lot of derivations due to the versatility and precision of movements they have. Being used as tools to eat, write, communicate through sign language, etc.

According to the American Occupational Safety and Health Administration (OSHA), from the 2 million Americans disabled workers each year, about 400.000 have hand injuries, being the most frequent location fingers (72%).

The industrial sector shows that 60% of amputations are on hands, with the workers who work with metals the most traumatic amputations recorded (6.7%).

In our country there are few studies on the epidemiology of occupational accidents, among them there is one made in textile companies, which reports that from the total accidents in work, 54.8% are at hand and 2.4% of them are traumatic amputation.

The hand injuries have an important relevance, because they are an exceptional anatomic region with high value, due to be used in almost all the professions or occupations. Any level of amputation leads to a degree of disability that may limit the human to perform basic activities like feeding and grooming permanently.

Nowadays, exists a lot of prosthesis more or less similar to the real hands and which can achieve almost half of the functions of the real ones, the newest can be moved by mental signals, or by nerve signals. Furthermore, the price of them cannot be afford by most of the people with a basic salary.

Therefore, this project is about creating a prototype of an orthopaedic hand which can be made with a low price. Using the new technology of the 3Dprinters and the movement will be performed by muscle signs.

1.2. Goals

The aim of this project is to create a prototype of a hand which can be placed on the forearm of the body, this hand would achieve the basic movements of grasping an object in two different ways:

- Impingement, with the index, middle finger and thumb. For smaller objects and more precision.
- Enclosure, with the whole fingers. For standard objects.

The main point of the project is the performance of the movement, it will be moved by the signals of force read at the biceps with some electrodes, amplified by an electronic device self-created and interpreted by *Arduino UNO*.

Arduino will move three angular servos each one pulling some strings attached to the hand to make the fingers shrink or extend depending on the force you make with the biceps.

The design of the hand will be taken from the Internet, "Inmoov" is an open-source 3D printed life-size robot, it has been decided to take just the right hand from this project and the forearm is going to be designed with CREO Parametric in order to set the hand on the human forearm and carry the servos, electronics and alimentation on it.

1.2.1. Block Diagram

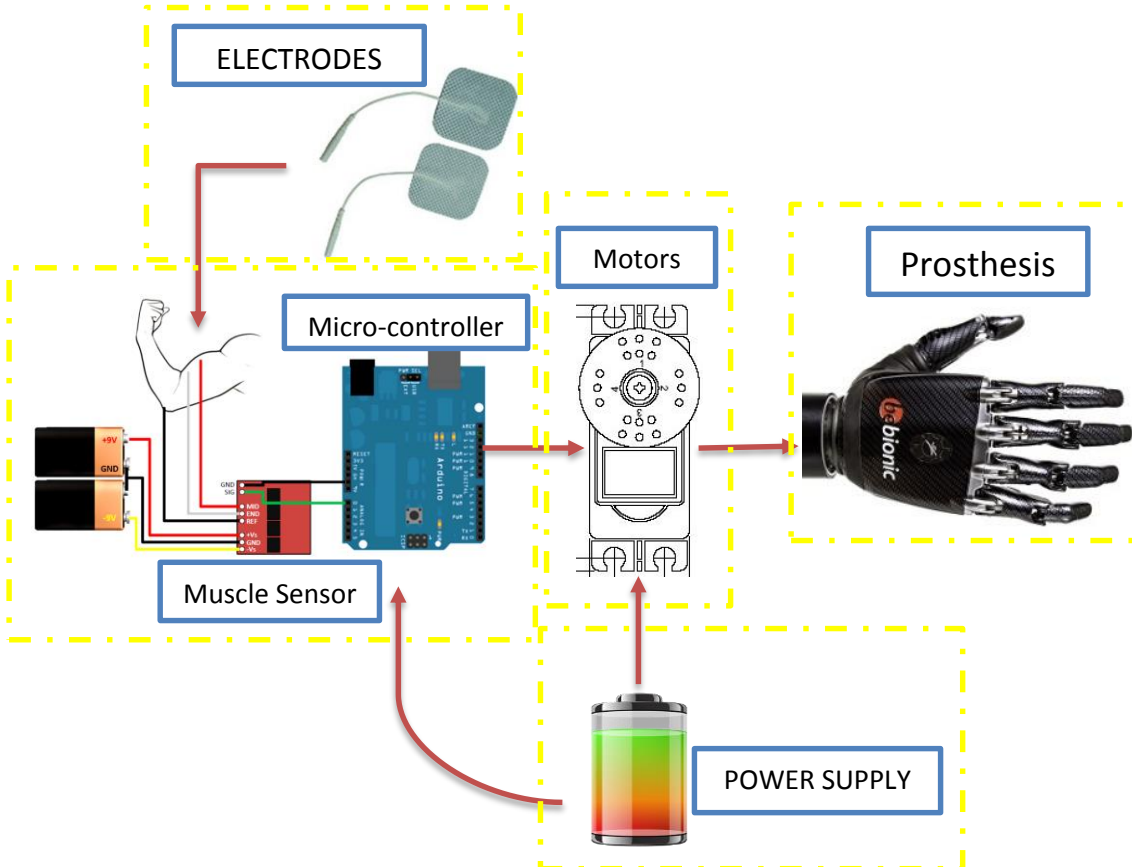


Figure 1: Block diagram



1.3. Outline

This project dissertation follows the following structure:

Chapter 2 shows a review of the state-of-art of some existing prosthesis, the way them work and a brief contrast of the advantages and drawbacks each one has.

Chapter 3 presents the most used mechanisms designs to perform the movement of the orthopaedic hands or fingers, either mechanic or robotic. Moreover, an explanation of the servomotors, its common uses and the principle of working.

Chapter 4 is focused on the development of the prototype, divided in two parts, the first one mechanical and the second one mostly electronic. It begins with the 3D designs used on this prototype, introducing the 3-Dimensional printer and the plastic based material used to print each piece. Followed by the step by step assembly of the hand and forearm. Otherwise, all the electronical devices with the interpretation of the designs used (Muscle sensor and Arduino's Shield) followed by the manufacturing of the PCB's and ending with the final assembly of the prototype and the performance of it.

On Chapter 5 are contrasted all the economic factors of the whole project manufacturing, in order to contrast them with the commercial prosthesis on market.

Therefore, Chapter 6 is to sum up the final knowledge and the experiences acquired during the project, in addition of concluding all the aspects that could be improved in the next future.

2. Orthopaedic hands: State-of-the-art

The main use of the hands is to pick up and hold objects, although of their multiple uses the most important and useful are this.

Therefore, the orthopaedic hands want to achieve this two actions in a way or another.

Now a days, exists some mechanisms to perform the movements of the hand:

2.1. Aesthetic prostheses:

This kind of prostheses are a passive solution, which are made just to emulate aesthetically the real hand. The resemblance is astonishing but as a drawback they cannot make movements.



Figure 2: Aesthetic prosthesis

2.2. Pulling strings:

In this method, the idea is just to tie a thread or a fishing line at the top of the fingers, and the same thread attached on the wrist or somewhere on the forearm. Then with the rotation of the wrist the threads are pulled back and the fingers are consequently fold.

To perform this method you must have the movement of the wrist, otherwise you cannot pull the strings. Even so, it is a good solution because is a really cheap making with the new 3Dprinters and an easy design of the pieces, also you can find them in the Internet for free use.

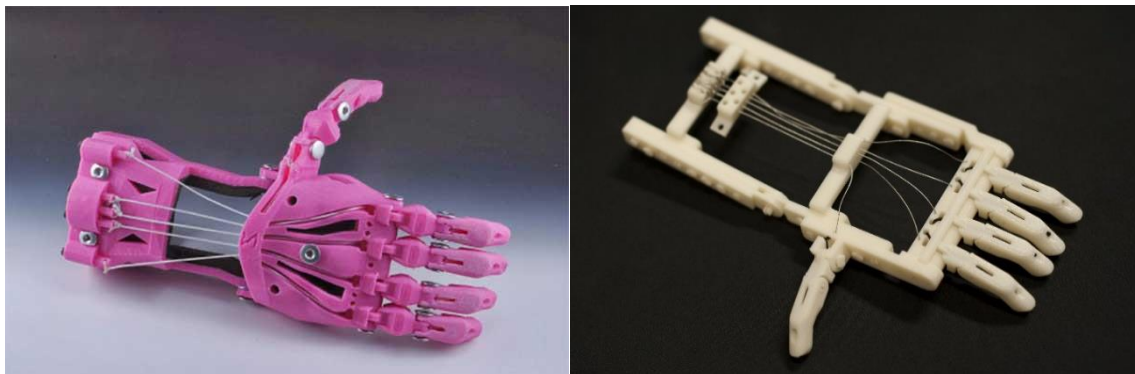


Figure 3: 3D printed prostheses with strings

2.3. Electric prosthesis

These prostheses use electric motors in the terminal device, wrist or elbow with a rechargeable battery. These prostheses are controlled in various ways, either with servo control with push button or button switch harness.

It is more expensive to purchase and repair, there are other obvious disadvantages and care to exposure to a moist environment and the weight of the prosthesis.



Figure 4: Greifer MyoHand VariPlus Speed by Ottobock

2.4. Myoelectric Prosthesis

The myoelectric prostheses are electric prosthesis controlled by a myoelectric external power, these prostheses are today the type of artificial limb with the highest degree of rehabilitation.

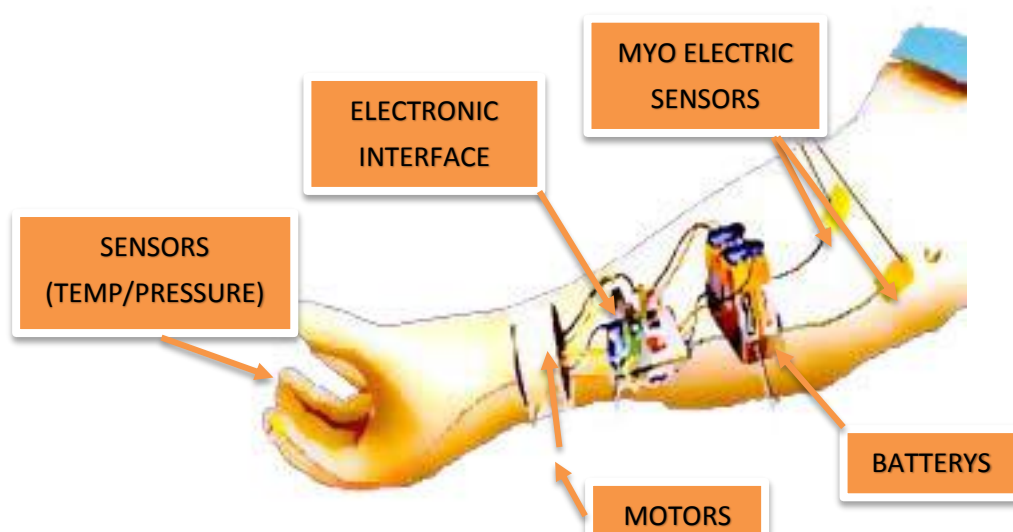


Figure 5: Myoelectric prosthesis scheme

Synthesize the best aesthetic appearance, have great grip strength and speed, as well as many possible combinations and enlargement.

The myoelectric control is probably the most popular control scheme. It is based on the concept that if a muscle in the body is contracted or flexed, a small electrical signal (EMG) that is created by the chemical interaction occurs in the body. This signal is very small (5-20 μ V).

Using sensors called electrodes in contact with the surface of the skin can record the EMG signal. Once registered, this signal is amplified and processed later by a controller that switches the engine turning them on and turning them off in your hand, wrist or elbow to produce movement and functionality.

This type of prosthesis has the advantage of only requiring a user to bend his muscles to operate. A disadvantage using a battery system that requires maintenance to recharge, discharge, discard and eventually replace it.

2.5. Mechanical prosthesis

The mechanical hands are devices that are used with the function of opening or voluntary closure by means of a harness which is fastened around the shoulders, chest and part of user-controlled arm. Its operation is based on the extension of a league through the harness for open or close, and closing or opening is made only with muscle relaxation respectively by a spring to perform a pressure force or pinch.

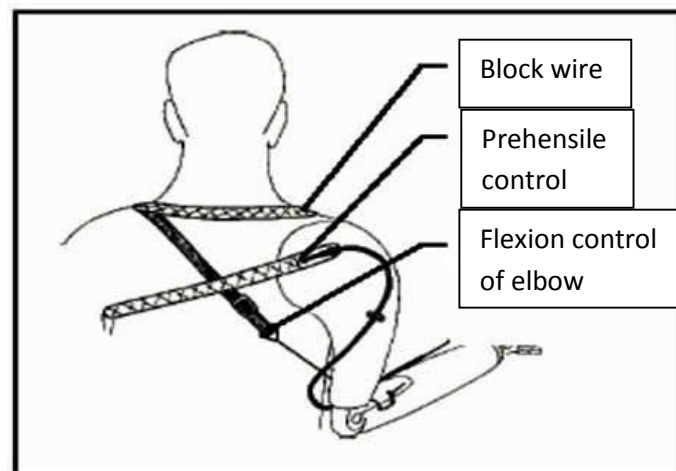


Figure 6: Mechanical prosthesis

Due to these prostheses are driven by the body, it is necessary that you have at least a general movement: chest expansion, depression and shoulder elevation.

2.6. Robotic hands

Many ground breaking dexterous robot hands have been developed over the past two decades. These devices make it possible for a robot manipulator to grasp and manipulate objects that are not designed to be robotically compatible.

Their challenge is to build machines that can help humans work and explore in space.

Also the innovations of these type of hands would be adapted in the next future for humans providing the knowledge with technology and biology increase.



Figure 7: Robonaut [NASA]

Each phalange has a little motor, with this method, you can move every single phalange the number of degrees you want. It is not an easy way to perform the movements of a real hand, also the sequences are hard to program, and the design of the different pieces is conditioned by the size of its motors and gears which could increase a lot the weight of the whole hand and consequently the price.



Figure 8: Finger with helical gears and electric motors

2.7. Michelangelo hand

Although we are in constant evolution, there are some existing myoelectric hands with a really similar movements as the robotic hands and controlled by human signals. *Michelangelo hand* is a good example of this progress. With 6 degrees of freedom Ottobock have performed this bionic hand with a really similar shape of the human hand and good benefits like water resistance and different forces and velocities to grab. Composed of steel and duralumin high strength. Which replace the structure endoskeletal of human bones, externally is covered with silicone elastomer replacing the soft structures as well as other plastics replacing the muscles and tendons.



Figure 9: Michelangelo hand

3. Mechanisms

The implementation of the movements of the articulations in the hand is based on some mechanisms to extend and fold the fingers. Basically the used ones are; transmission bars, drive pulleys, motors added in each articulation and just pulling strings.

3.1. Bars mechanism

The bars are configured in a way that they fold at the same time just moving one bar towards or forwards, with this system you don't need motors and the manufacturing is easy and cheap.

The movements are quite limited just because as it's said the flexion is produced at the same time in the three articulations.

N. Dechev et al. / Mechanism and Machine Theory 36 (2001) 1157–1173

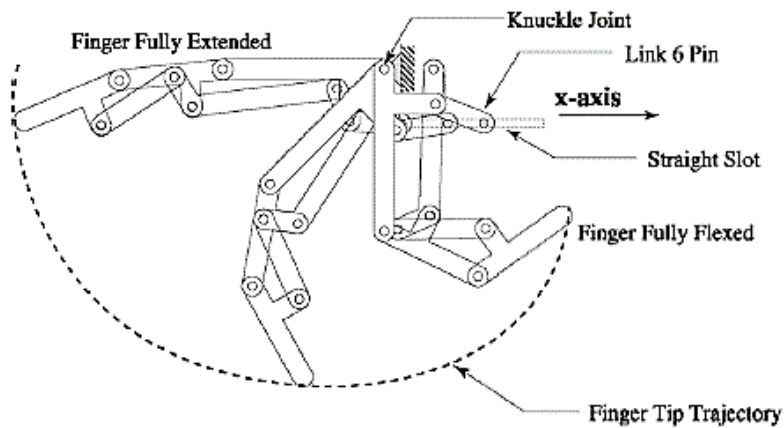


Figure 10: Toronto mechanism

Here we can see an example of this mechanism, this one is called Toronto's mechanism.

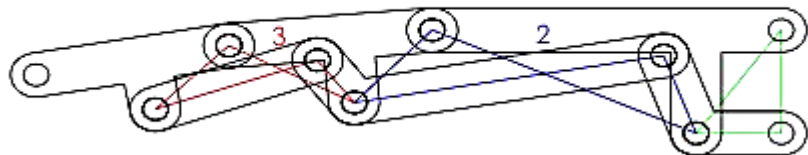


Figure 11: Toronto mechanism scheme

3.2. Mechanism with drive pulleys

The idea of this mechanism is really similar with the bars one, just because to achieve the movement you have to perform one action in this case make one pulley turn around, with a motor for example, and the configuration of the pulleys will make the others do the desired movement. This system makes the weight of the hand goes down and it's easy to implement.

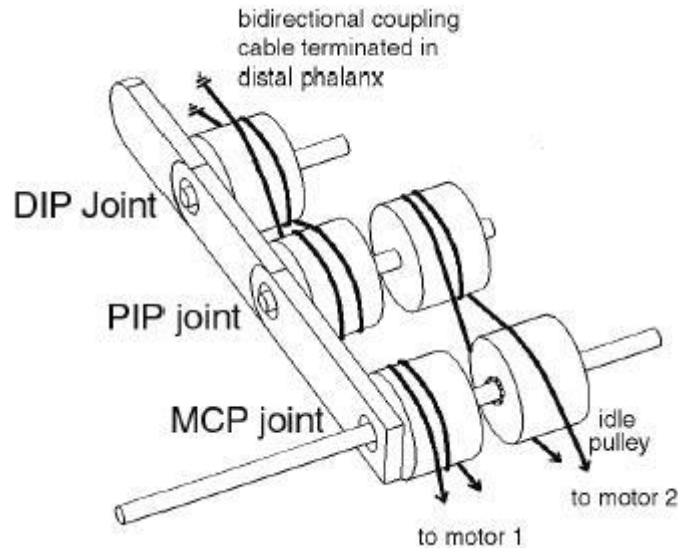


Figure 12: Driving Pulleys System scheme

In this figure we can see the way to configure this basic system to achieve movements like this.

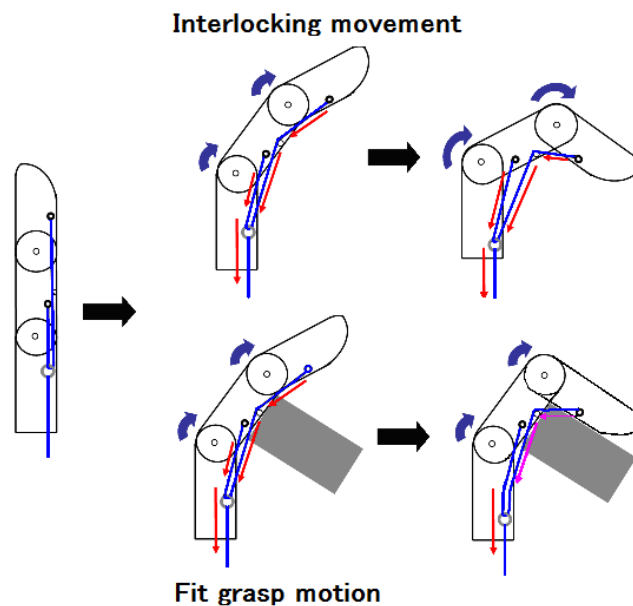


Figure 13: Adaptation of the finger with objects

If the motor is not bidirectional and it's wanted the finger to go back, a possibility is to put another motor like in figure 12 or as an easier and cheaper solution add a rubber at the top of the finger and when it folds down, the rubber will become in tension and when the motor stops the rubber will make the finger go back.

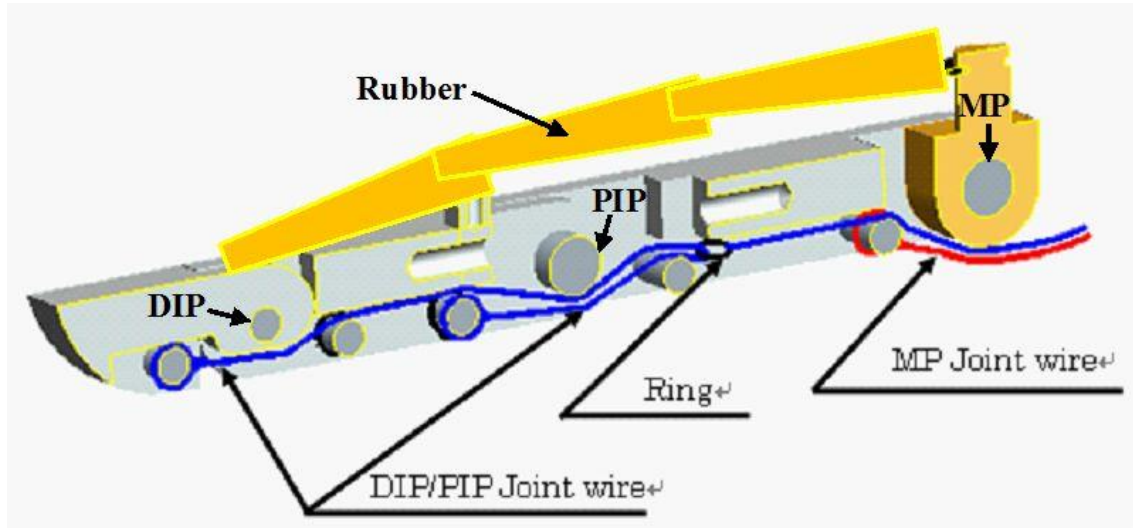


Figure 14: Mechanism of driving pulleys with rubber return

3.3. Mechanism with motors in each joint

This mechanism is used basically for robotic hands because when the movements are programmed they could be really complex and sophisticated. With this system you can move any single part of the finger independently and the number of degrees wanted. Of course the weight of the hand is strongly increased and if you think about how you can move any of this parts with muscle signals or parts of the body, it becomes so hard. There so, this type of hands usually are for robots or to be moved electronically but hardly as human prostheses.

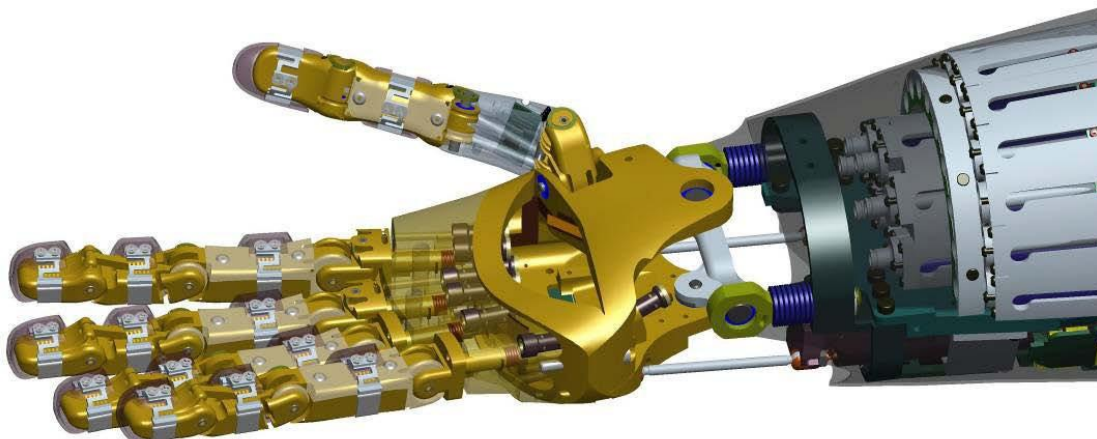


Figure 15: Computer simulation of robot hand



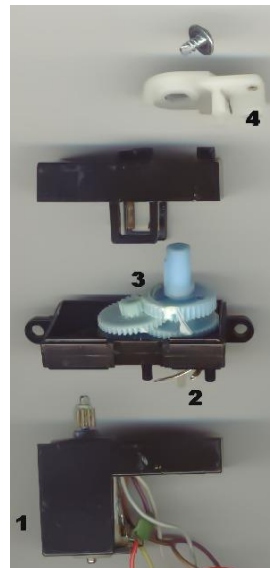
In the recent days, there have been some new researches about mind control in prostheses. If this continue evolving we would be able to move every single finger in one way or another just thinking about it.

3.4. Servomotors

A servomotor is a specific type of motor that is combined with a rotary encoder or a potentiometer to form a servomechanism. This assembly may in turn form part of another servomechanism. A potentiometer provides a simple analogical signal to indicate position, while an encoder provides position and usually speed feedback, which by the use of a **PID controller** allow more precise control of position and thus faster achievement of a stable position (for a given motor power). Potentiometers are subject to drift when the temperature changes whereas encoders are more stable and accurate.

Servomotors are used for both high-end and low-end applications. On the high end are precision industrial components that use a rotary encoder. On the low end are in expensive radio control servos (RC servos) used in radio-controlled models which use a free-running motor and a simple potentiometer position sensor with an embedded controller. The term servomotor generally refers to a high-end industrial component while the term servo is most often used to describe the inexpensive devices that employ a potentiometer. Stepper motors are not considered to be servomotors, although they too are used to construct larger servomechanisms. Stepper motors have inherent angular positioning, owing to their construction, and this is generally used in an open-loop manner without feedback. They are generally used for medium-precision applications.

RC servos are used to provide actuation for various mechanical systems such as the steering of a car, the control surfaces on a plane, or the rudder of a boat. Due to their affordability, reliability, and simplicity of control by microprocessors, they are often used in small-scale robotics applications. A standard RC receiver (or a microcontroller) sends **pulse-width modulation (PWM)** signals to the servo. The electronics inside the servo translate the width of the pulse into a position. When the servo is commanded to rotate, the motor is powered until the potentiometer reaches the value corresponding to the commanded position.



1. Electric motor
2. Position feedback potentiometer
3. Reduction gears
4. Actuator arm

Figure 16: Exploded servomotor

3.4.1. Pulse Width Modulation (PWM)

It is a type of modulation used to modify the load which arrives to the device connected in order to control its potency.

The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast rate. The longer the switch is on compared to the off periods, the higher the total power supplied to the load.

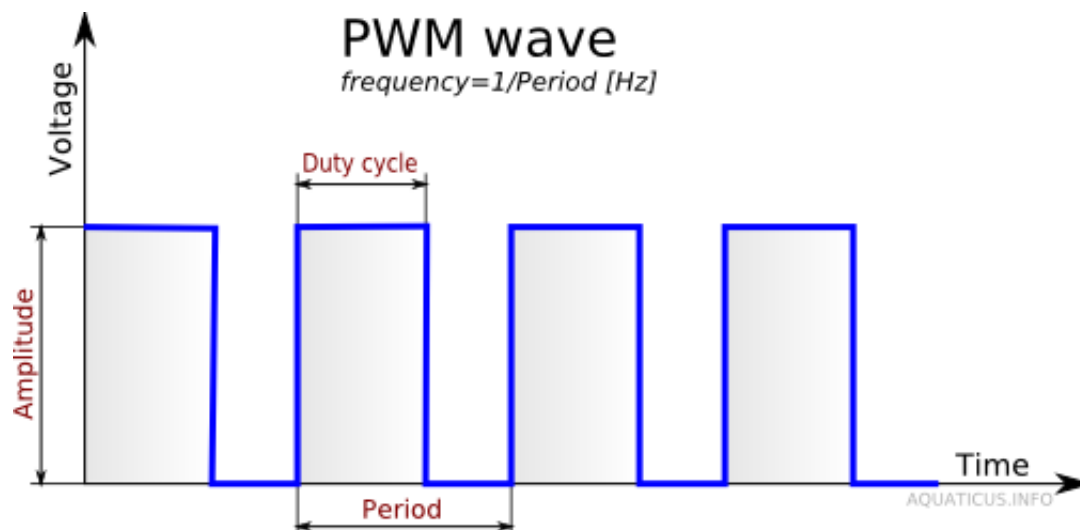


Figure 17: PWM graph of voltage with time

The term duty cycle describes the proportion of 'on' time to the regular interval or 'period' of time; a low duty cycle corresponds to low power, because the power is off for most of the time. Duty cycle is expressed in percent, 100% being fully on. When it is half time "on" and half time "off", then it supplies 50% of the power, and so on.

3.4.2. PID controller

A proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism(controller) widely used in industrial control systems. A PID controller calculates an error value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process through use of a manipulated variable.

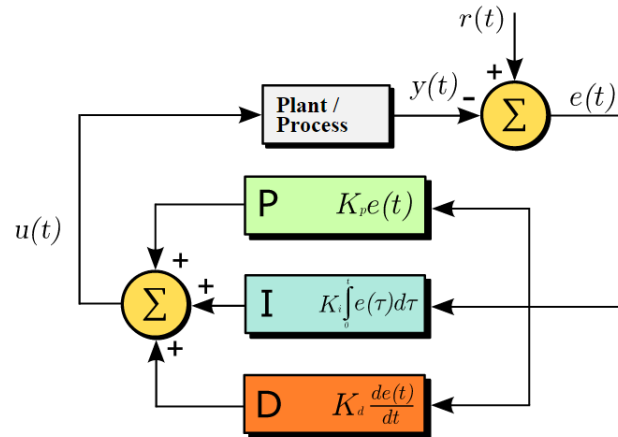


Figure 18: PID controller

The **proportional term** produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant K_p , called the proportional gain constant.

The contribution from the **integral term** is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain (K_i) and added to the controller output.

The integral term accelerates the movement of the process towards set point and eliminates the residual steady-state error that occurs with a pure proportional controller.

The **derivative** of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain K_d . The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, K_d .

Derivative action predicts system behaviour and thus improves settling time and stability of the system.

3.4.3. The Servo used

On this project, the kind of servomotor used is the one in the right figure, it can achieve positions between 0° to 180°.

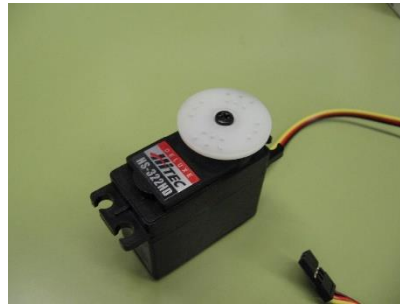


Figure 19: Servo HS-422

The model is *Hitec Hs-422*, with the following specifications in Figure 17:

Voltage	4.8 – 6.0 Volts
Torque	45.82/56.93 oz-in. (4.8/6V)
Speed	0.21/0.16 sec/60° (4.8/6.0V)
Direction	Clockwise/ Pulse Traveling 1500-1900usec
Rotation	180°
Gears	Nylon Gears

Table 1: Properties of the servo Hitec Hs-422

On the next figure is an explanation of the range it works from 0° to 180°.

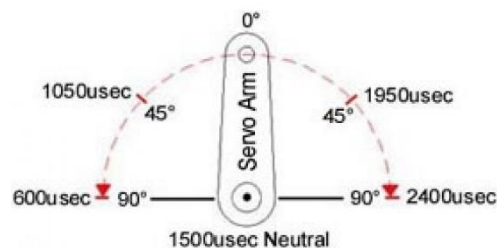


Figure 20: Angle interval

3.4.3.1. Schematic sizes:

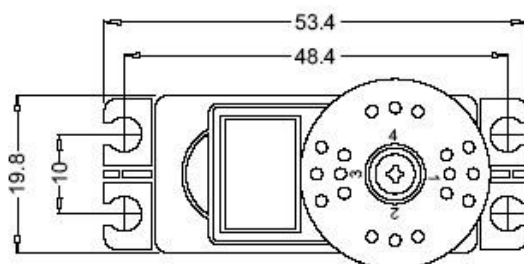


Figure 22: Servo sizes top view

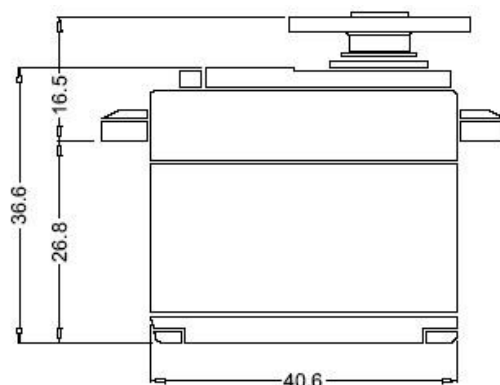


Figure 21: Servo sizes side view

4. Development of the prototype

4.1. 3D Designs

4.1.1. The Hand

InMoov is the first life size humanoid robot you can 3D print and animate.

Gael Langevin is a French model maker and sculptor. He works for the biggest brands since more than 25 years.

InMoov is his personal project, it was initiated in January 2012.

InMoov is the first Open Source 3D printed life-size robot.

Replicable on any home 3D printer with a 12x12x12cm area, it is conceived as a development platform for Universities, Laboratories, Hobbyist, but first of all for Makers.

Its concept, based on sharing and community, gives him the honour to be reproduced for countless projects throughout the world.



Figure 23: Inmoov Robot

Taking profit of this open-source project, it's been chosen the hand of its robot.

The main reasons are:

- The likeness with a real hand is more than acceptable
- The positions that can be achieved are sufficient for basic functions
- It can be printed by anyone who have a 3D printer at home
- It is easy to be moved by pulling simple fishing lines
- It is an open-source design with a Creative Commons license

Next figures show the 3D modelling of each parts:

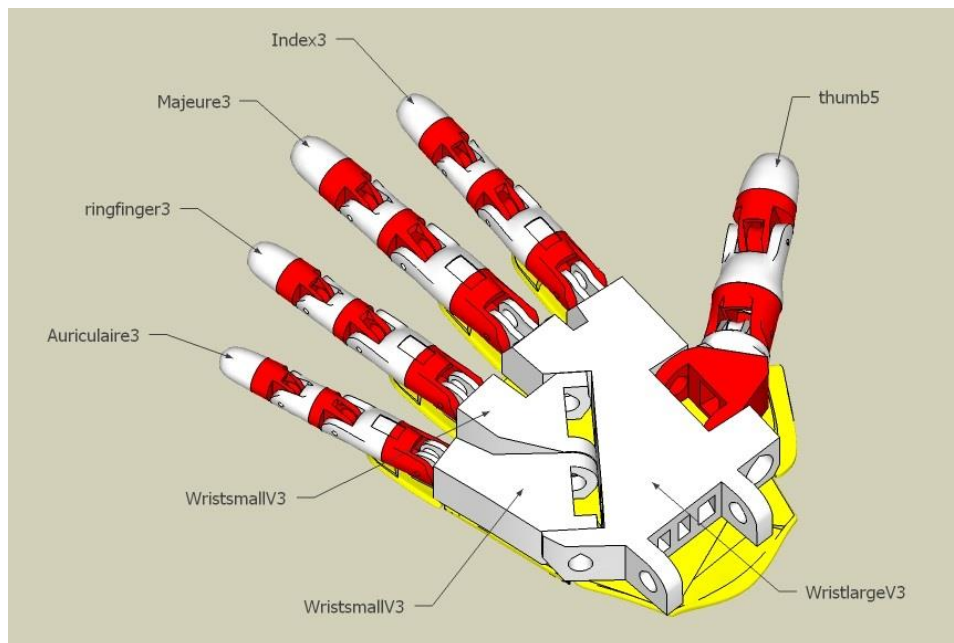


Figure 24: Inmoov right hand bottom

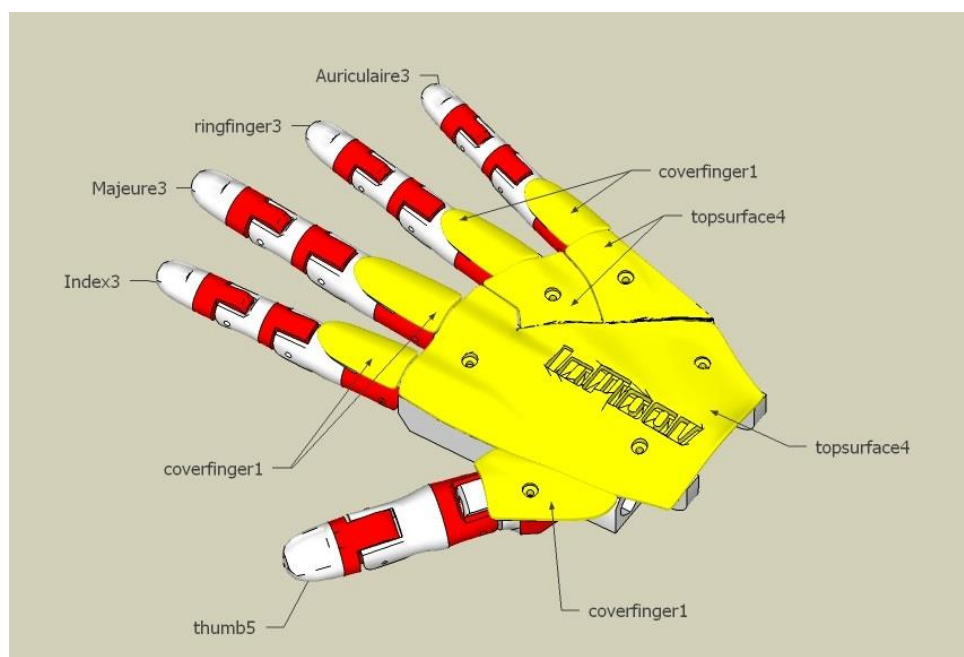


Figure 25: Inmoov right hand Top

Some graphic explanations about how it has to be assembled:

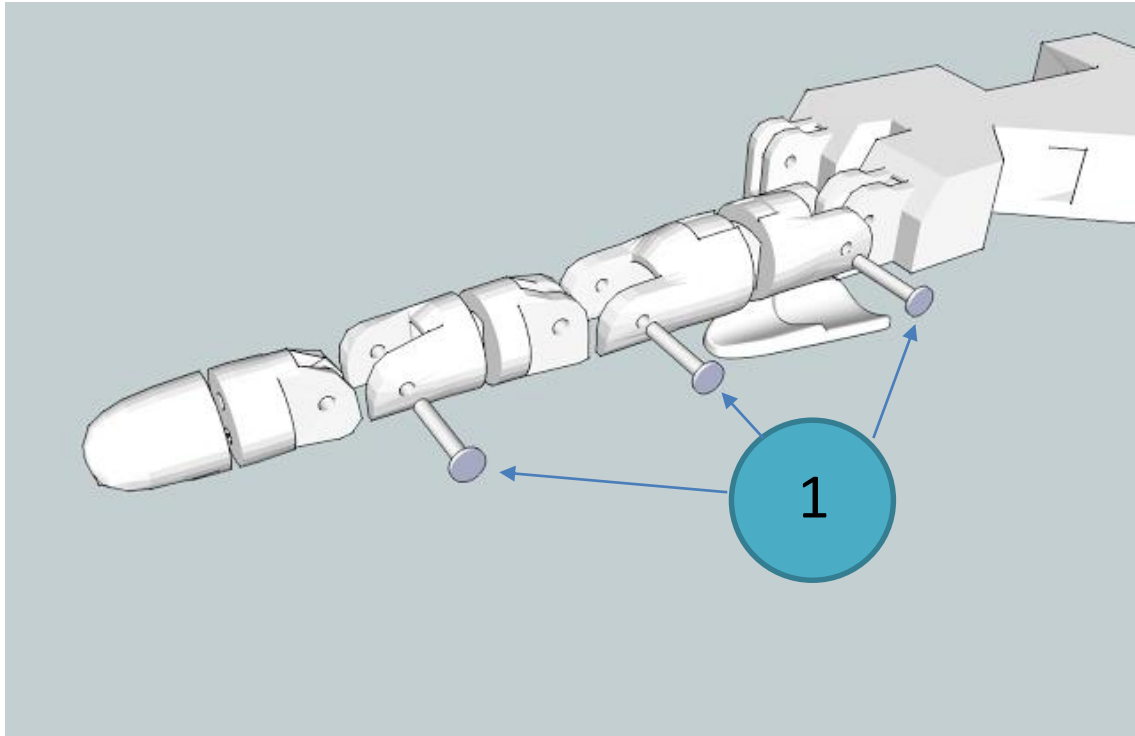


Figure 26: Middle finger assembly

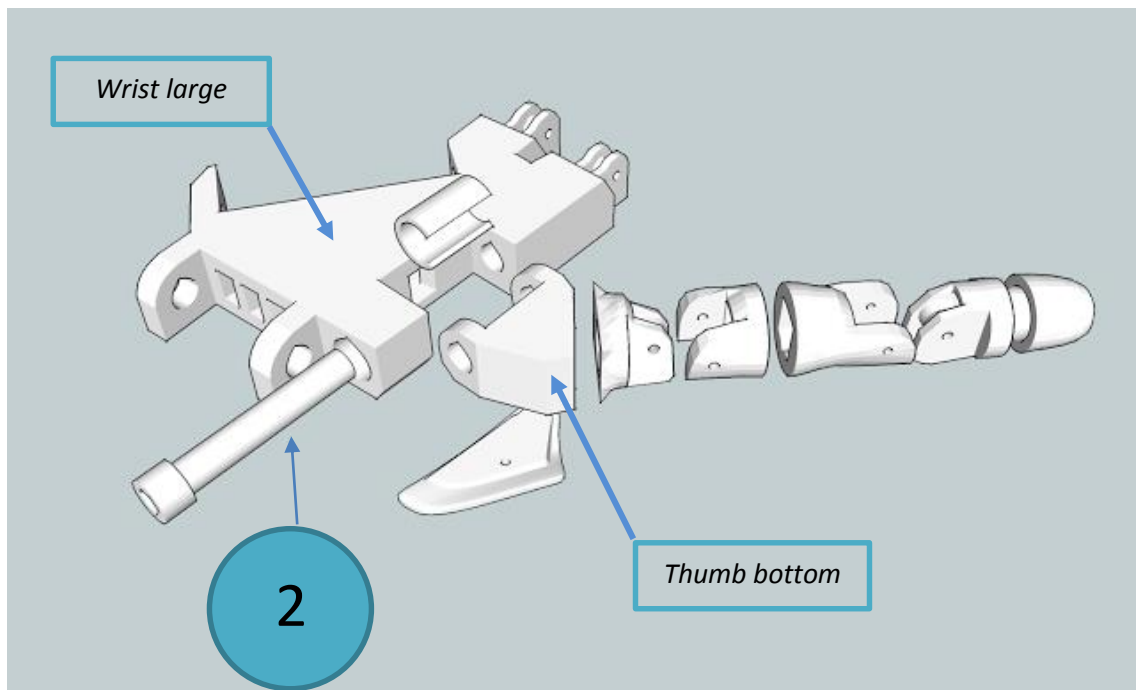


Figure 27: Thumb finger assembly

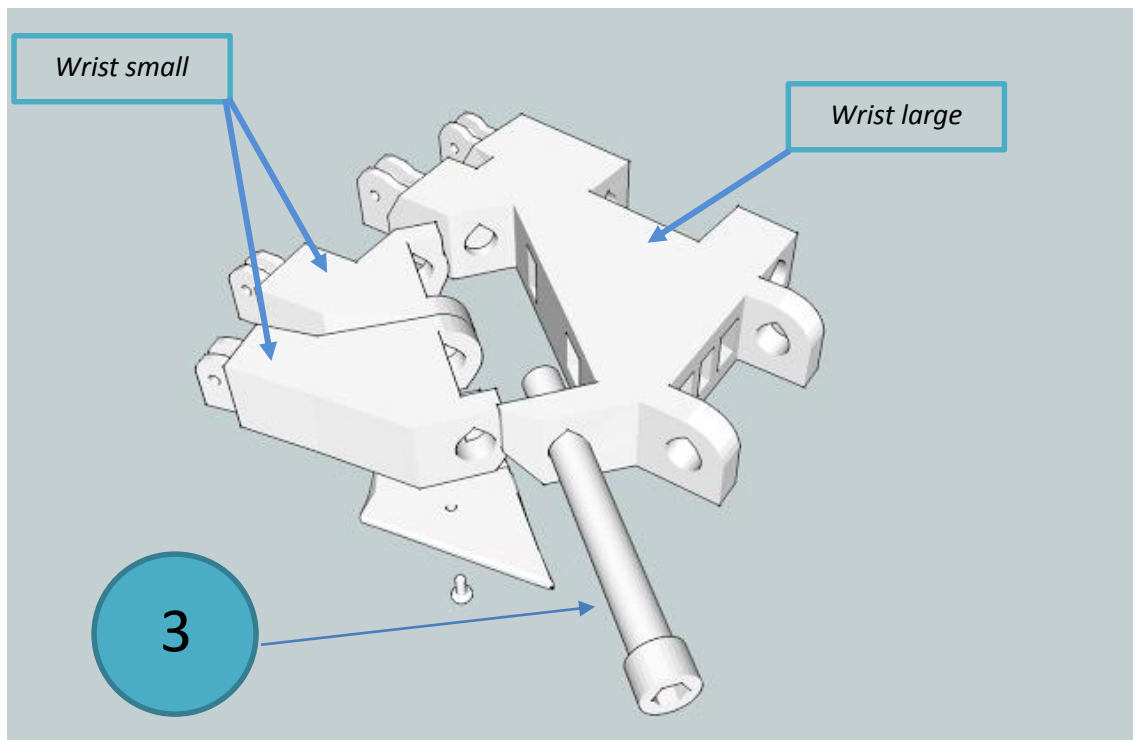


Figure 28: Wrist small with wrist large assembly

The Bolts:

14x2mmx2cm for the fingers (3 bolts each) [1]

1x8mmx4cm to attach *wristlarge* to *thumbbottom* [2]

1x8mmx8cm to attach *wristlarge* to *wristsmall* [3]

4.1.2. The Forehand

4.1.2.1. Design of the forearm pieces

It's been designed with *PTC Creo Parametric* some pieces in order to carry the hand, the servomotors and the electronic devices. Also the pieces are thought for a possible adaptation to an amputated arm, but for this prototype it's been put a simple bar to allow anyone use it inserting the hand inside and grabbing the bar.

First of all it's needed a piece which makes the function of a wrist.

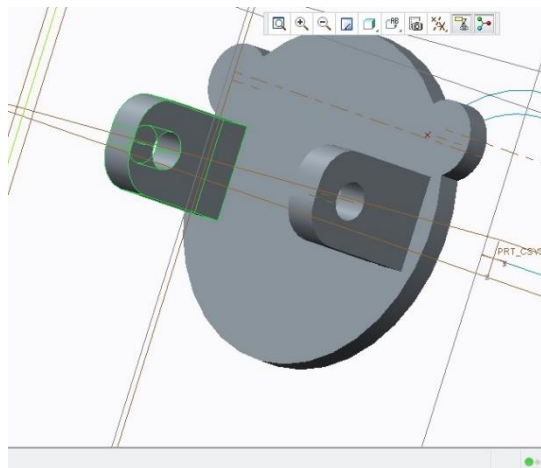


Figure 29: Wrist piece

That piece will hold the lowest part of the hand and it will be attached to the next forearm pieces.

The second piece is been designed in order to insert the servos taking into account the availability to access to the pulley easily and tense the strings and also be available to replace the servos with the screwdriver.

It's been tried to respect the proportions and the likeness to an arm.

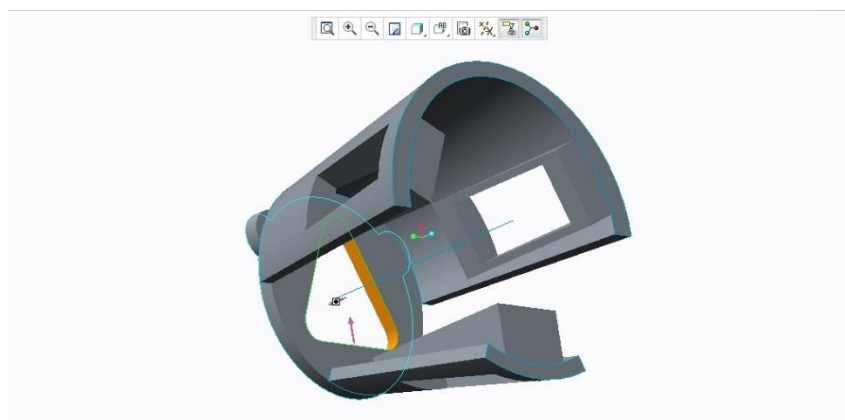


Figure 30: Servos inside piece

To see how the servos will adapt to the piece and to guarantee that they will not interfere between them, it's been modelled the specified servo and assembled all together.

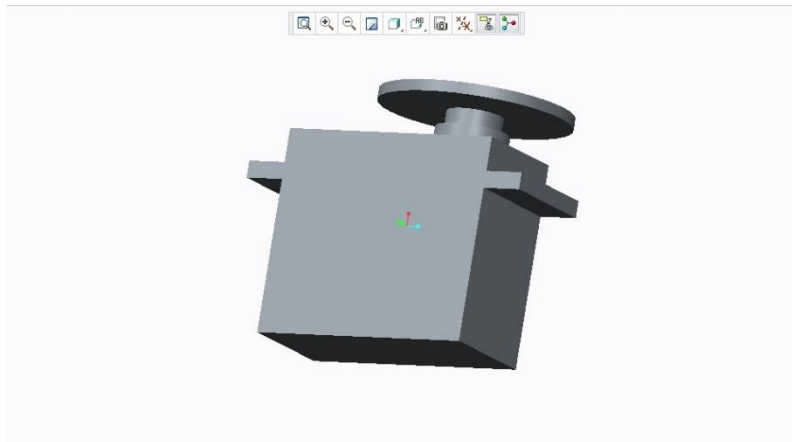


Figure 31: Servo modelling



Figure 32: Servos virtually assembled

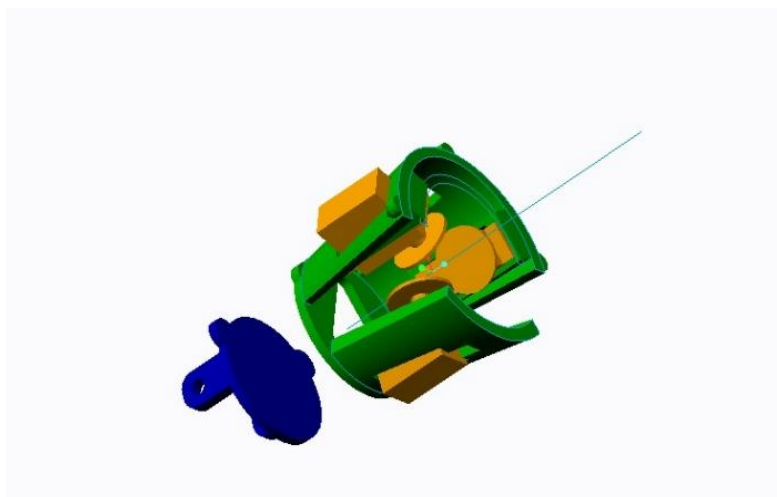


Figure 33: Servos virtually assembled and wrist piece

The final piece will be the one which will carry on the electronic board, *Arduino* with its shield, the 9 volts batteries and the 7 volts battery for the alimentation of the servos and *Arduino* device.

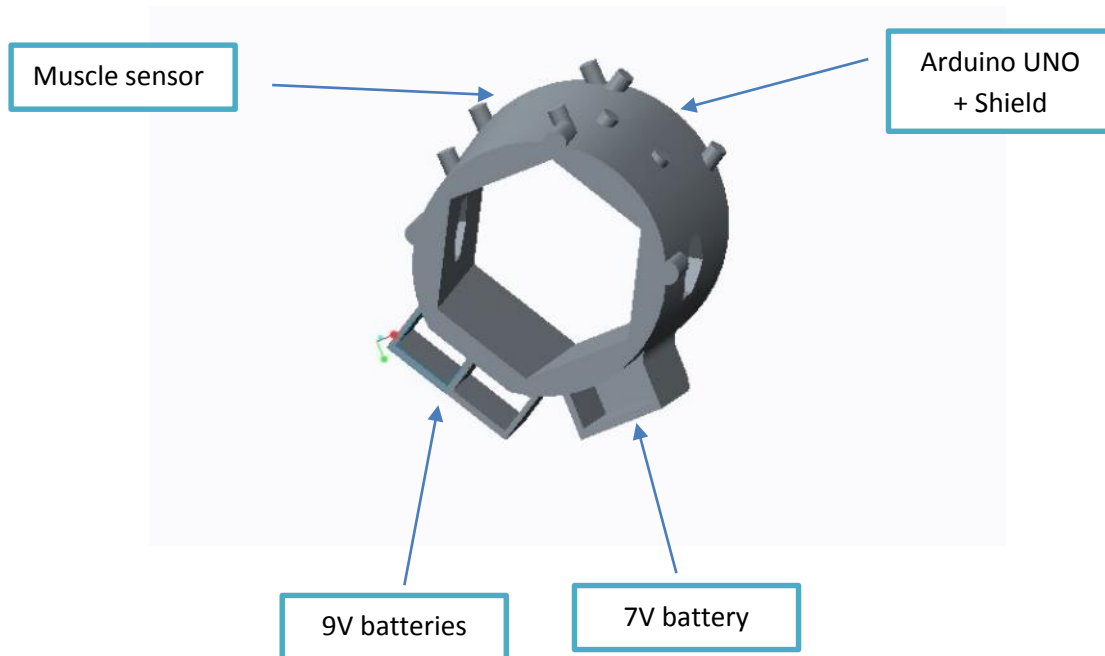


Figure 34: Holding piece of electronics

The hexagonal bar:

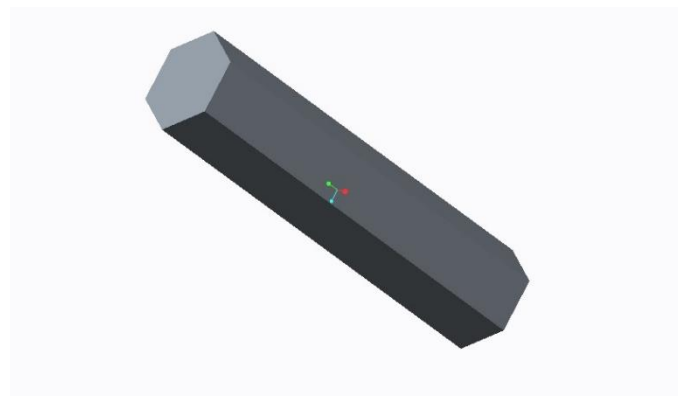


Figure 35: Hexagonal bar

The final virtual assembly of the forearm pieces.

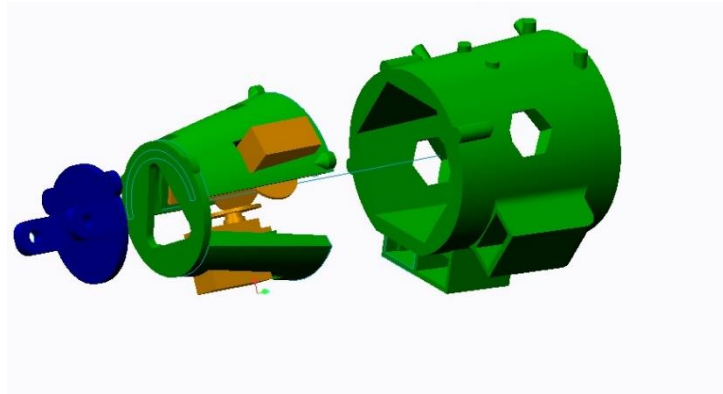


Figure 36: Assembly of the forearm pieces

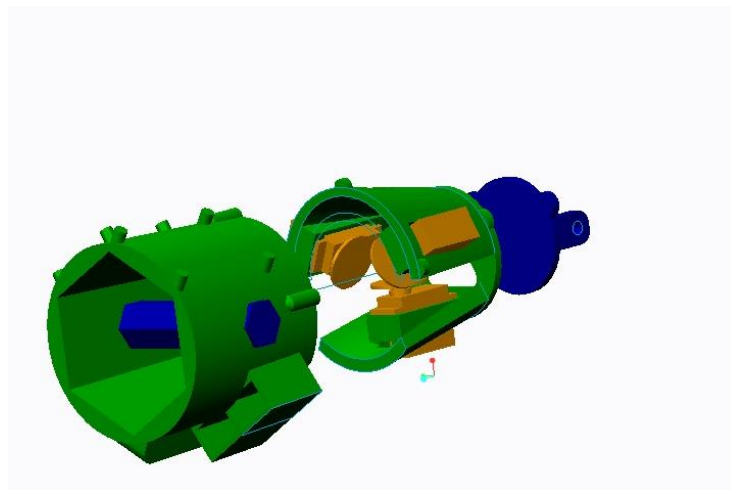


Figure 37: Assembly of the forearm pieces with hexagonal bar

4.2. 3D Printing

4.2.1. The 3-Dimensional Printer

In order to develop all the 3D pieces, a 3D printer called Prusa i3 has been used.

The Prusa Mendel i3 is the third version of the open source 3D printer Prusa Mendel. Our version is based on an aluminium frame cut by water jet cutter and threaded rods. Axis motion is made on linear bearings, belts and pulleys or threaded rods and NEMA 17 motors.

4.2.2. Specifications

	Single Sheet Frame	Box Frame
Plastic Parts (exc. Extruder)	26	29
Non Printed Parts approx.	337	293
Cost	\$300-1000?	\$300-1000?
Controller Electronics	Almost all RepRap	Almost all RepRap
Printing Size	200x200x200	200x200x270
Motors	5xNEMA 17 stepper	5xNEMA 17 stepper
Frame Material	6mm Aluminium, Wood	12 mm Wood
Frame Manufacture	Laser Cutter, CNC, Water Jet	Basic Woodwork Tools

Table 2: Prusa Mendel i3 specifications

- Print Area Dimensions: 200mm x 200 mm x 155 mm
- Maximum Print Speed: 150mm/sec
- Minimum Print Layer Height: 0.1mm or 100microns
- Frame: 5083 Aluminium 6mm thick, 10mm diameter galvanised threaded rod

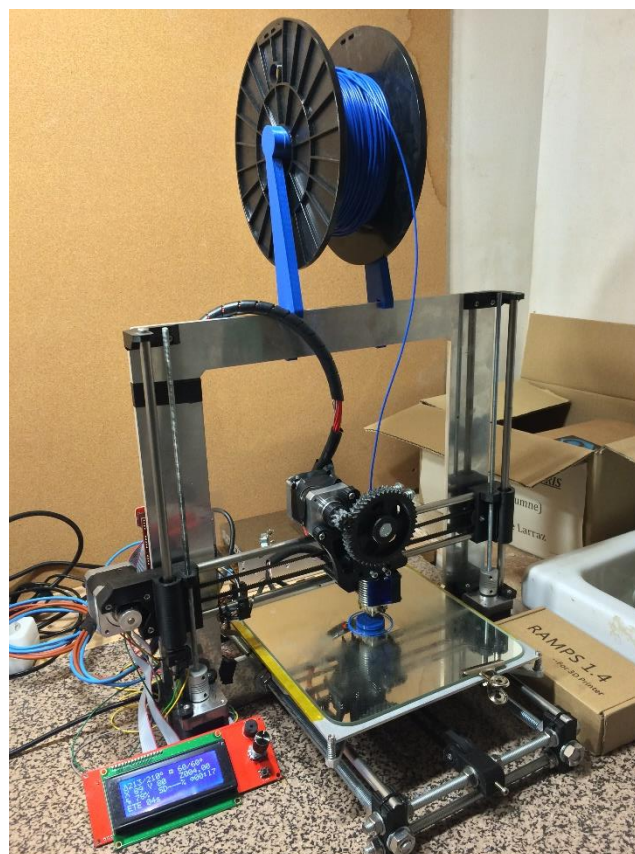


Figure 38: Rep Rap Prusa Mendel i3

4.2.3. Printing Material

The material used for printing all the pieces is PLA.

Polylactic acid or **polylactide** (PLA, Poly) is a biodegradable thermoplastic aliphatic polyester derived from renewable resources, such as corn starch (in the United States), tapioca roots, chips or starch (mostly in Asia), or sugarcane (in the rest of the world).

It is harder than ABS, melts at a lower temperature (around 180°C to 220°C), and has a glass transition temperature between 60-65 °C, so is potentially a very useful material. It does exhibit higher friction than ABS however which can make it difficult to extrude and more susceptible to extruder jams.

4.2.3.1. Configuration

There so, the configuration of the Prusa i3 has been such as:

- Base Temperature: 60°C
- Extruder Temperature: 210°C
- Resolution: 100µm

4.2.3.2. Software

Once the printer is well configured and the pieces have been modelled in a specific format to be printed (.stl), the next step is to upload them to Repetier-host, a free program for Prusa i3.

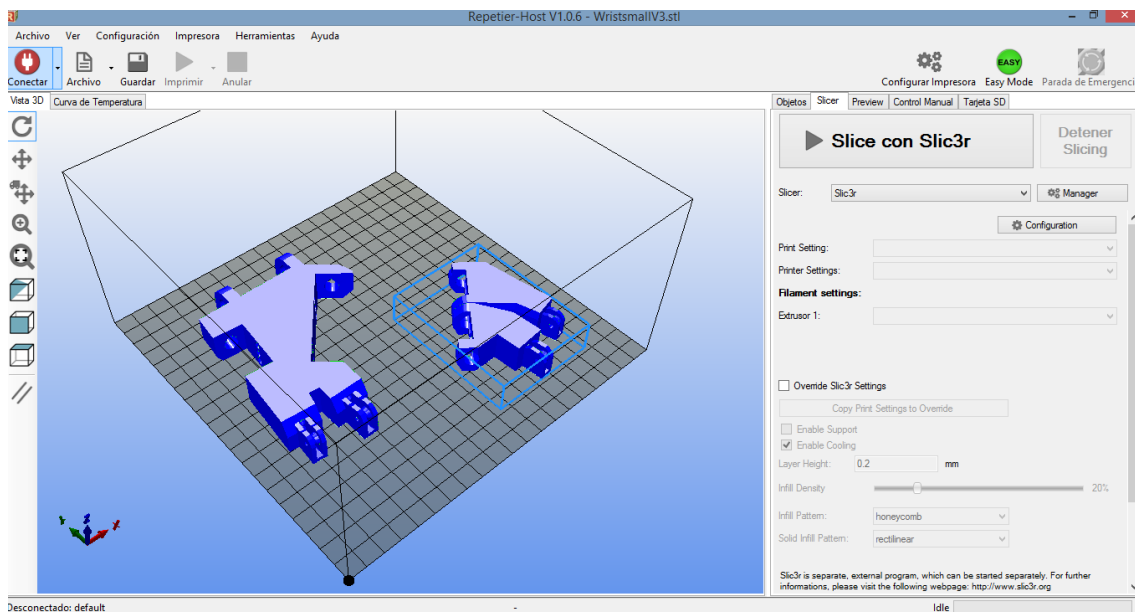


Figure 39: Repetier Host with wrist small and wrist large pieces

In the previous figure, it can be seen the size of the pieces on the base and the way they will be printed.

Slic3r is the program which generates a code full of coordinates (x, y, and z) to make the extruder move around them and perform the piece layer by layer.

In Figure 40 an example of the G-code created by Slic3r to generate the *Wrists* parts.

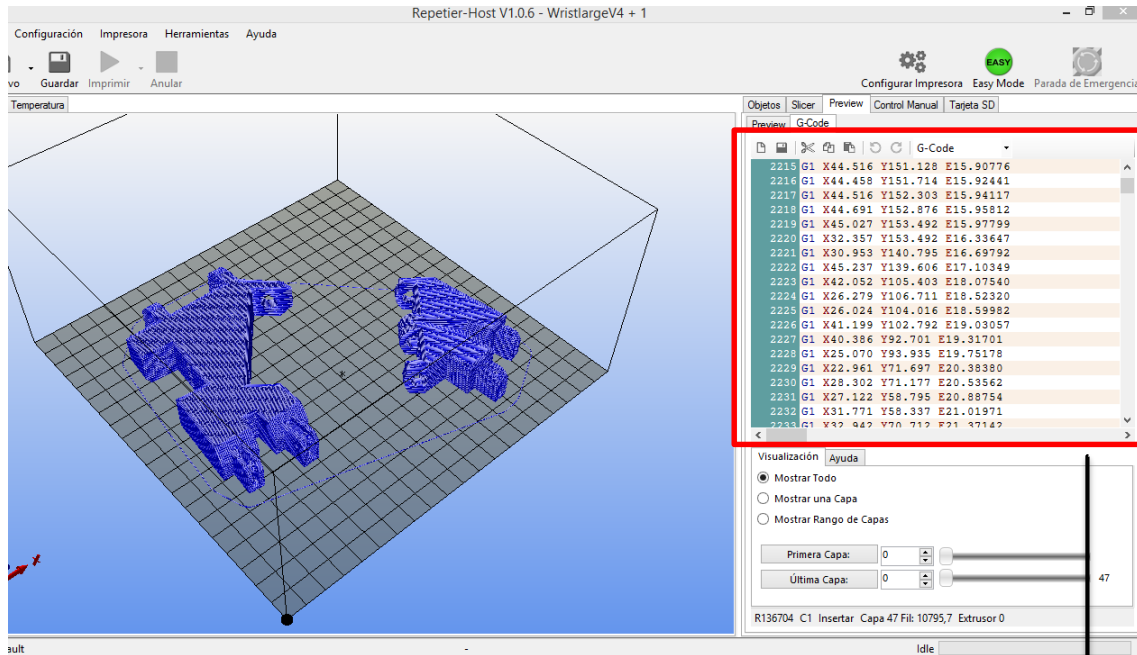


Figure 40: Pieces interpreted with Slic3r generating the G-code

In fact for these pieces the *Slic3r* has generated 136.703 lines of code.

A zoom with the example of the G-code generated:

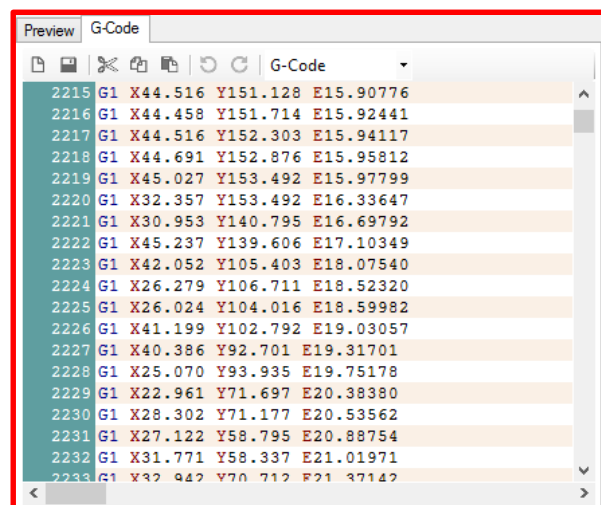


Figure 41: Zoom of the G-Code

Once the G-code has been generated the printing process can start.

4.3. Printing the 3D pieces

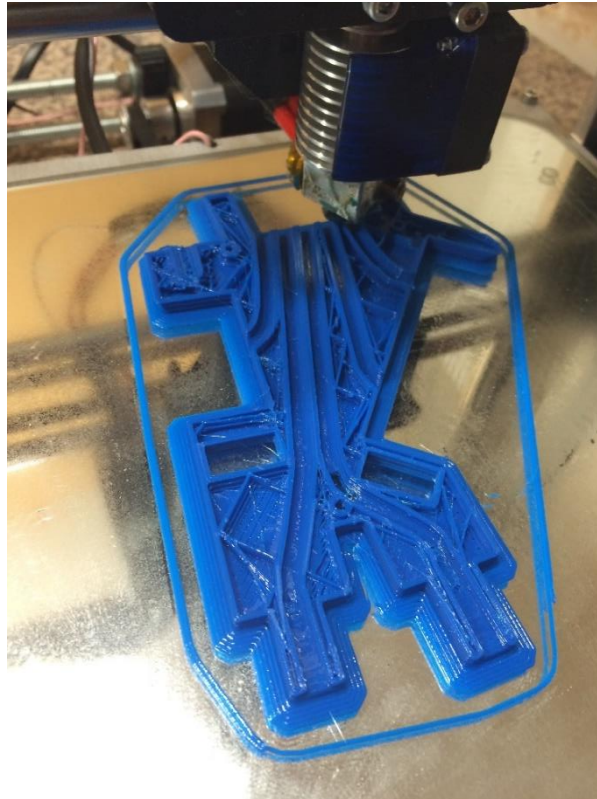


Figure 42: Printing wrist large piece

Sometimes if it isn't taken care about the speed of the extruder/extrusion or the temperatures of the base or the plastic extruded some problems like the following could appear.

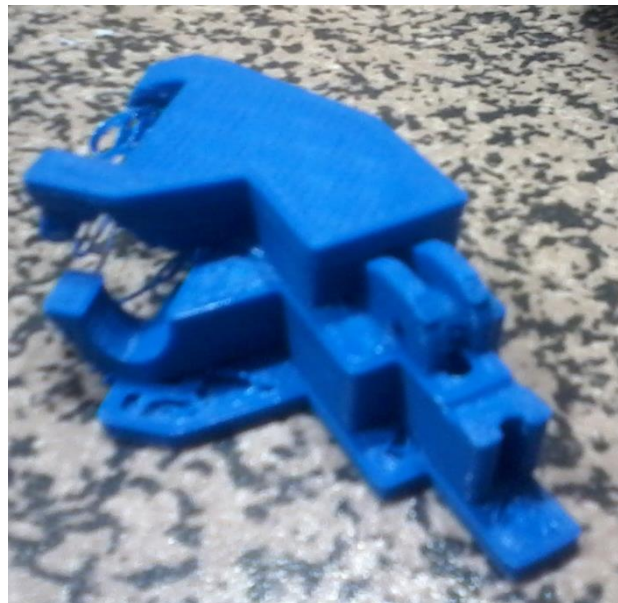


Figure 43: Failed piece, layers displaced

Some pieces of the hand printed.

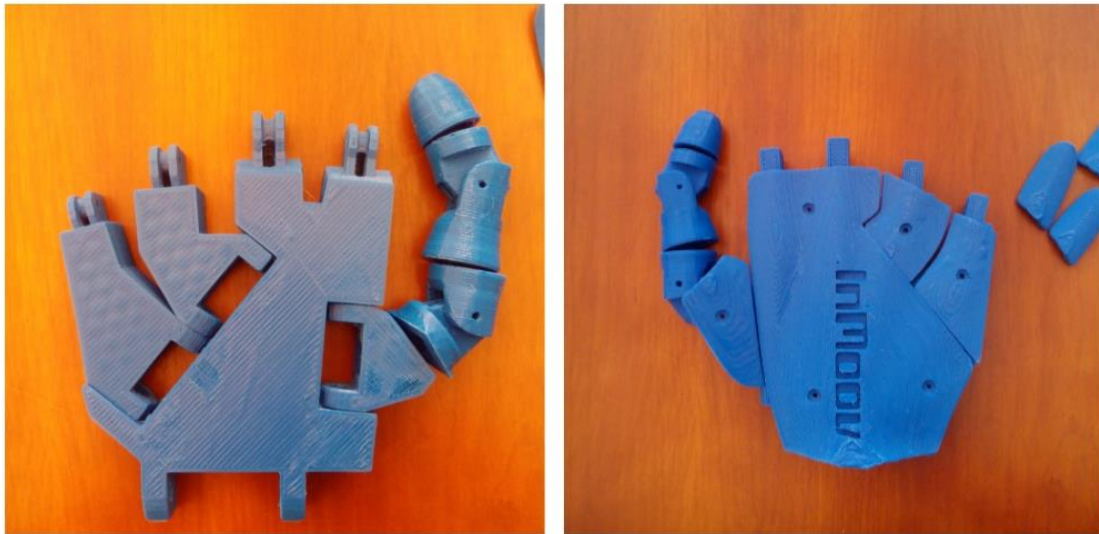


Figure 44: Inmoov Right hand pieces

The result of the *Inmoov* hand printed:



Figure 45: Inmoov Right Hand printed

The result of the designs with the 3D printer.



Figure 46: Wrist piece profile

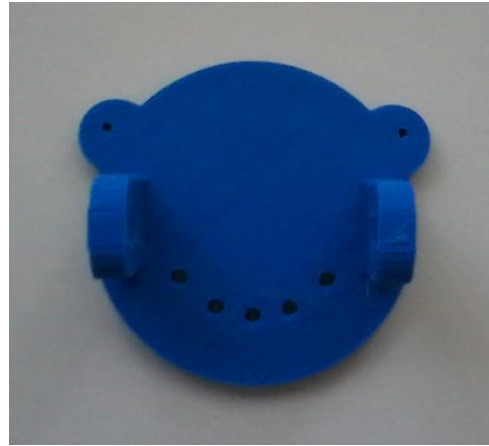


Figure 47: Wrist piece top view

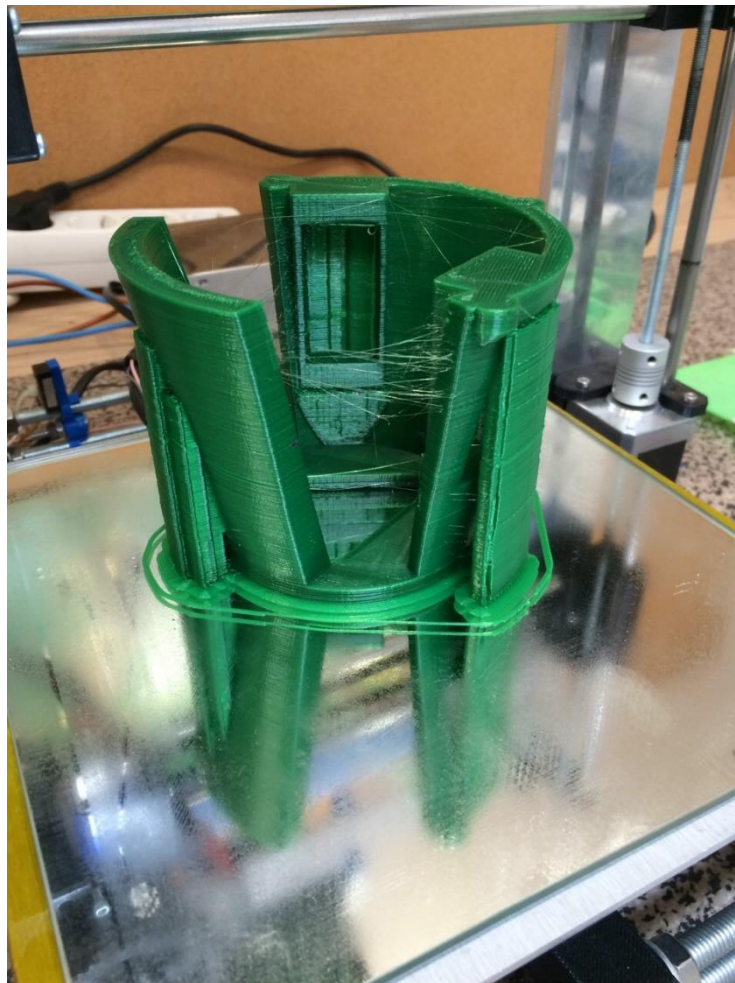


Figure 48: Servos piece just printed (18h 30 min)

Printing the last piece of the electronic devices:

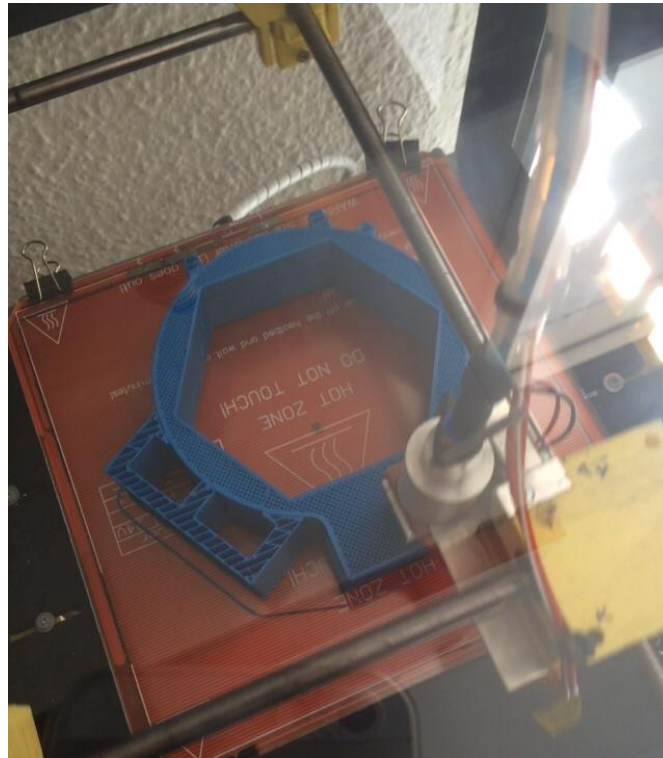


Figure 49: Electronics piece while printing

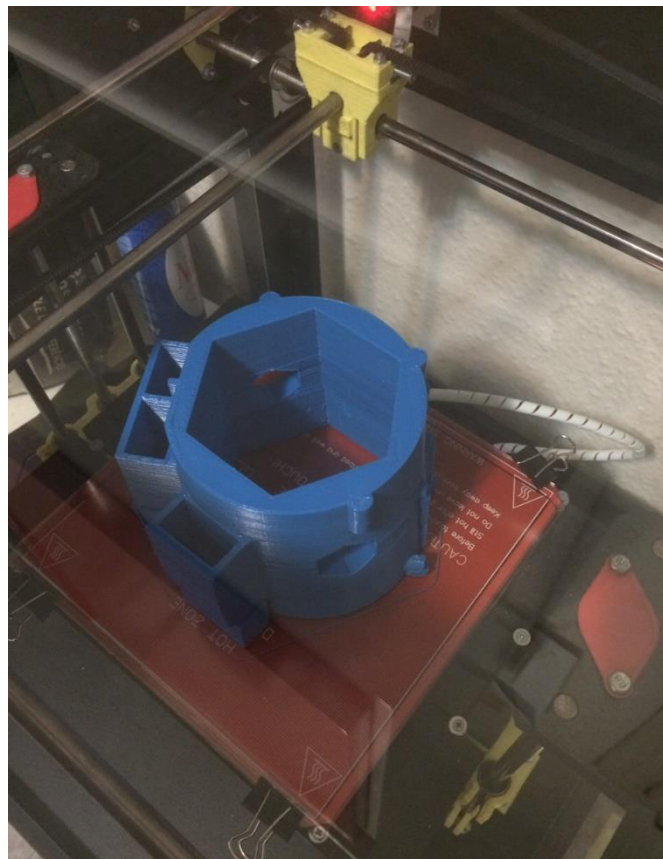


Figure 50: Electronics piece printed (13h. 18min)

4.4. Assembly of the prototype

Make the holes wider with a 3mm drill in order to pass the screws easily through.



Figure 51: Drilling pieces with 3mm drill

Sand the inner part of the pieces to achieve a low friction between moving parts.



Figure 53: Sanding the piece



Figure 52: Sanding inner parts

Pass through the holes the 2mm bolts and test the friction of the movement, rather be as low as possible.

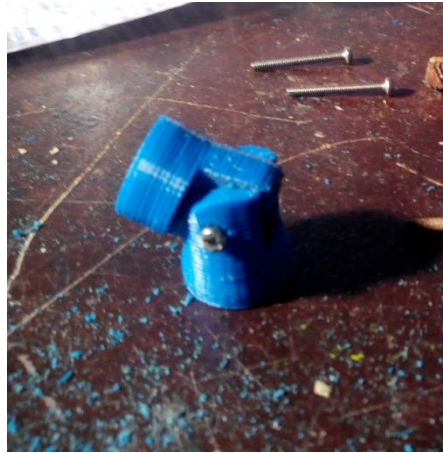


Figure 54: Pass the 2mm screw and test friction

The combination of this two glues make a stronger one to attach the wanted pieces



Figure 55: Combination of two powerful glues

Hold the glued pieces with tweezers at least during 10 hours.

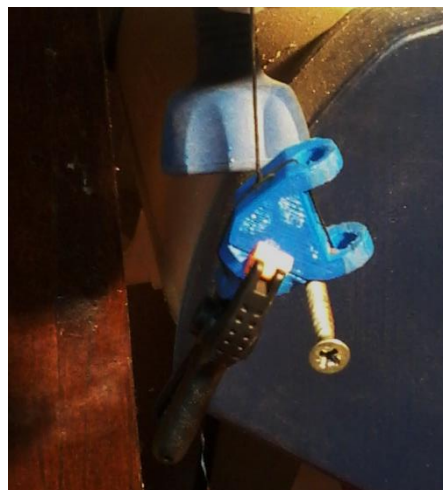


Figure 56: Holding the glued piece

Make at the wrist parts the 8mm holes with a drill.



Figure 57: Drilling the 8 mm holes

Then pass the fishing lines through the inner holes until the top of each finger.

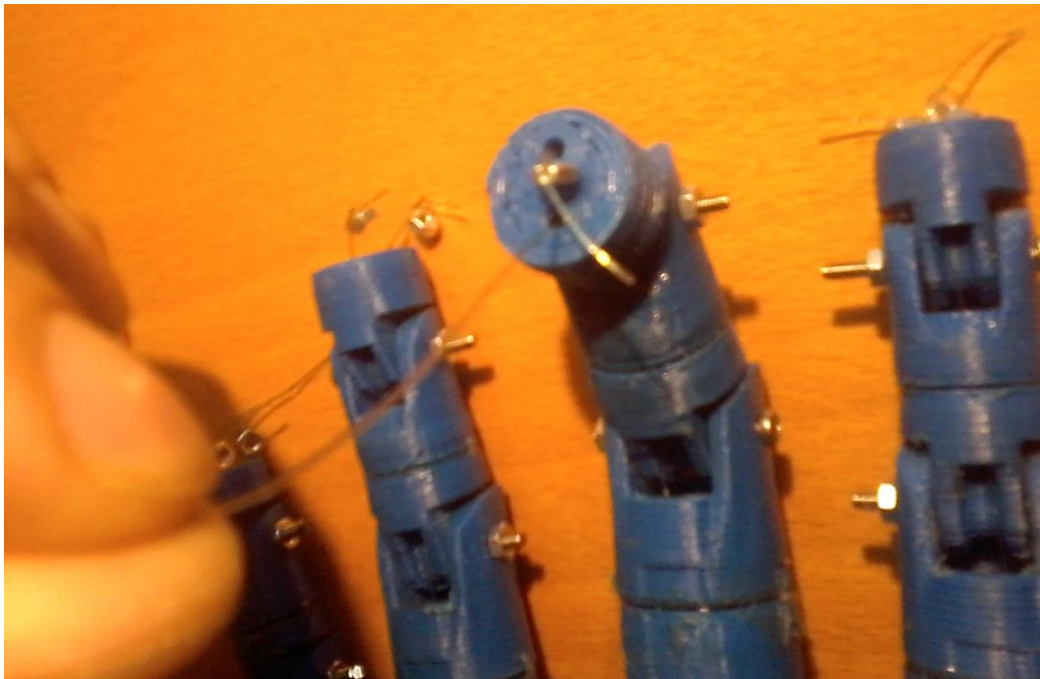


Figure 58: Passing and attaching the fishing lines



Pass the Allen bolts to attach the fingers and tighten the screws with nuts.

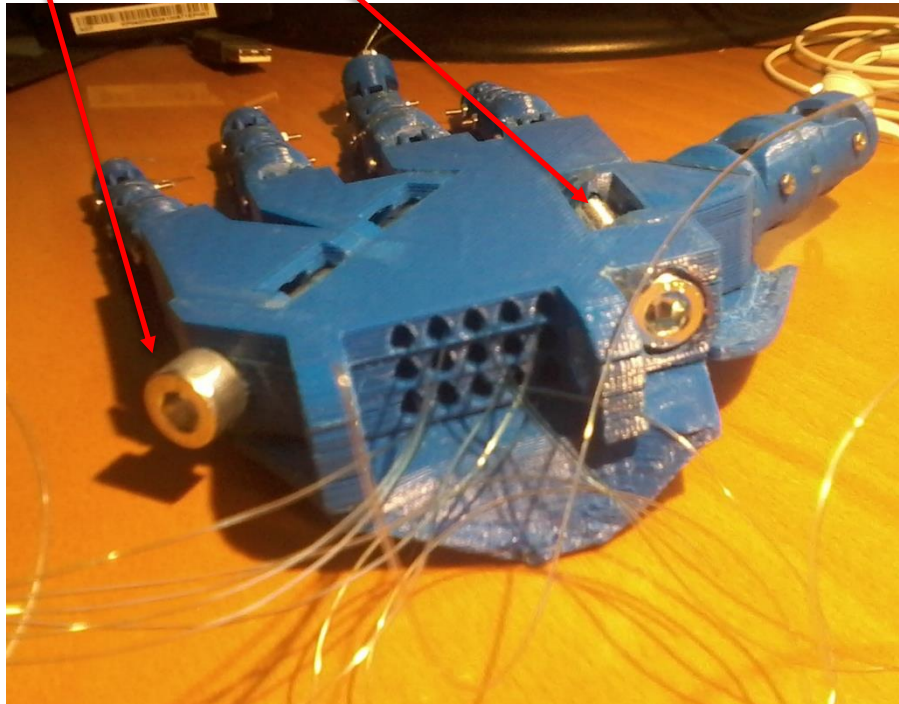


Figure 59: Allen screws

Let the strings go out through the bottom holes of the *wrist large* part.

Disposal of the two holes to fit the nuts and twist the screws throughout:

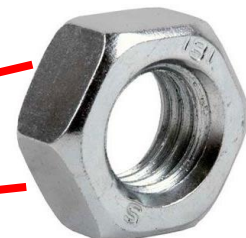
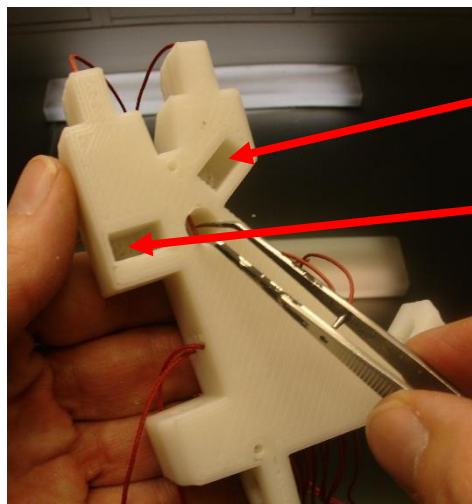


Figure 60: Female allen nuts position

Once assembled all the parts, test the movement pulling the fishing lines, the fingers must flip.

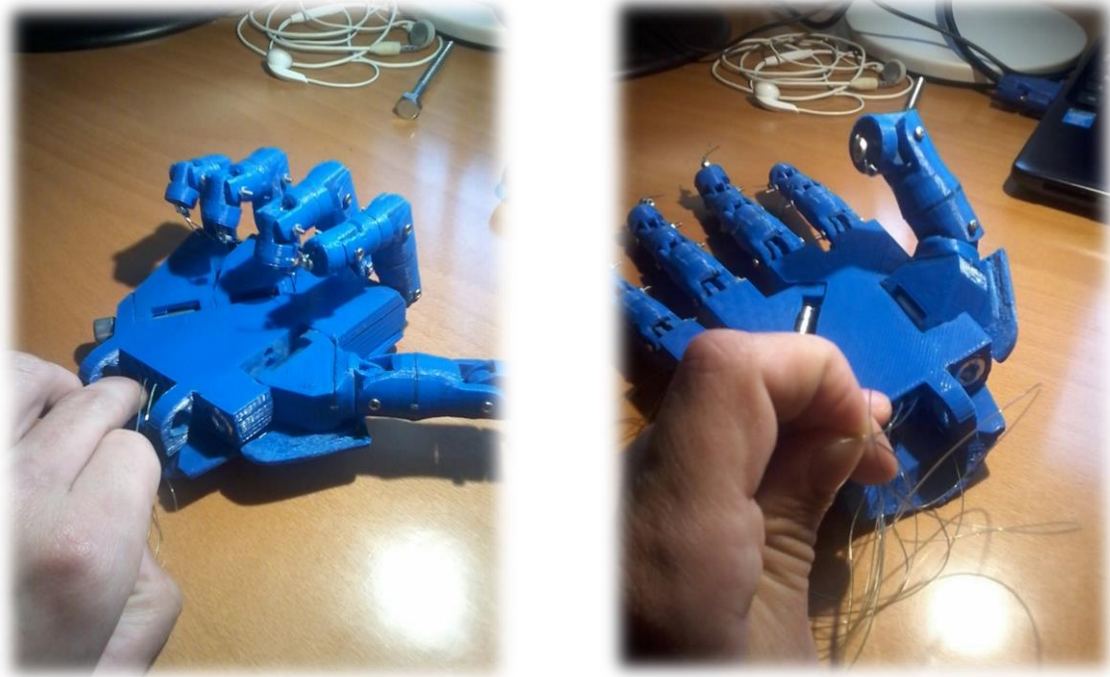


Figure 61: Testing the fingers movement

Wrist attach with an 8mm bolt, that wrist doesn't allow any movement because of some considerations of the force of the servomotors, could it be a problem for the movement

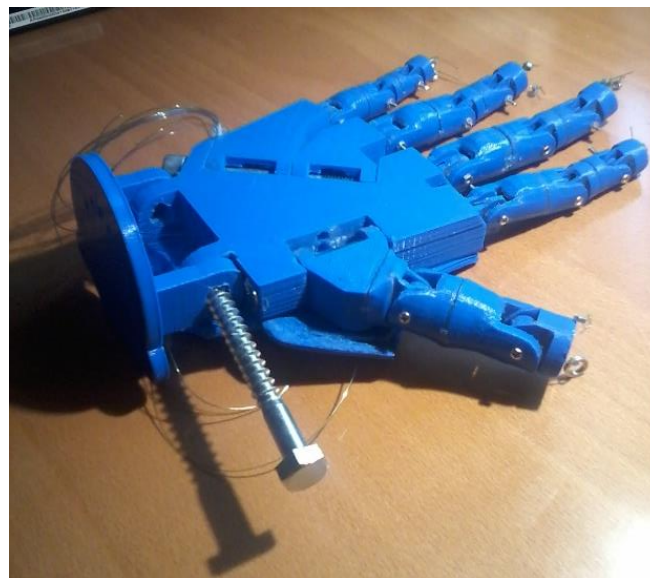


Figure 62: Hand with wrist attached

Screwing the servos to the forearm piece.

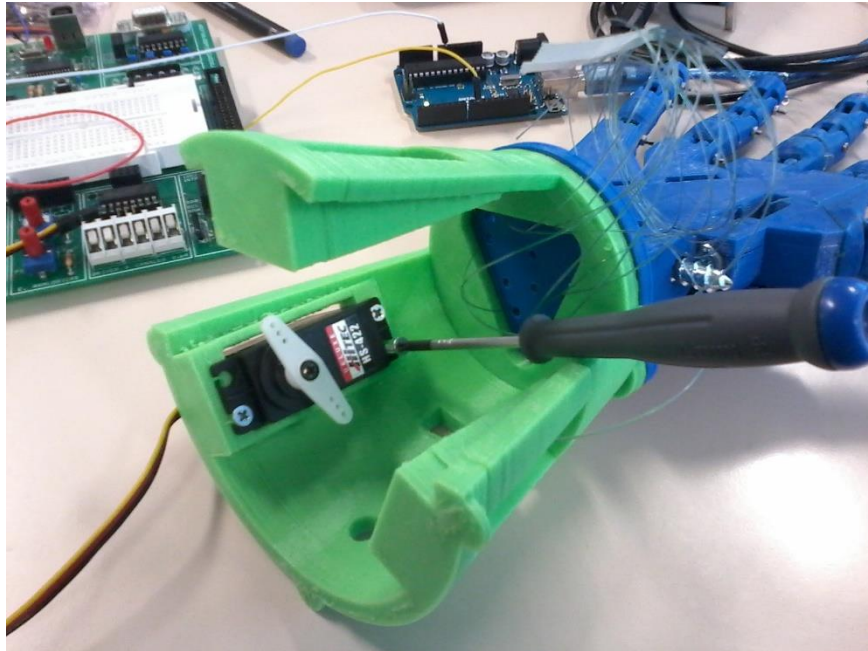


Figure 63: Servo attach to servos piece

Tie the fishing lines to the pulley of the servo, then it's been hold each line with an improvised system of screws and threads in order to maintain the tension.



Figure 64: Fishing line attach to servo



Figure 65: Holding System of strings

Results of the servos assembled and tied.

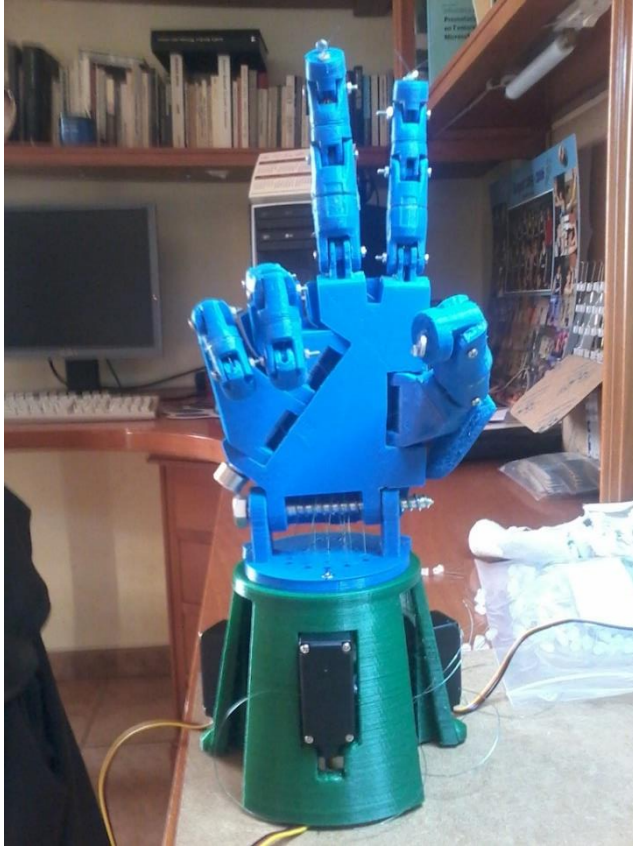


Figure 67: Hand and wrist with servos piece



Figure 66: All servos with each strings

Join the last piece where the power supply and the electronic boards will take place.



Figure 68: Assembly of all pieces

4.5. PCB design and manufacturing

4.5.1. The Muscle Sensor Board

In this part the main idea is read signals from the muscles, actually from the biceps, send them to Arduino and through it move three servos in function of the force, from 0 to 180 degrees.

The signal will be processed electronically, three electrodes are going to be placed in the arm in this way:

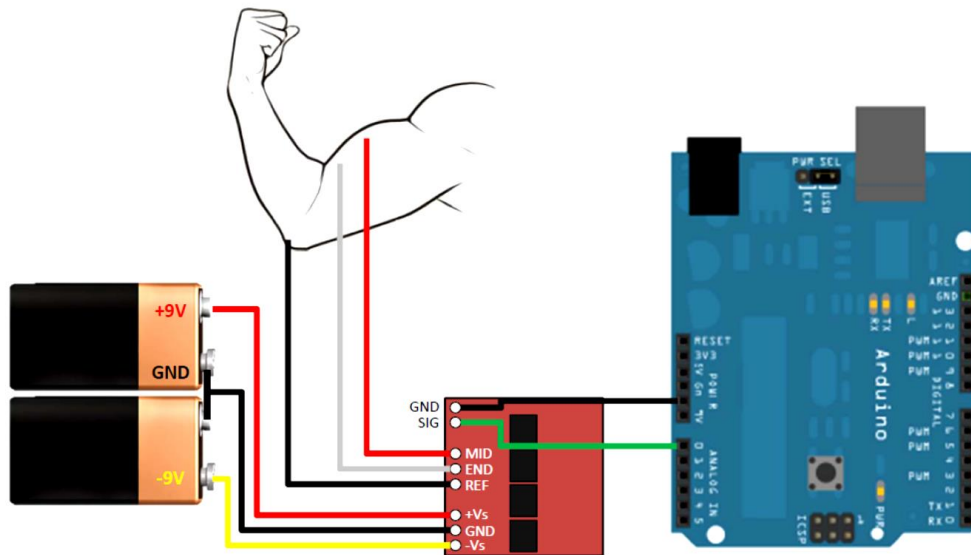


Figure 69: Scheme of connections between Arduino, electrodes, muscle sensor and batteries

4.5.1.1. The Electrodes

Figure 70 shows the kind of electrodes used:



Figure 70: Electrodes

An electrode is a conductor, not necessarily metallic, through which a current enters or leaves a non-metallic medium, as an electrolytic cell, arc generator, vacuum tube, or gaseous discharge tube.

We take profit of the bidirectional way it could be used to read the small voltage signals of an arm.

The electronic device is based on a scheme from Advancer Technologies of Muscle Sensor, there are some models of muscle sensor devices in his webpage. The chosen one and modified in relation of the progresses is the latest version v3:

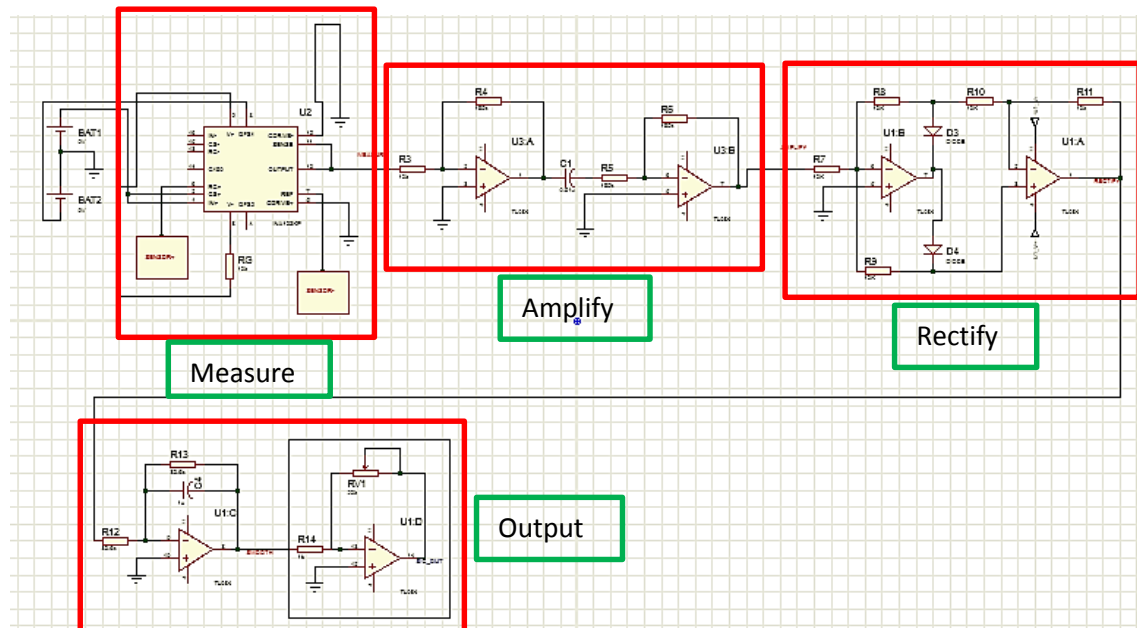


Figure 71: Proteus scheme of muscle sensor

It has been schematized in *Proteus 8* and made the design of the final board with the same program.

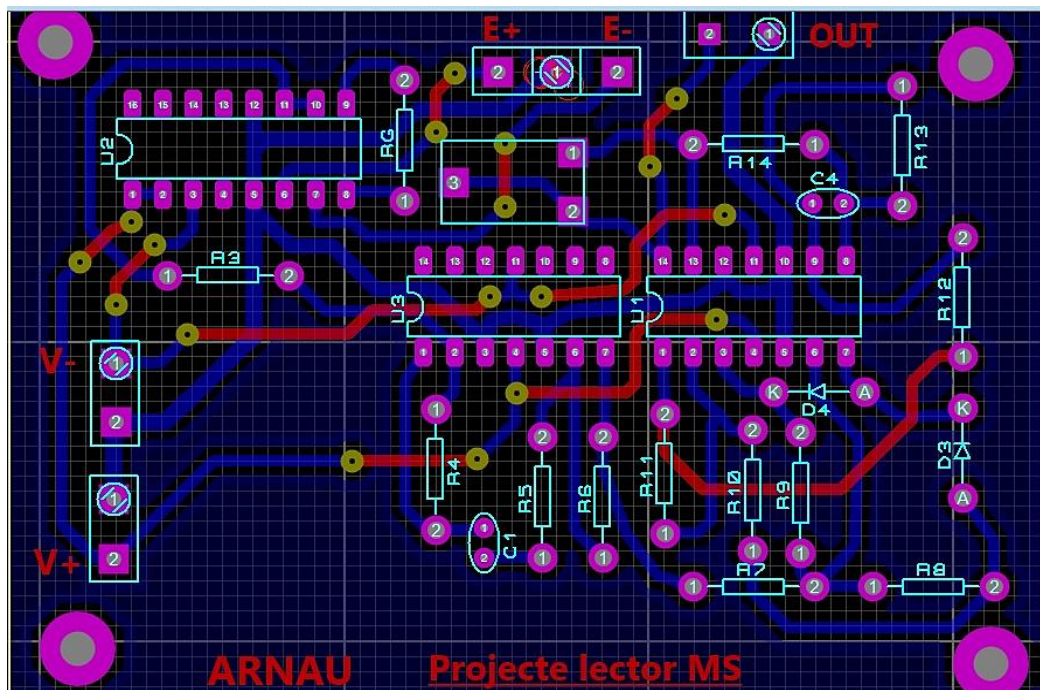


Figure 72: Layout scheme of final muscle sensor board

— Top layer — Bottom layer

The Circuit Schematic of the muscle sensor has four steps:

Measure:

Measure of the signal coming of the sensors, recommended an INA106, actually used in this project with an INA125P.

INA125P is a differential amplifier that amplifies the difference between two input voltages but suppresses any voltage common to the two inputs. There so, it amplifies the difference between the two electrodes but not its own voltage.

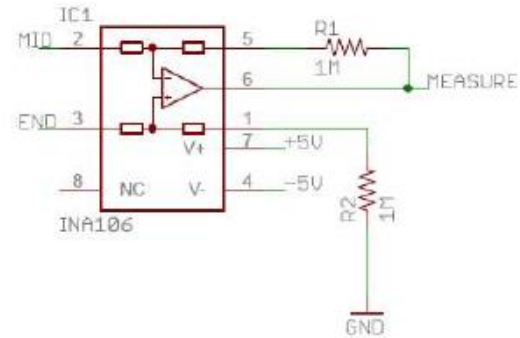


Figure 73: INA106 (Measurement)

To calculate the output voltage is such as:

$$I_1 = \frac{V_1 - V_a}{R_1}, I_2 = \frac{V_2 - V_b}{R_2}, I_f = \frac{V_a - V_{out}}{R_3}$$

Summing point $V_a = V_b$

And $V_b = V_2 \left(\frac{R_4}{R_2 + R_4} \right)$

If $V_2 = 0$, then: $V_{out(a)} = -V_1 \left(\frac{R_3}{R_1} \right)$

If $V_1 = 0$, then: $V_{out(b)} = V_2 \left(\frac{R_4}{R_2 + R_4} \right) \cdot \left(\frac{R_1 + R_3}{R_1} \right)$

$$V_{out} = V_{out(a)} + V_{out(b)}$$

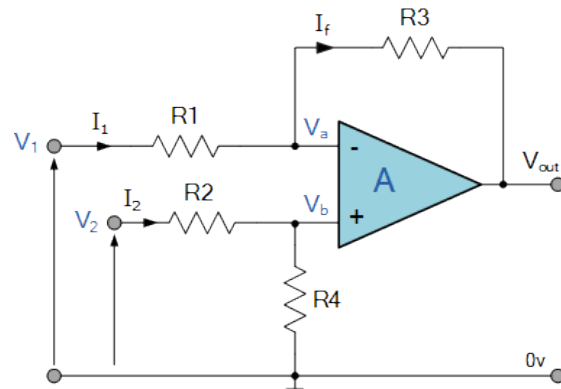


Figure 74: Differential amplifier

When resistors, $R_1 = R_3$ and $R_3 = R_4$ the above transfer function for the differential amplifier can be simplified to the following expression:

$$V_{out} = \frac{R_3}{R_1} \cdot (V_2 - V_1)$$

Amplify:

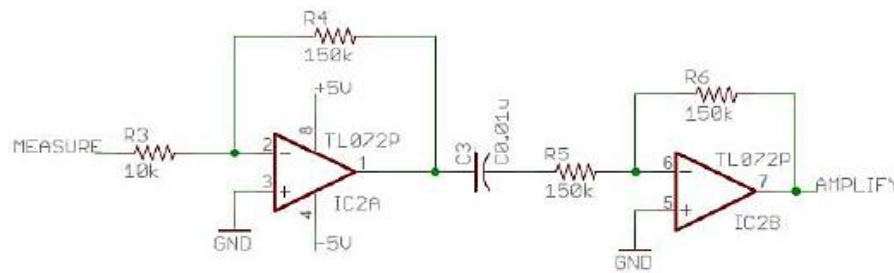


Figure 75: TL072P (Amplify)

Amplify the signal from the measurement with operational amplifiers, there recommended TL072P but actually used TL084.

Basically the operational amplifiers take an input signal and amplifies it with a relation of gain between resistances.

That kind of operational amplifier shown on Figure 769 is actually an inverting amplifier.

$$V_{out} = -V_{in} \cdot \frac{R_f}{R_{in}}$$

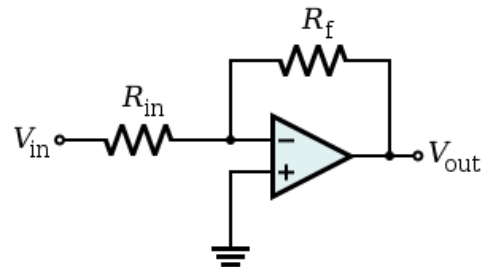


Figure 76: Inverting Operational Amplifier

In addition are placed some capacitors at the entrance of the signal in order to suppress any alternate current.

Rectify:

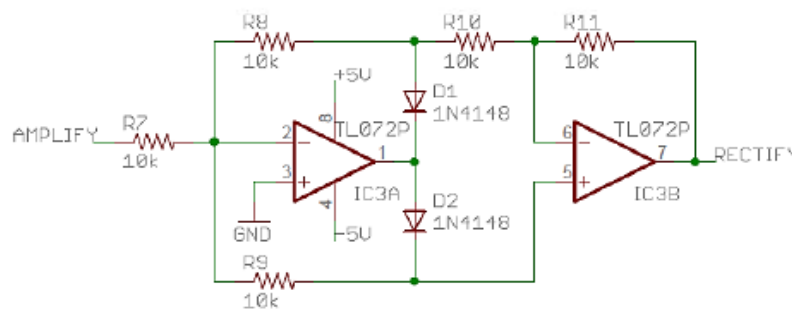


Figure 77: Rectification stage

On this phase the amplified signal is rectified in order to not allow the signal take negative values. It flips all the negative values to positive with the diodes.

Output:

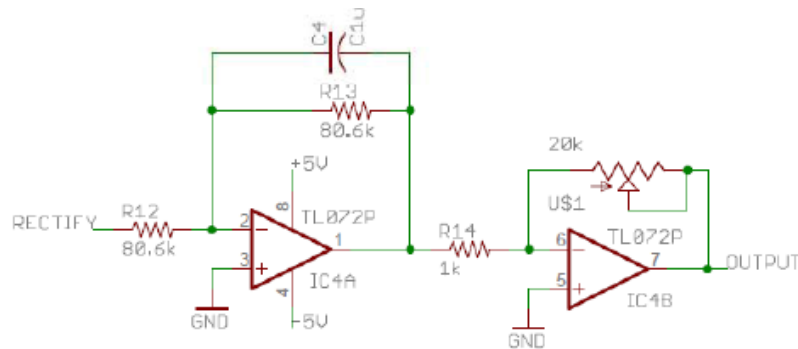


Figure 78: Output

The final stage is composed by an integer of the signal and another operation amplifier with an adjustable output gain in order to control the peaks of the outgoing voltage. The integer softens the signal.

To sum up with an example of wave, in Figure 7952 it's shown what would happen with a Sine wave through the circuit.

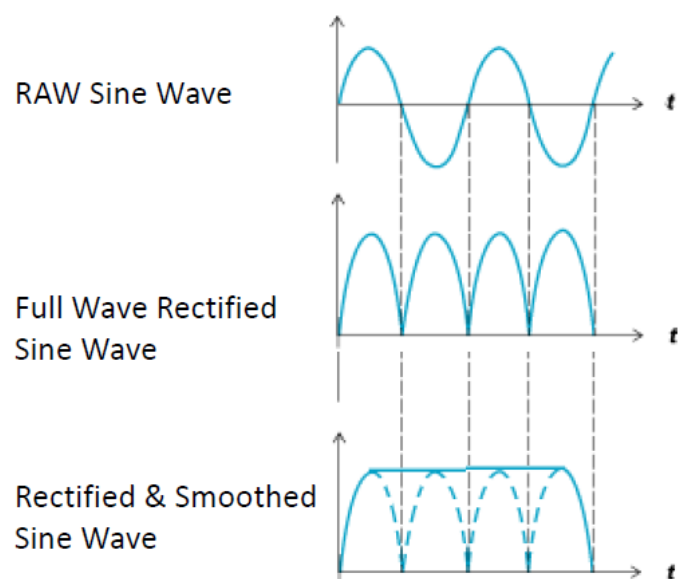


Figure 79: Example of the board working with Sinusoidal wave

4.5.2. Arduino Shield

It's been decided to put an information screen above, for this kind of gadgets added to an Arduino, exists an invention called Shield. It consists on an electronic board which fits perfectly in Arduino's pins and it's possible to manage for connecting a lot of different devices.

In this case it has been decided to put a TFT screen, also will be the board where servos will be connected and provided with energy and the signal. Due to the Shield will cover all the pins from Arduino's board, all the connections (in / out) have to be on it. In fact, the Analogical Input from Muscle sensor board will also be on. Moreover, the power supply from the 7V battery will also feed Arduino and the servomotors as it is going to be explained below.

4.5.2.1. The TFT Screen

The screen chosen its TFT ILI9225 with 2.2" of screen and a resolution of 176x220.

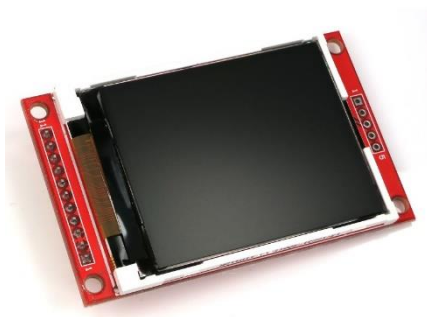


Figure 81: TFT screen front

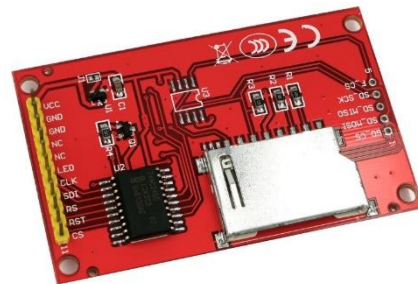


Figure 80: TFT screen back

11th Pin connection:

1. VCC--Power supply (5V/3.3V)
2. GND-- Power supply (GND)
3. GND-- Power supply (GND)
4. NC--No connect
5. NC--No connect
6. LED-- Pick IO control back light off or PWM brightness control
7. CLK--SPI clock signal
8. SDI--Serial data input pin sda
9. RS--Command (RS=0)/ Parameter (RS=1)
10. RST--Reset pin
11. CS-- Chip selection pin

4.5.2.2. The Shield Scheme

Basically the shield will contain all the elements shown below.

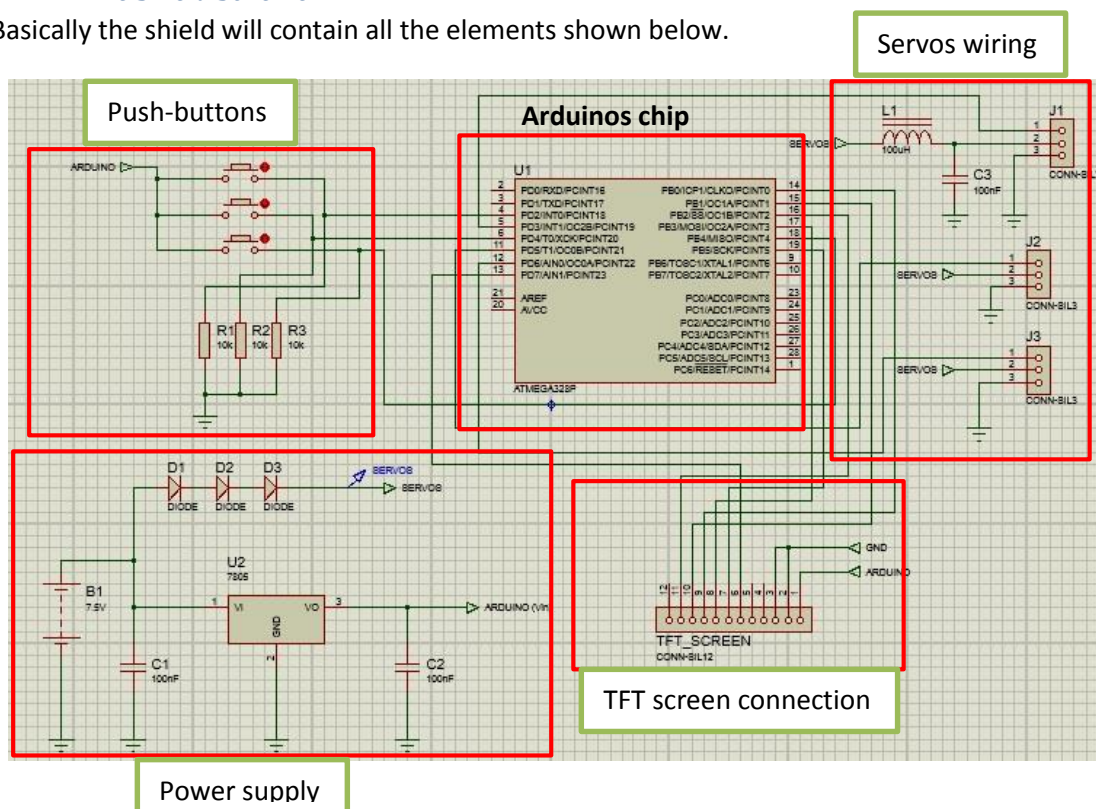


Figure 82: Shield scheme and configuration

Push-Buttons:

It's been added three surface push-buttons in order to facilitate the configuration of different aspects such as force calibration and Mode changing.



Figure 83: Push-button

Servos wiring:

The Servos connection is composed by three wires, two of alimentation (V+ / V-) and the third to send the signal which commands the position of the servo between (0-180°).

As the Figure 84: Servos wiring shows, the type of connector is like that.

It's been added a filter based on a capacitor and a coil to avoid interferences from the servos.

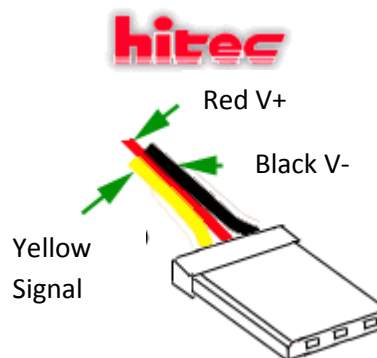


Figure 84: Servos wiring

Arduinos chip (ATMega328P):

Although it is not going to be used Arduino's chip apart of its board, only the scheme has been designed as if the chip has been ejected and connected with a socket, respecting the inner configuration of the corresponding pins of the board, as it is shown in the next figure.

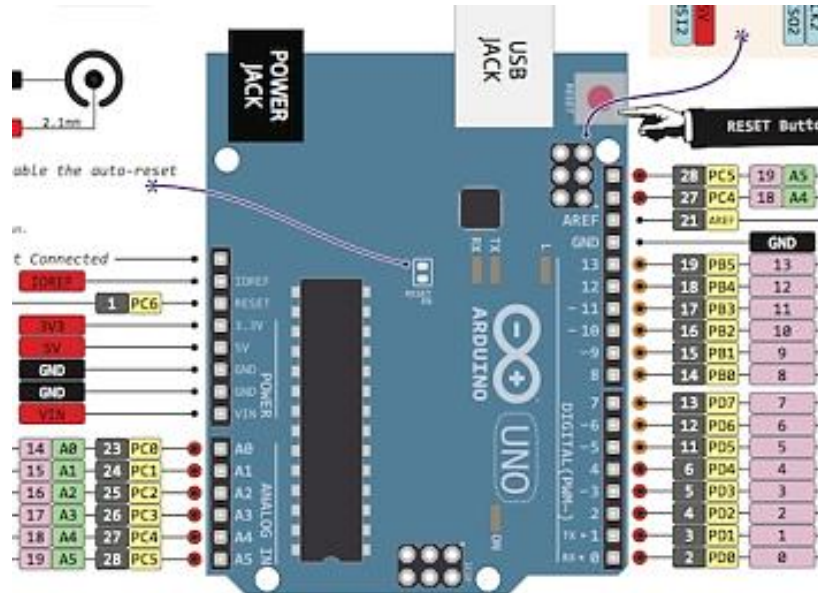


Figure 85: Arduino's inner configuration

Power supply:

On the one hand, it is needed an external device to provide energy to each servomotor, it's been thought to put a battery about 7 Volts and to supply the servos just add three diodes to make the voltage decrease from 7V to 5V ($7 - (0.7 \times 3) \approx 5V$), which is the best voltage to work for this type of servos.

On the other hand, the same battery will provide the energy to Arduino in order to make it work. Arduino's board, needs a 5V voltage to turn on, at the *Vin* pin. To make sure that this voltage will be constant and exact, it's been decided to put a device which takes an input voltage between 7 to 25 V and supplies a constant 5V output voltage. This device is called LM7805

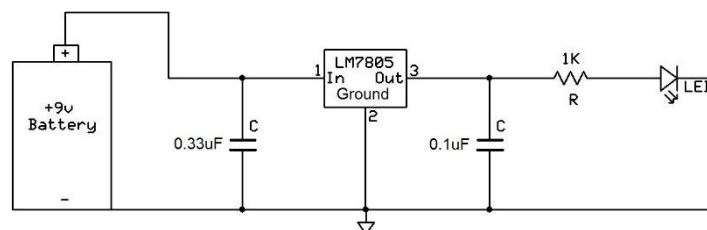


Figure 86: LM7805 configuration

At Figure 86 it is shown the way to be connected properly.

Finally the Shield configuration on the board obtained is shown on the figure below.

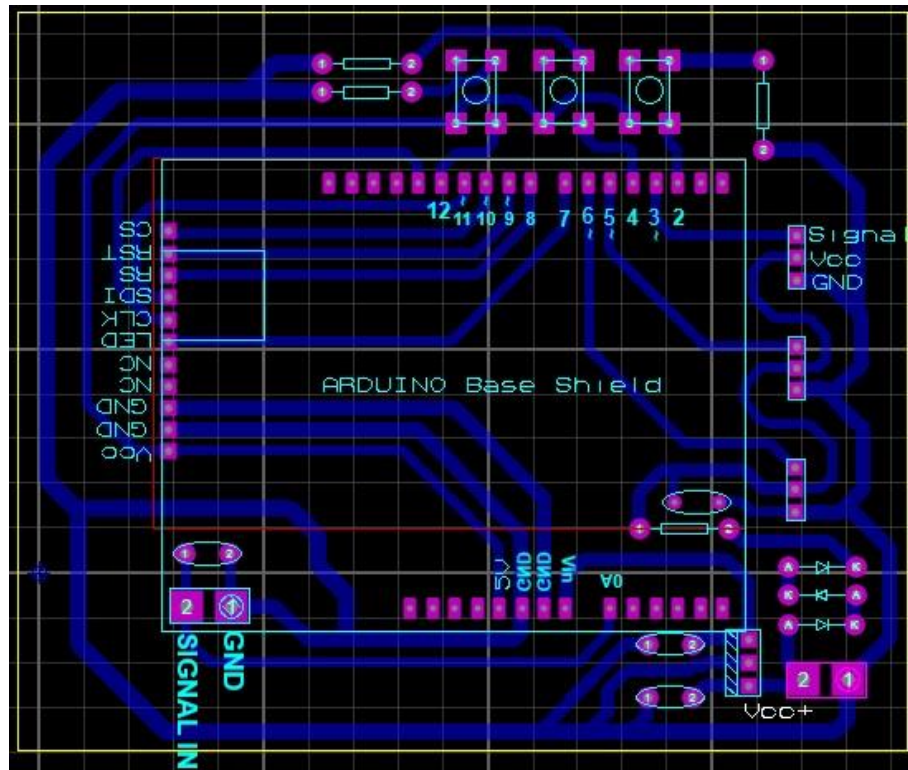


Figure 87: Arduino Shield for TFT, servos and buttons

4.5.3. The Power Supply

4.5.3.1. The LIPO battery

It's been bought this LIPO battery below to supply the servos and the Arduino.



Figure 88: 7.4V LIPO battery

Battery sizes:

7V battery:

43 x 30 x 5 mm

4.5.3.2. The symmetric power supply

It is needed a symmetric alimentation because the operational amplifiers needs it to work properly in order to achieve positive values as well as negatives.

The configuration of the batteries to make a symmetric alimentation is such as in the next figure.

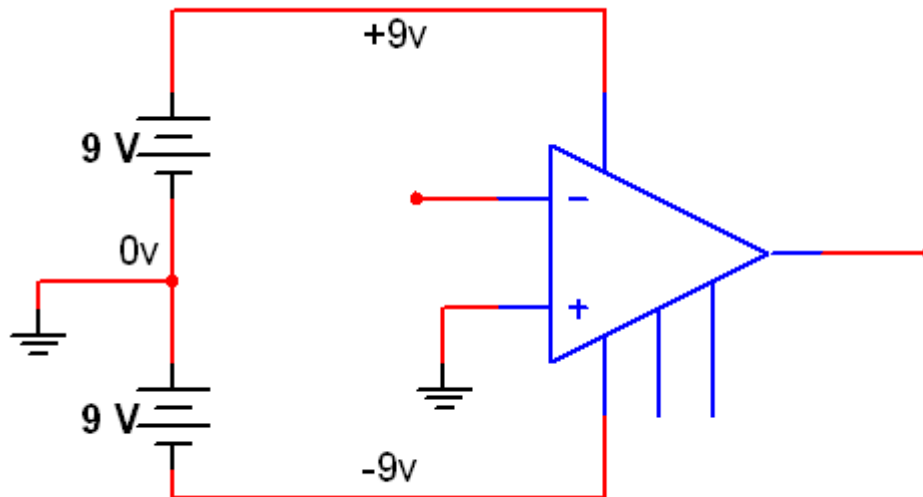


Figure 89: Scheme of the symmetric alimentation

It's been used the standard 9V batteries:



Figure 90: Standard 9V battery

Battery sizes:

9V battery: 26 x 48 x 17 mm

4.5.4. Manufacturing Muscle Sensor PCB

Once the layout has been finished, it is possible to obtain the photolith. This process consists on printing in a special transparent paper the copper sides of the circuit with a normal printer, better in the best quality. Next figure shows the obtained photolith.

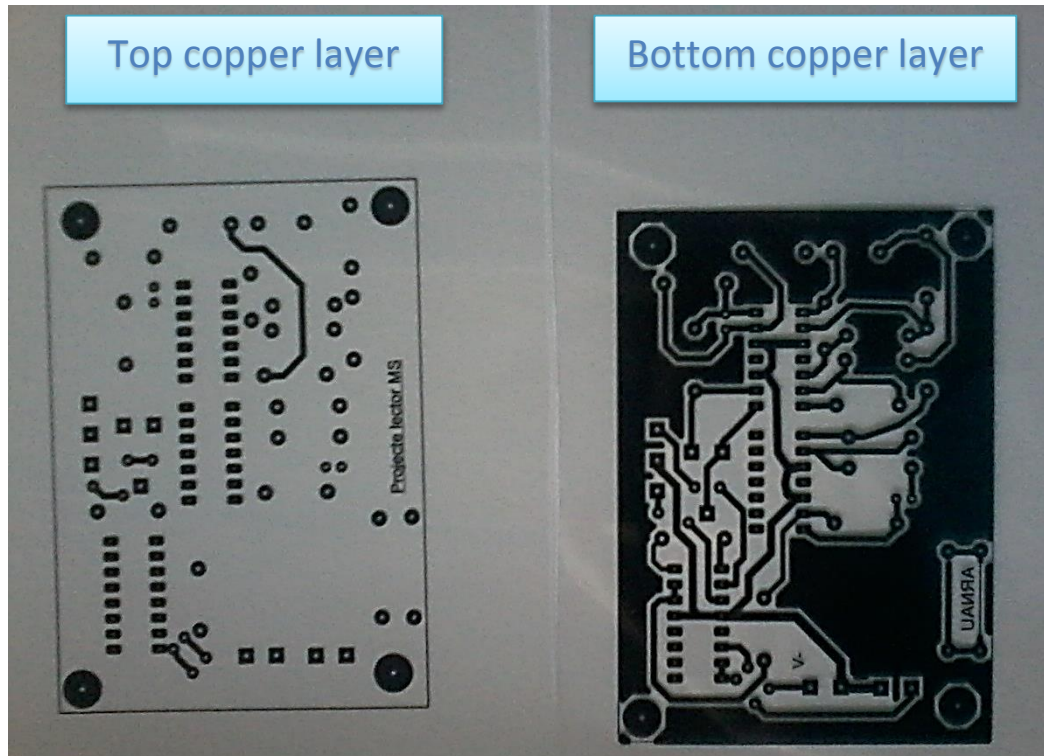


Figure 91: The obtained photolith

After align both sides perfectly, it's been made a “sandwich” with the copper plate between.



Figure 92: Copper plate in the insolation machine

It's been put in the insolation machine and left each side 80 seconds.



Figure 93: Insolation machine



Figure 94: Revelation liquid

Reveal until it gets completely contrasted

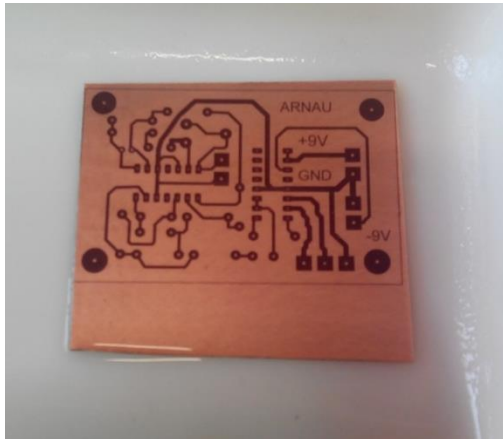


Figure 95: Board revealed



Figure 96: Revelator liquid

Attack the copper with the acid.

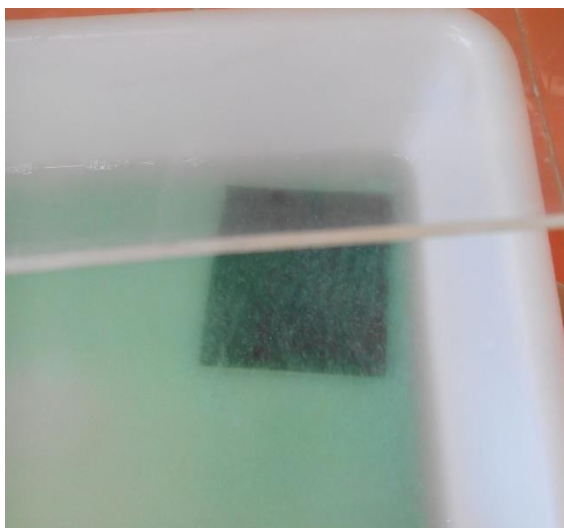


Figure 98: Acid attack



Figure 97: Acid product

It's important not to breathe the gases emitted from the reaction, so it's been put below this extractor:



Figure 99: Extractor

Realise the remaining copper disappears.

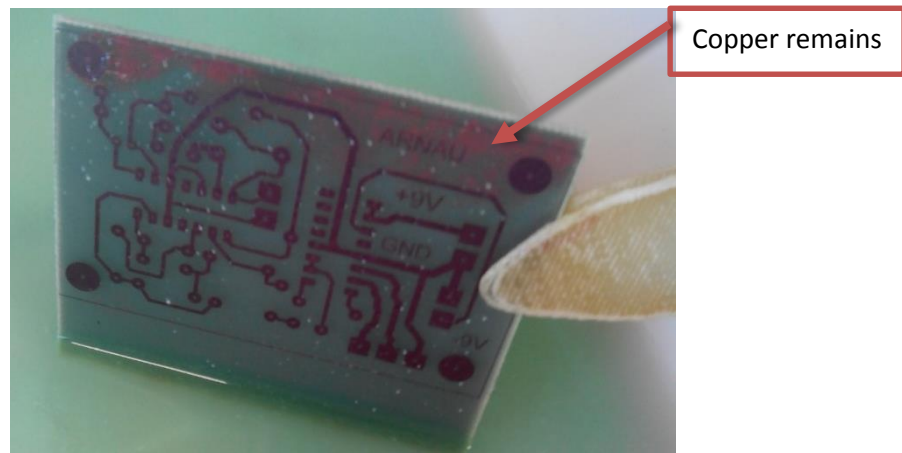


Figure 100: Copper remains

Make the holes with a 3mm drill

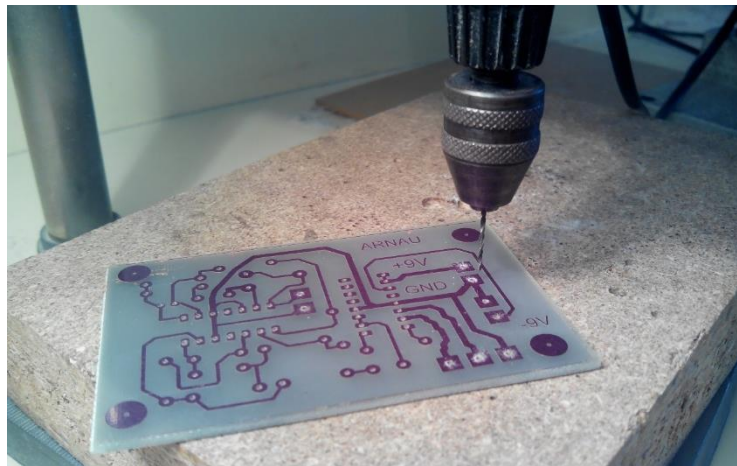


Figure 101: Pierce the board

Weld the chips, capacitors, resistances and sockets.

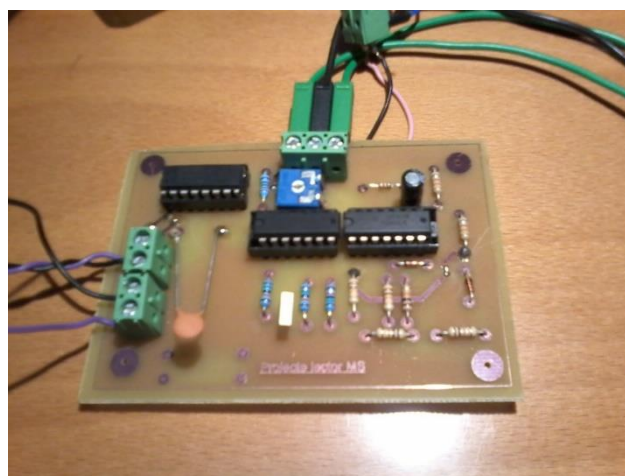


Figure 102: Final muscle sensor board

4.5.5. Manufacturing of the Arduino's Shield

It's been followed the same steps to reveal and hole the electronic board as in the previous chapter 4.5.4.

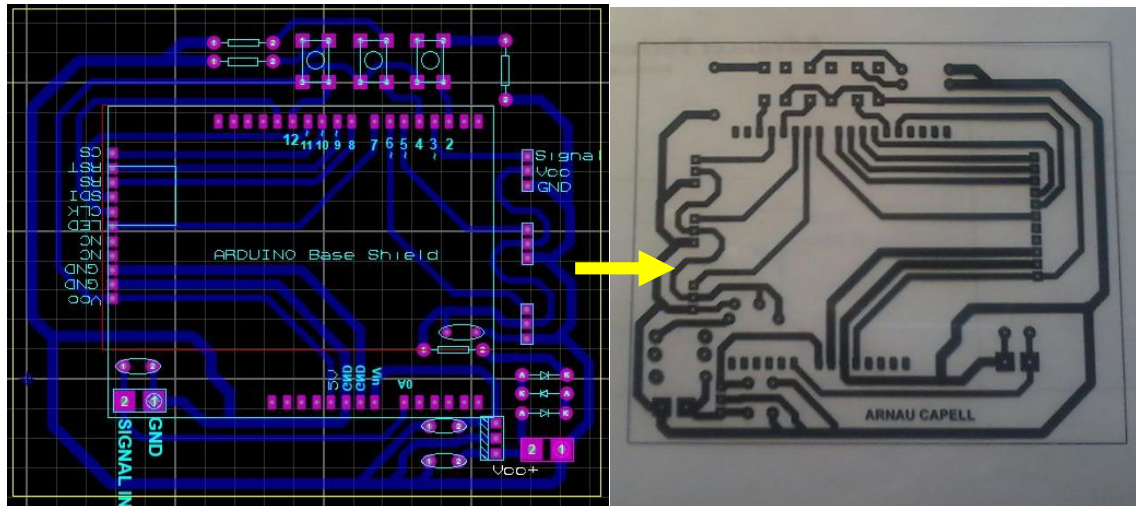


Figure 103: Shield design from computer to special paper

An image of the shield revealed:

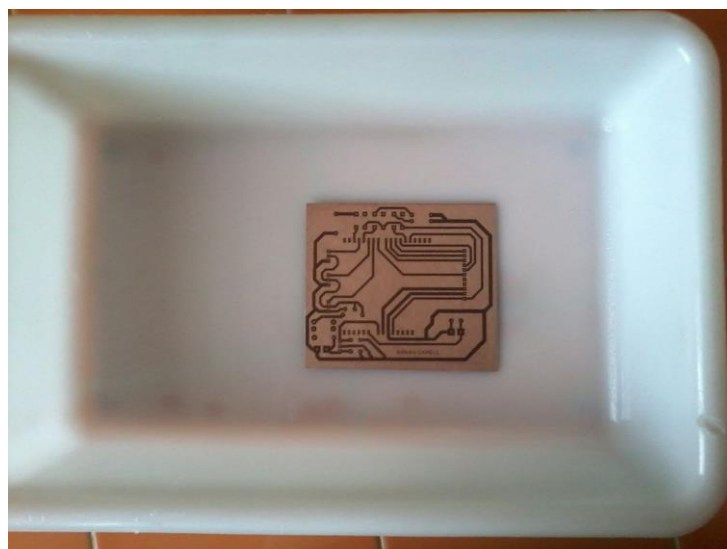


Figure 104: After revealing the shield

A pliers has been used to nail the different pins through the board.



Figure 106: Using pliers to nail the pins



Figure 105: Type of pins used

Image of the pins welded at the PCB.

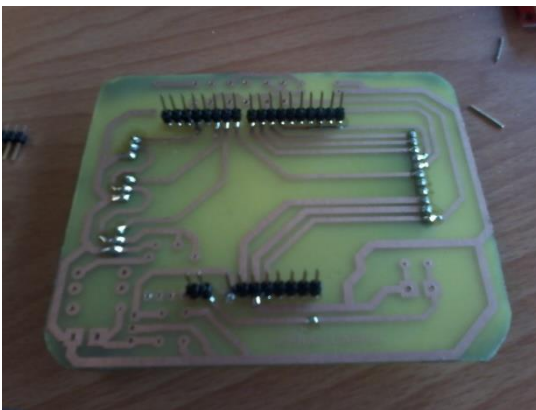


Figure 108: Welded pins bottom shield



Figure 107: Welded pins top shield

After welding all the components:



Figure 110: Final shield with screen plugged in

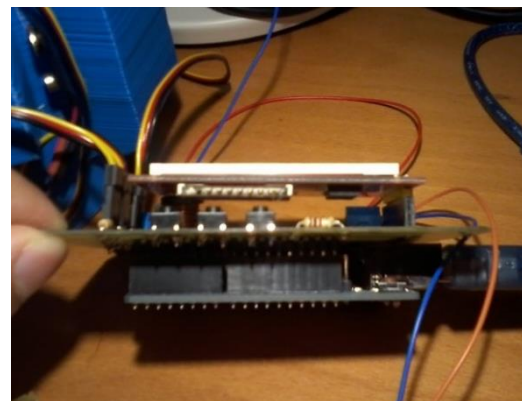


Figure 109: Side view of the shield with Arduino

4.5.6. The Final Assembly

Once obtained the PCB's they have been attached to the whole structure of the forearm with screws onto the respective supports.

First it's been put the Muscle sensor board:

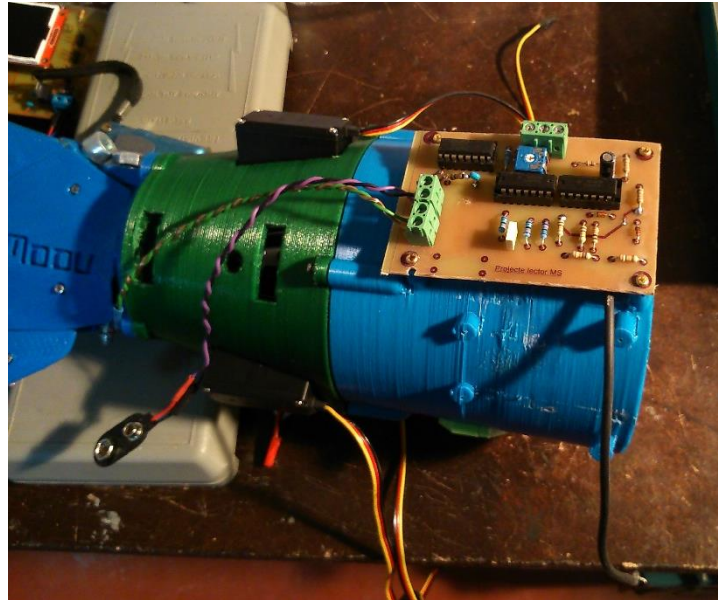


Figure 111: Muscle Sensor attached to forearm

And then the Arduino UNO:



Figure 112: Arduino attached to forearm

Finally added the shield onto Arduino board and it's obtained the final prototype of the prosthesis.

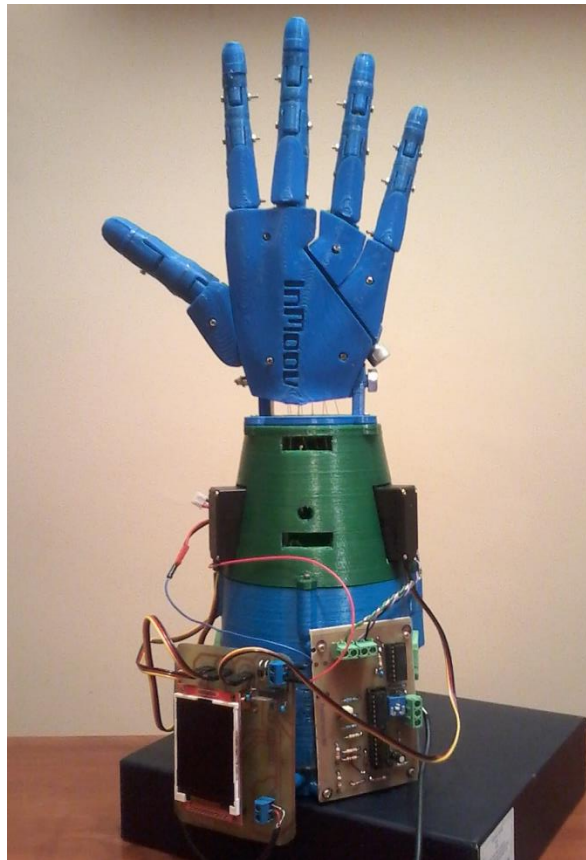


Figure 113: Final Assembly

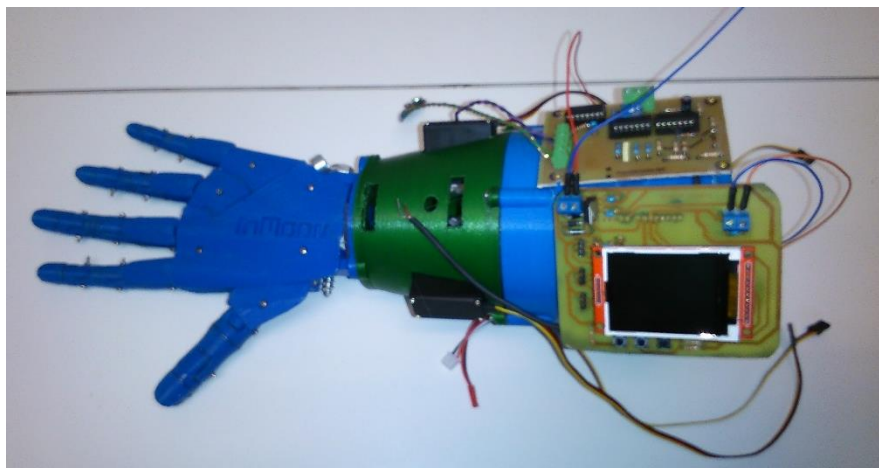


Figure 114: Prototype finally assembled

4.6. Experimental Result

It is needed to evaluate the output signal in order to make sure that the range of the voltages are between 0 - 5V because the analogical input of Arduino works between this values, also it is preferable to obtain a signal really smooth in order to make the servos work properly later.

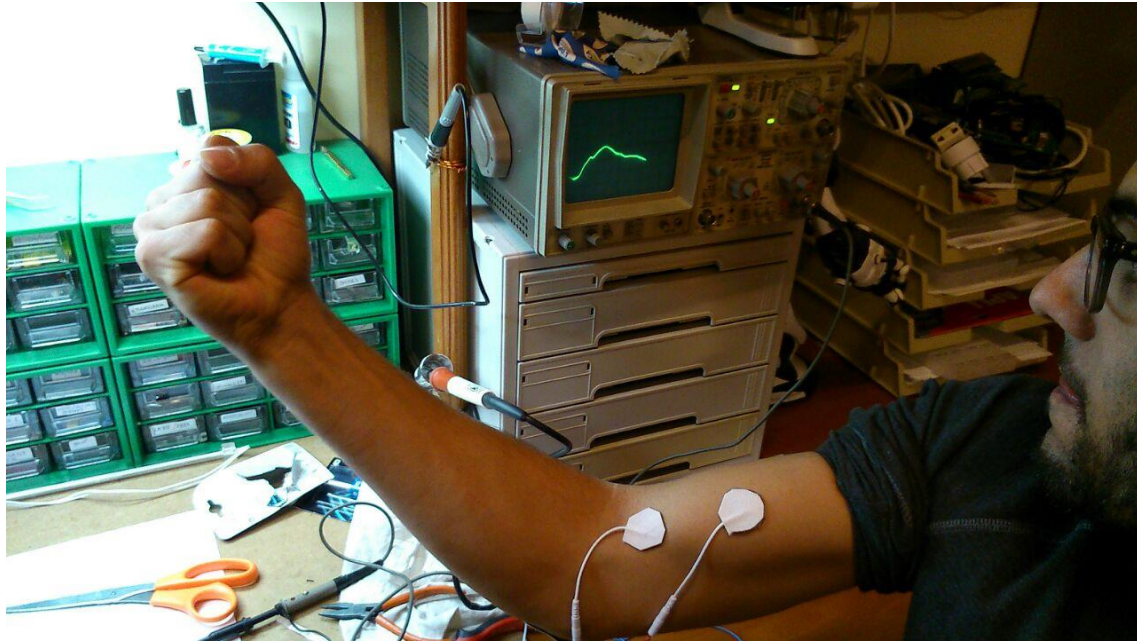


Figure 115: Signal output from the biceps

It has been connected the output to an oscilloscope and tested the values of the signal with different forces. To make the signal not exceed 5V, there is a variable resistance to change the final gain of the pulses.

Once the final signal has been well limited, it is the time to connect the Arduino and make a simple test with the servomotors.

The analogical to digital converter of Arduino UNO goes from [0, 5] V, as it has been said, to [0, 1023] bits. First of all, calibrate the minimum value to make the servos start moving. Then, after overcoming this value the servos will start moving from 0 to 179 degrees.

4.6.1. Signal Tests

Analysing the output signals from the different amplifiers.

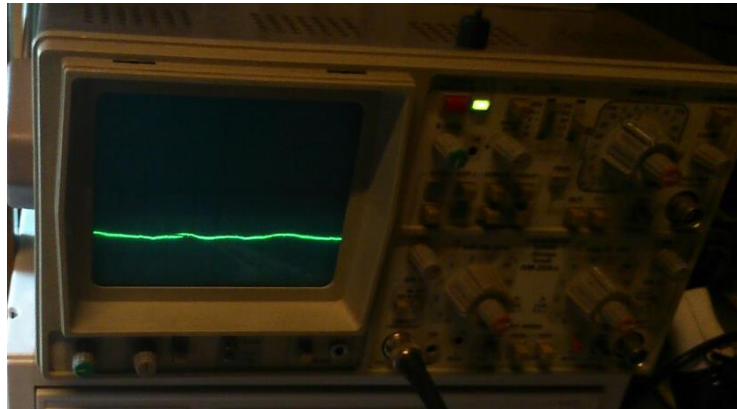


Figure 116: Output signal INA125P

Output signal from INA125P, it's observed that the signal is really small, scale of millivolts.

Signal obtained after amplified in TL084.

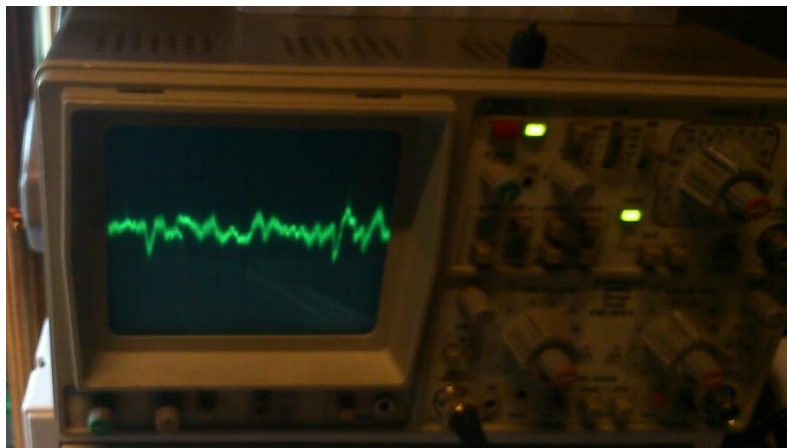


Figure 117: Signal obtained after amplify

It is shown that the signal after Amplify step is too much unstable but quite much bigger than the first one.

Finally the signal output from the Muscle Sensor is like that:

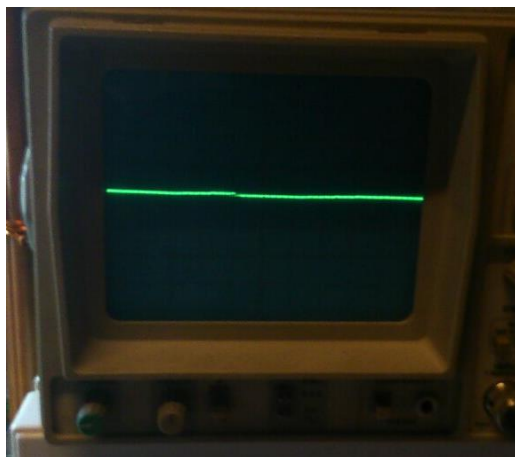


Figure 119: Signal without force

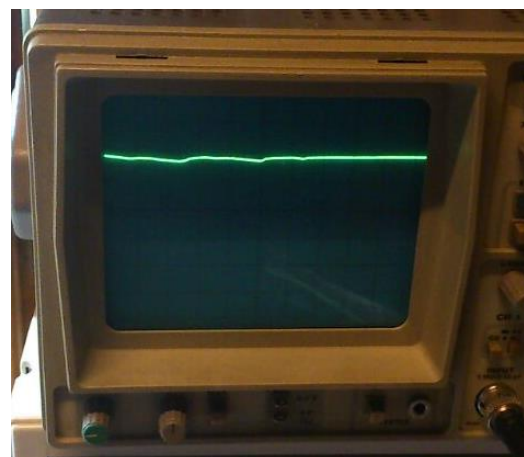


Figure 118: Signal with force

As the figures above show, in the figure 119 the signal is stable at 0V because there is no force done. Afterwards, it is applied some force and the signal rise up until more or less 5V really stable.

4.6.2. The Prototype performance

Once the signal is well processed it can be interpreted by Arduino's analogical input properly.

It's been written a program, as it can be found in Annex 2, to make the prototype act in function of the force done.

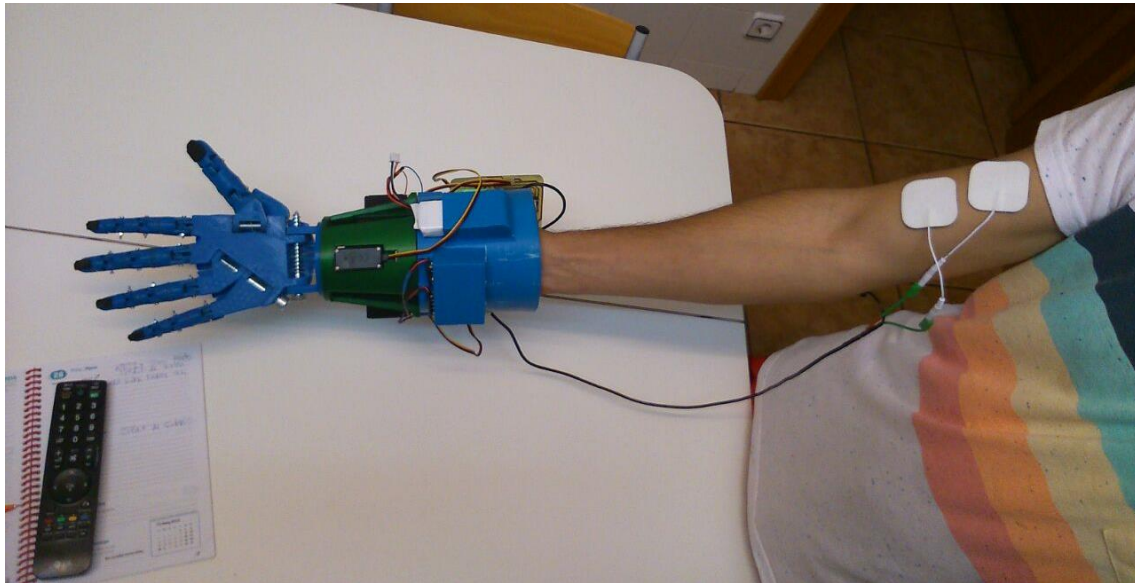


Figure 120: Prototype state without doing force

That figure shows the relax state of the prototype without doing any force.

And the next one demonstrate when force is applied the prosthetic hand shrinks.

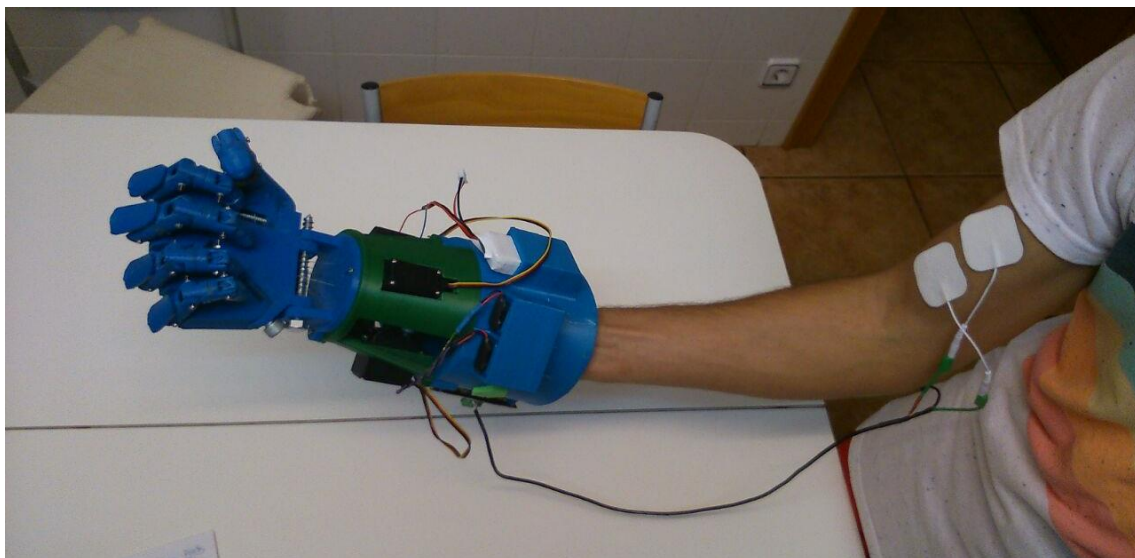


Figure 121: Prototype when force applied

5. Costs and Economic Aspects

In order to demonstrate this project is low price compared with other myoelectric prostheses it's been taken into account all the economic aspects.

5.1. 3D Printer prices

<i>Parts</i>	<i>Time (min.)</i>	<i>Price (€)</i>
<i>Auriculaire</i>	146	
<i>Coverfinger1</i>	154	
<i>Index3</i>	207	
<i>Majeure3</i>	335	
<i>Ringfinger3</i>	187	
<i>Thumb5</i>	304	
<i>Topsurface4</i>	393	
<i>WristlargeV4</i>	1300	
<i>WristsmallV3</i>	192	
Subtotal Hand	3218	30,34 €
Wrist	214	3,67
Servos inside	1110	11,7
Electronics	798	15
TOTAL		60,71 €

Table 3: 3D Printing prices

5.2. Other elements

Pack 100 screws M2x20	3, 51 €
Allen bolts	1, 56 €
TFT ILI9225	15, 59 €
Fishing line	3, 99 €
Arduino UNO	20 €
Pack 16 electrodes (4x4cm)	9, 99 €
7.4V LIPO Battery 1200 mAh	12, 8 €
PCB boards & components	16 € (approx.)
9V battery (x2)	7, 4€
TOTAL amount of the prototype:	151, 55 €



6. Conclusions

Making reference to the objectives reflected at the goals description, both types of movement have been achieved with the prototype, enclosure and impingement. As a prototype the final design hasn't been designed in order to be perfectly adapted to an amputated arm, it has been thought to be used by anyone such as a prosthesis tester and it accomplish the expectations.

During the project it's been overcome a lot of difficulties, beginning with the 3D printing, because it is not so easy as it seems, I've learnt that the 3D printers are so sensible to any minimal change of the temperatures and if the base or the extruder are not properly configured it is so easy to get an imperfect piece. Also, all the designs invented on this project has been changed several times because of some problems such as, a difficult design which is impossible to be printed with the 3D printer used due to its complex geometry, or because some physical properties weren't well designed before and the shape has had to be changed.

Otherwise, the electronics used are quite complex, the signal from the muscles is a really low signal and highly changeable, it has been amplified and smoothed in order to work with it properly, anyway, some troubles have appeared to read the signal, there so, the designs of the PCB has had to be modified also. It's been realized the complexity of the signal processing with the human body and I have to highlight the admiration to the people who works implementing the current prostheses which works in a similar way.

Stand out the improvement of my knowledge in 3D design with CREO Parametric and programming with Arduino Software, which is close to Dev. C++.

As an important point of this project, is been demonstrated the viability of self-making a prosthesis compared with the prices found on the Internet, the current project has cost around 150€ rounded, and the professional prosthesis are around 5.000-9.000€ and much more. Price impossible to afford for many people.

Another point of the conclusions is the improvement or optimization of the final prototype. Surely, it has a lot of aspects to be upgraded, but the most important for me are:

- Adding a pressure sensor to control the force of the grab when there is something.
- Improve the design of the forearm pieces in order to achieve a more natural and real shape.
- Try to add more servos to be able to move each finger with independence.
- Continuous improvements of the program and the signal reading.

Goals not as far as they seem.

To sum up, it's concluded that the myoelectric prostheses are so difficult to develop but nowadays is growing a new revolutionary idea such as the 3D printing which make easier and cheaper the self-created devices. In the next future these printers would be available and affordable for anyone and the open source designs on the Internet will make the crafting easier.

The future is in our hands.

7. Annexes

7.1. Annex 1: Advancer Technologies Muscle Sensor

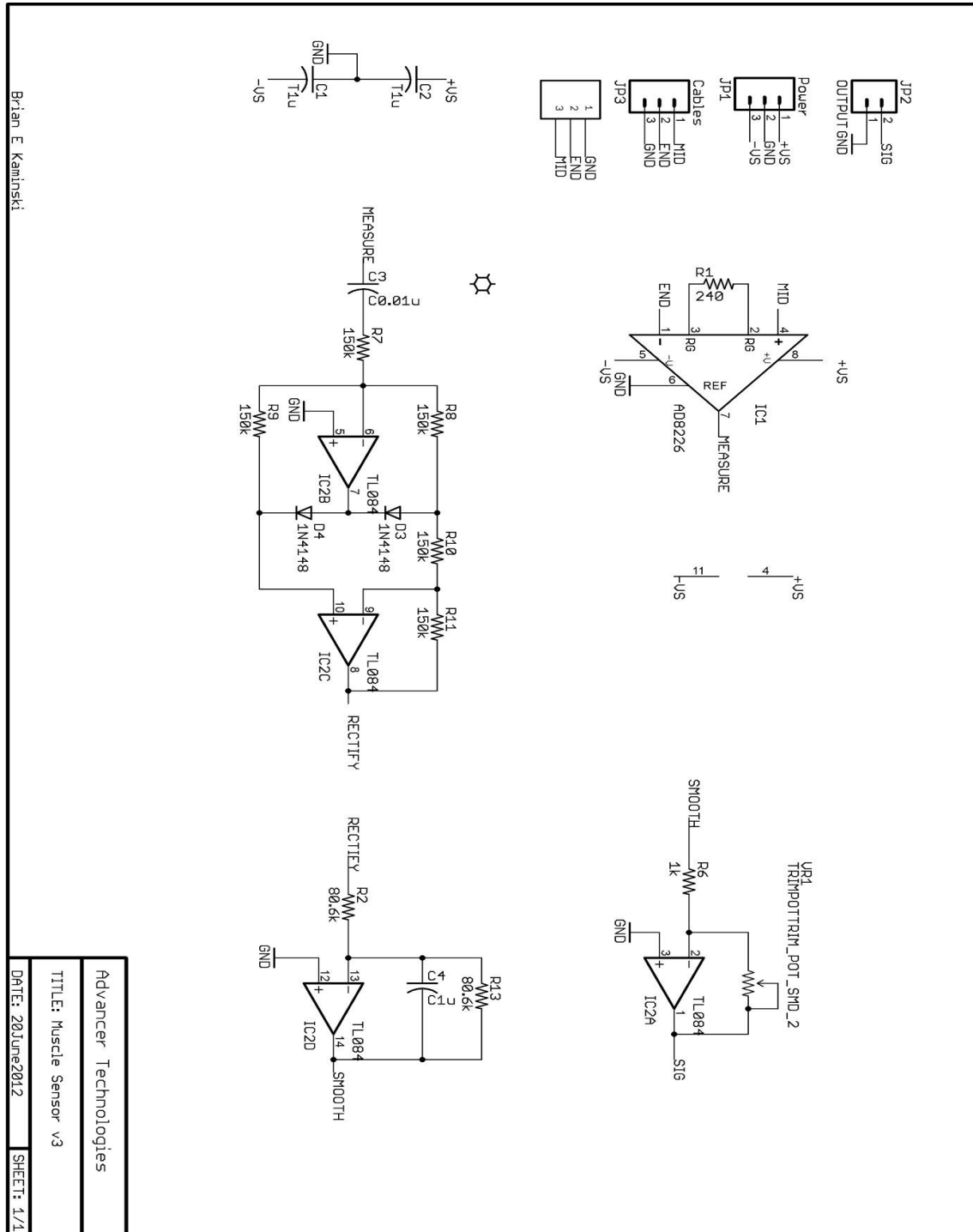


Figure 122: Advancer Technologies Muscle Sensor v3



7.2. Annex 2: Arduino code

```
/*Input voltages between 0 and 5 volts into integer values between 0 and 1023.
This yields a resolution between readings of: 5 volts / 1024 units or, .0049 volts (4.9 mV)
per unit. The input range and resolution can be changed using analogReference().
It takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading
rate is about 10,000 times a second.*/
#include <SPI.h>
#include <TFT_22_ILI9225.h>
#define TFT_RST 9
#define TFT_RS 8
#define TFT_CS 10 // SS
#define TFT_SDI 11 // MOSI
#define TFT_CLK 13 // SCK
#define TFT_LED 7 // 0 if wired to +5V directly
#include <Servo.h>
// Use hardware SPI (faster - on Uno: 13-SCK, 12-MISO, 11-MOSI)
TFT_22_ILI9225 tft = TFT_22_ILI9225(TFT_RST, TFT_RS, TFT_CS, TFT_LED);
int analogPin = 0; // select the input pin for analog values
Servo index;
Servo pulgar;
Servo petit;
int val; // variable to store the value coming from the sensor

byte alfa;
int temps;
byte minim;
int maxim;
byte start;
byte calibrate=0;
byte ini_x, ini_y, max_x, max_y ;
byte y;
byte t;
byte linea;
byte borra;
int valor, valor1, valor2;
byte matriu1[220], matriu2[220], matriu3[220], matriu4[220];
boolean flag = false;

void setup() {
  tft.begin();
  Serial.begin(9600); // setup serial
  index.attach(3);
  petit.attach(6);
  pulgar.attach(5);
  pinMode(2,INPUT);
  pinMode(4,INPUT);
  pinMode(12,INPUT);
  t = 0;
  borra = 0;
  ini_x = 0;
  max_x = 220;
  ini_y = 0;
  max_y = 104;
  start=0;
  tft.setOrientation(1);
```



```
}

void loop() {

    limits();
    if(start==0){oscilloscope();}
    movement();

}

void limits(){
    if(calibrate==0){
        while(digitalRead(2)==0){
            //write to TFT to push the appropriate button

            tft.setFont(Terminal11x16);
            tft.drawText(ini_x+ 20, ini_y +10, "PUSH BUTTON 1", COLOR_GREEN);
        }

        if(digitalRead(2)==1){
            do{
                minim=analogRead(analogPin);
                tft.drawText(ini_x+ 20, ini_y +30, String(minim), COLOR_GREEN);
                delay(100);
                tft.fillRect(ini_x, ini_y + 30, ini_x + 100 , ini_y + 80 , COLOR_BLACK);
            }while(digitalRead(2)==1);
            tft.clear();
        }
        while(digitalRead(4)==0){
            //write at TFT to push the second button
            tft.setFont(Terminal11x16);
            tft.drawText(ini_x+ 20, ini_y +10, "PUSH BUTTON 2", COLOR_GREEN);
            tft.setFont(Terminal6x8);
            tft.drawText(ini_x+ 20, ini_y +100, "Min:", COLOR_GREEN);
            tft.drawText(ini_x+ 50, ini_y+ 100,String(minim), COLOR_GREEN);

        }

        if(digitalRead(4)==1){
            do{
                maxim=analogRead(analogPin);
                tft.setFont(Terminal11x16);
                tft.drawText(ini_x+ 20, ini_y +30, String(maxim), COLOR_GREEN);
                delay(100);
                tft.fillRect(ini_x, ini_y + 30, ini_x + 100 , ini_y + 80 , COLOR_BLACK);
            }while(digitalRead(4)==1);
            tft.setFont(Terminal6x8);
            tft.drawText(ini_x+ 20, ini_y +100, "Min:", COLOR_GREEN);
            tft.drawText(ini_x+ 50, ini_y+ 100,String(minim), COLOR_GREEN);
            tft.setFont(Terminal6x8);
            tft.drawText(ini_x+ 20, ini_y +100, "MAX:", COLOR_GREEN);
            tft.drawText(ini_x+ 50, ini_y+ 100,String(maxim), COLOR_GREEN);
            delay(3000);
            tft.clear();
            calibrate=1;
        }
    }
}
```



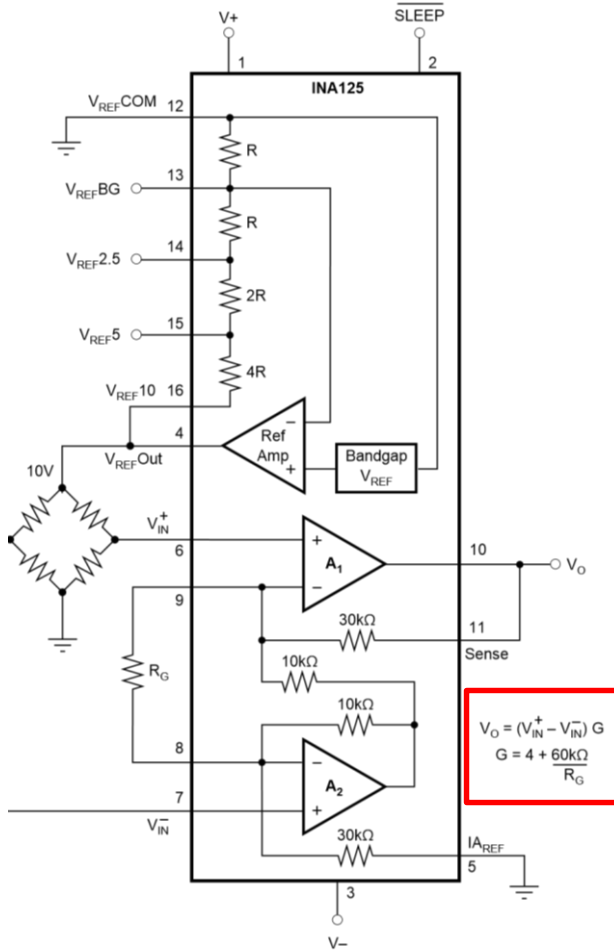
```
}  
void movement(){  
  // read the value from the sensor:  
  val = analogRead(analogPin); // read the input analog pin  
  
  if(val<=minim+100){  
  
    pulgar.write(10);  
    delay(250);  
    index.write(170);  
    petit.write(170);  
    pantalla();  
  
  }  
  if(val>minim+150){  
    //When the minimum value is exceed make the servos turn  
    for(alfa=170;alfa<50;alfa--){  
      val = analogRead(analogPin);  
      temps=map(val,100,600,400,50);  
      index.write(alfa);  
      petit.write(alfa);  
      delay(temps);  
  
      if(val<=minim+100){break;}  
    }  
  
    while(val>minim+100){  
      pulgar.write(179);  
      index.write(50);  
      petit.write(50);  
      val = analogRead(analogPin);  
      pantalla();  
    }  
  }  
  else{  
    pantalla();  
  }  
}  
  
void pantalla(){  
  for (int i = 0; i < max_x; i++) { // Dibuixar ONA  
    valor = analogRead(A0); // Llegir valor Analog  
    valor1 = max_y - (valor / 10);  
    matriu3[i] = valor1; // Fiquem el valor1 en la Matriu  
    //tft.drawText(1, max_y + 10, String(valor), COLOR_GREEN); // Escriu valor analogic  
    delay(1); // designa el temps de mostreig entre dues captures  
    t++; // increment de TEMPS eix X  
    valor2 = max_y - (analogRead(A0) / 10);  
    matriu4[i] = valor2; // Fiquem el valor2 en la Matriu  
    tft.drawLine(i, matriu1[i], t, matriu2[i], COLOR_BLACK); // BORRAR primerament  
    tft.drawLine(i, valor1, t, valor2, COLOR_GREEN); // Dibuixa LINEA Oscil·loscopi entre dos coordenades  
    //tft.drawText(1, max_y + 10, String(valor), COLOR_GREEN);  
  }  
  t = 0;  
  for (int i = 0; i < max_x; i++) { // MATRIUS per memoritzar els valors anteriors i poder borrar la pantalla  
    matriu1[i] = matriu3[i];  
    matriu2[i] = matriu4[i];
```



```
}  
}  
void oscilloscope(){  
    tft.setOrientation(1); // Pantalla en HORIZONTAL  
    tft.drawLine(ini_x, ini_y, max_x, ini_y, COLOR_YELLOW); // Dibuixa linea recta superior  
    tft.drawLine(ini_x, ini_y + 1, max_x, ini_y+1, COLOR_YELLOW);  
    tft.drawLine(ini_x, max_y + 1, max_x, max_y + 1, COLOR_YELLOW); // Dibuixa rectes inferiors  
    tft.drawLine(ini_x, max_y + 2, max_x, max_y + 2, COLOR_YELLOW);  
    tft.drawLine(ini_x, max_y + 30, max_x, max_y + 30, COLOR_YELLOW); // Dibuixa linees sota de lletres  
    tft.drawLine(ini_x, max_y + 31, max_x, max_y + 31, COLOR_YELLOW);  
    tft.setFont(Terminal11x16);  
    tft.drawText(ini_x + 50, max_y + 10, "OSCILOSCOPE", COLOR_GREEN); // Dibuixa text inferior  
    tft.setFont(Terminal6x8);  
    start=1;  
}
```

7.3. Annex 3: Gain estimation of the Muscle Sensor

INA125P:



In the INA125P datasheet it's been specified the equation to calculate the Gain.

$$G = 4 + \frac{60k\Omega}{R_G}$$

$$R_G = 261 \Omega$$

$$G = 4 + \frac{60k\Omega}{R_G} \cong 234$$

Figure 123: INA125P Datasheet scheme and gain equation

TL084CN:

$$G_1 = \frac{R_4}{R_3} = \frac{150k}{10k} = 15$$

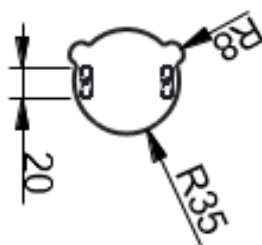
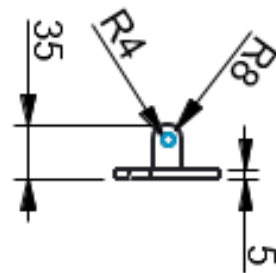
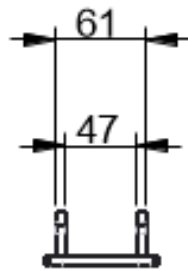
$$G_2 = \frac{R_6}{R_5} = \frac{150k}{150k} = 1$$

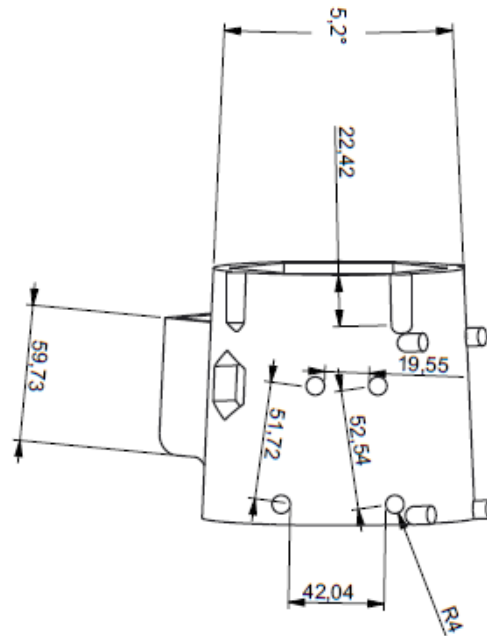
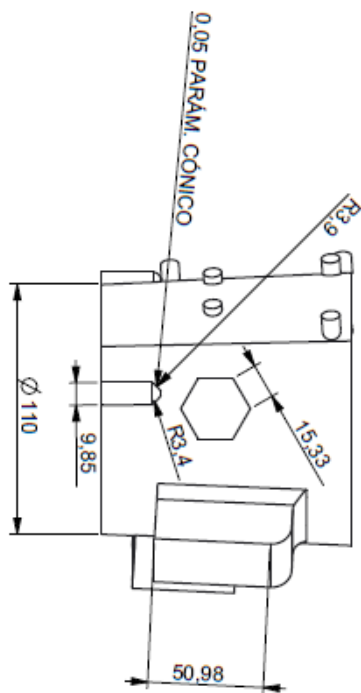
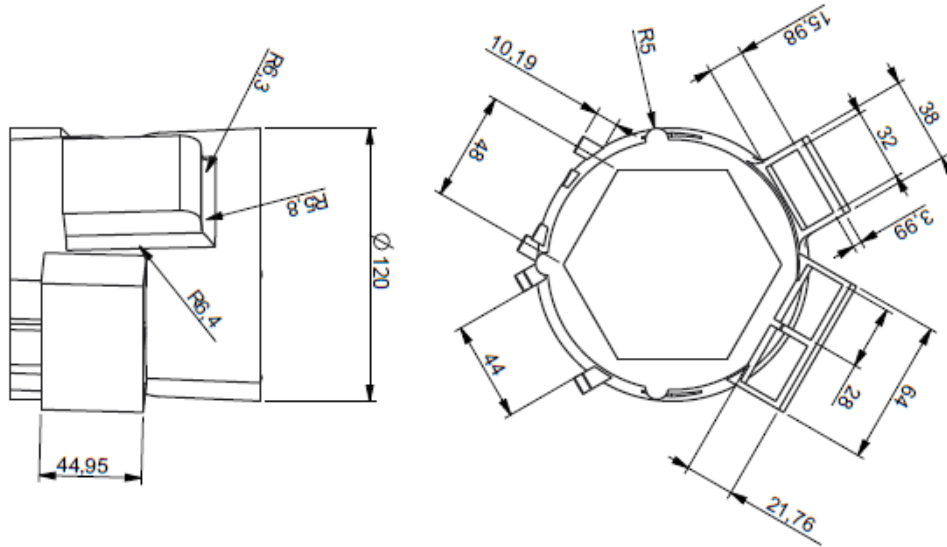
$$G_{var} = \frac{R_{v1}}{R_{14}} = \frac{22k}{1k} = 22(max.)$$

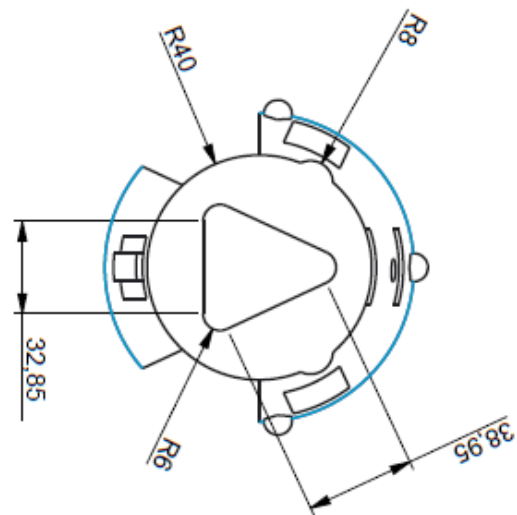
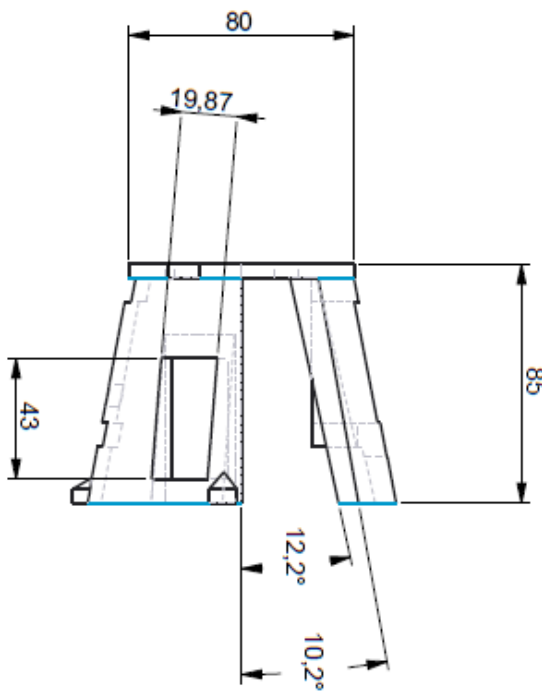
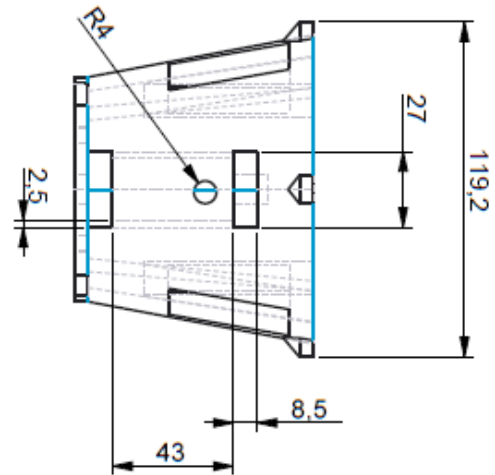
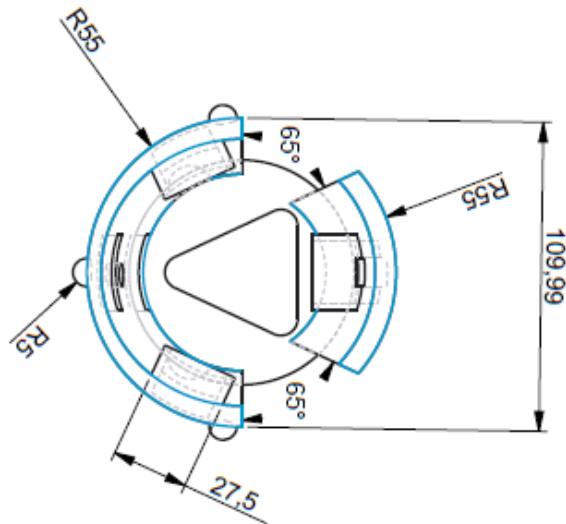
$$G_{total} = G \cdot G_1 \cdot G_2 \cdot G_{var} = \begin{cases} G_{max} = 77220 \\ G_{min} = 3510 \end{cases}$$

7.4. Annex 4: Designed pieces drawings

All the heights are in millimetres [mm].









8. Bibliography

Arduino. (2013). At *Taller Arduino*. Retrieved May 9th 2015 from:
<http://tallerarduino.com/tag/shield/>

Arduino (2015). At *Arduino Home*. Retrieved May 15th from:
http://store.arduino.cc/index.php?main_page=product_info&cPath=11&products_id=195

Camacho-Conchucos, H.T. (2010). Patients amputated by work accidents: characteristics and years accumulated of potential productive life lost. Retrieved March 4th, 2015 from:
<http://www.scielo.org.pe/pdf/afm/v71n4/a11v71n4.pdf>

Fishing line (2015). At *Decathlon*. Retrieved May 10th 2015 from:
http://www.decathlon.es/fully-300-m-id_8240217.html

Dorador González, J.M. (2004). *ROBÓTICA Y PRÓTESIS INTELIGENTES*. Retrieved March 15th 2015 from:
<http://www.revista.unam.mx/vol.6/num1/art01/art01-2d.htm>

Electronics. (2015). At *Wikipedia*. Retrieved May 9th 2015 from:
http://en.wikipedia.org/wiki/Pulse-width_modulation

Electronica. (2012). At *Ravc_cs*. Retrieved May 10th 2015 from:
<http://ravc00cs.blogspot.com.es/2012/07/pwm.html>

Electronics. (2015). At *ElectronicsTutorials*. Retrieved May 15th 2015 from:
http://www.electronics-tutorials.ws/opamp/opamp_5.html

Electronics (2013). At *Advancer technologies*. Retrieved February 17th 2015 from:
<http://www.advancertechnologies.com/p/muscle-sensor-v3.html>

Instrumentation amplifier (2015). At *Texas Instruments*. Retrieved May 19th 2015 from:
<http://www.ti.com/lit/ds/symlink/ina125.pdf>

López Sullaez, L.C., and Estrada Ruiz, R (2009). *Impact Occupational of Traumatic Amputation in Fingers of the Hand by Work Accident*. Retrieved March 6th, 2015 from:
<http://scielo.isciii.es/pdf/mesetra/v55n217/original4.pdf>

Operational amplifiers (2015). At *Texas Instruments*. Retrieved May 19th 2015 from:
<http://www.ti.com/lit/ds/slos081h/slos081h.pdf>

Prusa Mendel. (2015). At *RepRapWiki*. Retrieved May 10th 2015 from:
http://reprap.org/wiki/Prusa_i3

Robotics. (2012). At *Inmoov*. Retrieved January 8th 2015 from:
www.inmoov.fr/download/

Screen (2015). At *ebay*. Retrieved May 5th 2015 from:
<http://www.ebay.es/itm/2-2-IL9225-SPI-TFT-LCD-Display-mit-Arduino-Library-C51-STM32-CP11019-J53-/281353505156>