ELEX – Entwicklung eines Ellenbogens-Exoskeletts

Development of an elbow's exoskeleton

Handbook and Tutorial

English

Prof. Dr. Hans-Georg Enkler

Markus Scherzinger
Illya Gents
Gerardo Garcia

Source of BEYOND ROBOTICS

1. Index

2. Preface

The robotic Exoskeleton, a different way of learning robotics by ourselves. We as bachelor students from HFU have taken the goal to improve this device so it could help others' experience and achieve a safer way of operation. This work is directed mostly to students that want to make a change in technologies as we did.

Our work covers different fields which involve the use of different software. Before any use, we highly recommend previous knowledge of CAD software and Arduino programming. Even if you don't know much about these areas, a bit of internet searching and video tutorials will help you get through this interesting world of optimization. Through our experience, we can assure you, that your skills in these areas will develop more than you think. Since we just had three months of work, we could not solve every issue and enhance a lot from the Exoskeleton, nevertheless we invite more ambitious students to join us in the search of knowledge and new fun ways of learning. We are hoping for your new ideas to come by and reinforce our project.

As a part of the job, the customized parts will not be available to download but only the printable version of them. As for the Arduino code, some parts of the program will be placed in this handbook only as a reference. We made this for legal restrictions and personal agreements with the creators of this gadget. Either way, don't worry about trying to do all the work by yourself, since each chapter of the book will describe our job in the different areas we had covered through our job. We will try to go as deep as possible so that any possible doubt will be solved by consulting this handbook.

We appreciate the support of the EduExo team as for this work would not have been possible with their help. We also want to thank the HFU University to give us this task and try to make a change to help others. This project wouldn’t have been possible without your help.

We hope you enjoy this new adventure as much as we did and that this boosts your hunger for knowledge, investigation and studies.

3. Teamworkers

Prof. Dr. Hans-Georg Enkler

HochschuleFurtwangen

Leader of the project

Markus Scherzinger

HochschuleFurtwangen

Wirtschaftsingenieurwesen - Product Engineering

In charge of CAD area, development, construction and execution.

Illya Gents

Hochschule Furtwangen

Wirtschaftsingenieurwesen - Product Engineering

In charge of the area of installation of the electrical parts as well as helping in the Arduino programming.

Gerardo Garcia

Tecnologico de Monterrey - Bachelor of Science in Mechanics with minor in Industrial Engineering

In charge of the Arduino programming Area.

4. Introduction

After deepening in our different areas and preparing ourselves to the adventures of ELEX, we made extraordinary discoveries. We achieved to enhance our knowledge through internet surfing and using daily entertainment tools like YouTube.

Starting from this point, we designed the ELEX from scratch and tried optimized it to the point in which our arm looks totally different than the beginning. We remade the whole system and assembled it so that it would work even better than the original one.

This optimization includes CAD work, coding and a new setup for the electronic parts.

The CAD field included a deep optimization of both aesthetic and functionality improvement. Now, the arm looks way more advanced and technological and is not so hard to build.

The CAD and electronic parts go along, because with this new setup, the cables are not exposed and are now safe.

As for the coding, we had a lot of problems, mainly due to the lack of time and the expertise level of the project. Either way, we achieved to ameliorate the code and introduced it to the new electronic parts added to the arm.

An interesting part of this project is that we encouraged ourselves to take an extra step and we went far beyond our expectations. Although we couldn't achieve our main goal, we decided our achievement was enough for just three students.

5. Logo

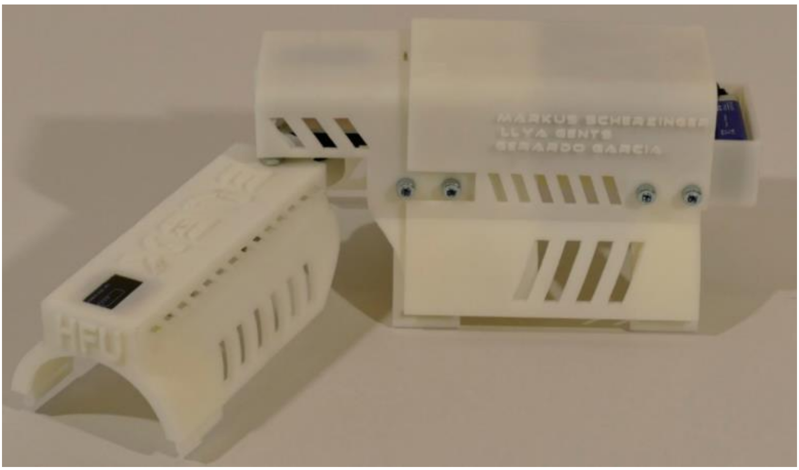
ELEX means the development of an elbow's exoskeleton. This logo has been developed as a task from the university in contact with Volker Bartenbach from the company EDUEXO. The writing type was created with the font ETHNOCENTRIC and the color green, in order to relate it with the logo of the Hochschule Furtwangen University. For the letter L, a built exoskeleton from the company EDUEXO was taken as a reference to the ELEX project. In order to create branding, a small part of this project was to develop our own logo and print it to the CAD components.

6. Mechanical parts

Summary

In this part, the optimized parts and elements are explained. One of the main tasks was to enhance the existing parts made by the company EDUEXO. Optimizations have been made for all components in order to benefit from all advantages. As read in the specifications, the cables were hidden through a cable duct and redesign an arm that could be more protected, so that the electronics are also safe against splashes and bumps. As well, the idea of our side was to be able to adjust the arm in 4 different sizes, allowing to adapt it to different sizes. Another idea from us was to install a screen on the forearm to display the battery level.

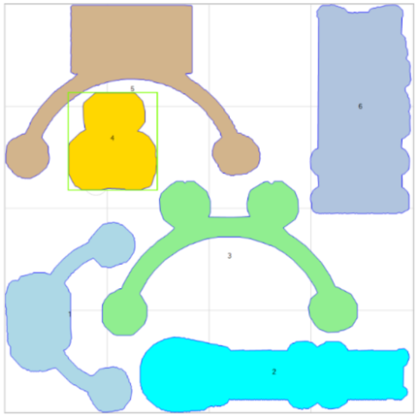
The original mechanism of the arm was designed by the company EDUEXO out of 6 parts. After optimizing the individual components and developing the protective arm, the whole assembly consists of 12 components.



ill.1 Finish elbow´s exoskeleton

6.1 3D Printer

ill.2 3D Printer



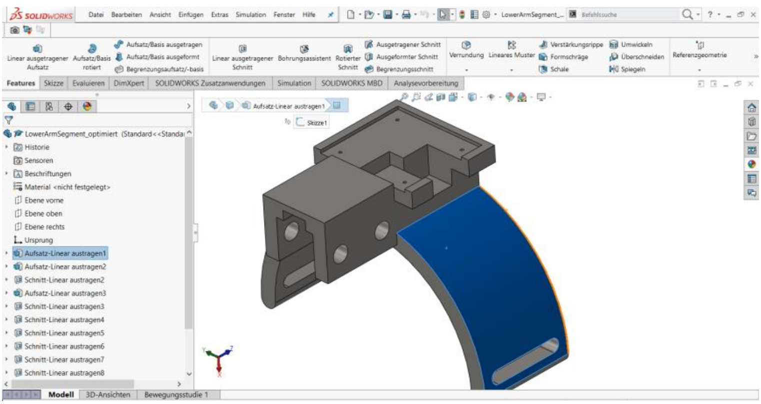
ill.3 3D Printparts

The components were printed through the 3D printer shown above, from the company Alphacam Type Dimension bst 768. An advantage for the project was to use the university's own 3D printer. Next, the components were analyzed to be optimized and drawn with the design program Solid Works. After all components have been saved as STL files, they were sent to the 3D printer.

6.2 CAD Program

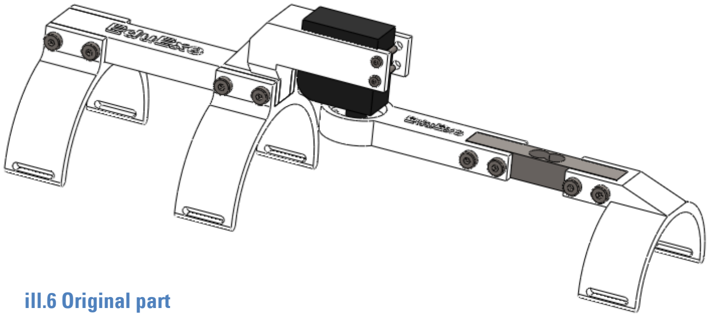
ill.4 Solid Works logo

Through this software, all of the parts were designed and optimized. As a student of the Hochschule Furtwangen University, we receive a license for this program. First, all the original parts from EDUEXO were measured and designed in SolidWorks 1:1. In order to make optimizations, an analysis of the original parts was made, test their function and test the feasibility of possible optimizations. Then the construction of the new components could be started. The Solid Works program offers features to add nuts and bolts so you can do everything in detail. After all components were drawn, a tree group was created and, afterwards, all the parts were joined together.

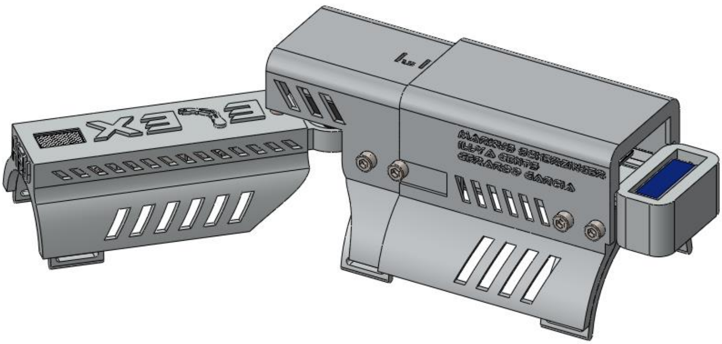


ill.5 Solid Works example

6.3 Original part vs optimized part

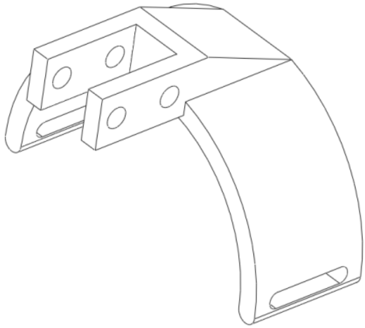


ill.6 Original part

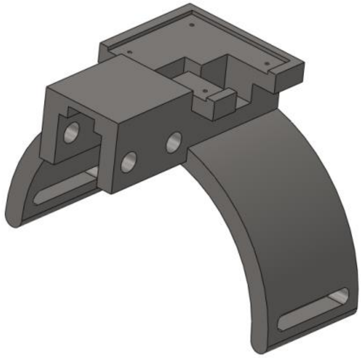


ill.7 Optimized part

6.4 Part 1 -LowerArmSegment



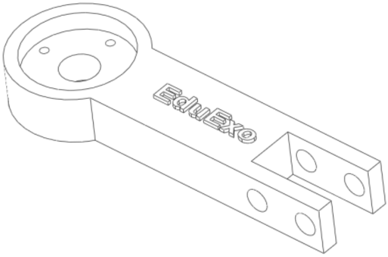
ill.9 Lower Arm Segment original



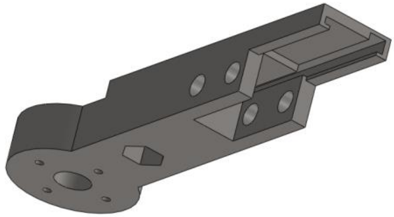
ill.8 Lower Arm Segment optimized

It was really important, in this part, to widen the radius of the arc to ensure a better and more comfortable grip for the forearm. The holes at the end of the Velcro strip attached were offset a few millimetres above to have more material for a possible tensile load. In order to mount a small screen on this component, a bracket was also built, so the screen can be attached by 4 screws. The screen lies in a frame designed with the exact dimensions to ensure a better grip. In order to protect the cables of the screen, a small cable duct was constructed underneath the bracket of the screen, as well as a type of lid.

6.5 Part 2 -Interface Motor Sensor



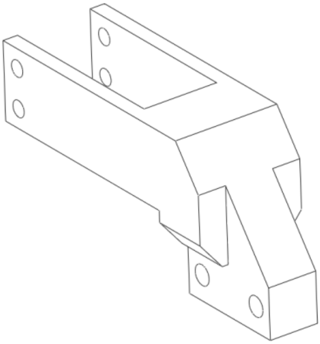
ill.11 Interface Motor Sensor original



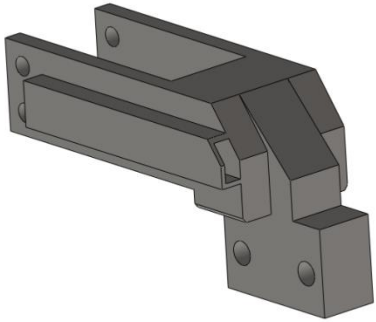
ill.10 Interface Motor Sensor optimized

Part 2, or Interface Motor Sensor, connects the sensor with Part 1 and the other one with the motor. To protect the sensor and the cables, an extra lid was designed over the sensor. In addition of this, the channels in this lid serve also as cable channels. To mount the protective arm, the entire component Interface Motor Sensor has been moved up a few millimeters. Therefore, the protective arm rests on this component and can be installed with a M6 screw, in which the nut can be pressed with the component, as seen on the picture. There was no logo added to this part (see original part), due to the lack of view with the mounted protective arm mounted over it.

6.6 Part 3 - Motor Adapter



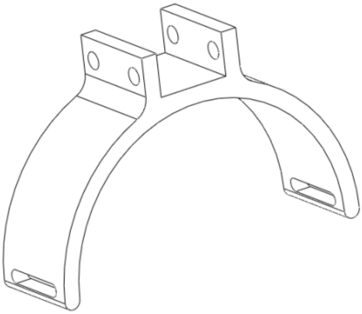
ill.12 Motor Adapter original



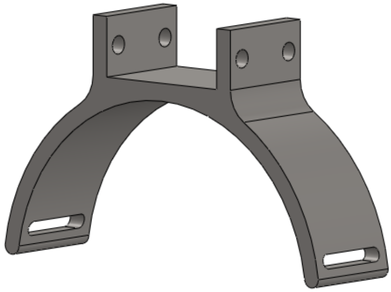
ill.13 Motor Adapter optimized

The motor is attached to this part, the motor adapter, and added to the part 4 (see below). To protect the cables an external cable duct was designed and installed. The cables are only routed from part 1 to part 2 and then to part 3 ending with the board. With this part, only few things were added to achieve the required optimization. The slopes and length have been optimized to make the part fit the next one. Since the entire assembly was reinforced, the lower part was widened of the motor adapter.

6.7 Part 4 - Upper Arm Cuff1

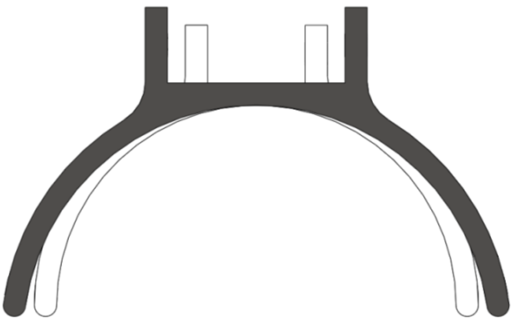


ill.15 Upper Arm Cuff 1 original



ill.14 Upper Arm Cuff 1 optimized

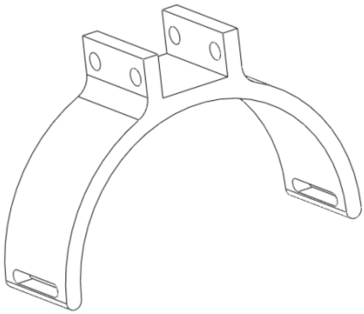
The Upper Arm Cuff1 was made with the same radius as the Lower Arm Segment to provide better grip on the arm. In addition, the bracket for part 5 has been widened to make it more stable.



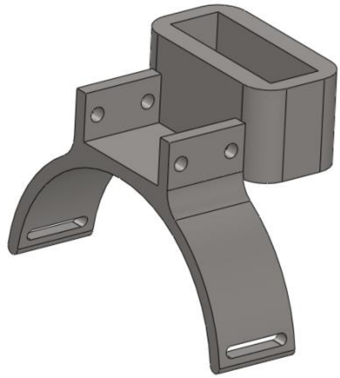
ill.16 Difference from original to optimized parts

Before the radius was optimized, we had the chance to measure the arm of different people so we could calculate a mean. Adding an elastic hook and loop fastener makes the attachment to the arm adjustable.

6.8 Part 5 - Upper Arm Cuff2



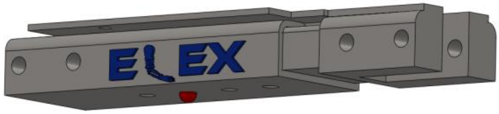
ill.18 Upper Arm Cuff 2 original



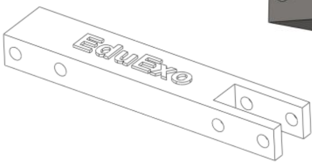
ill.17 Upper Arm Cuff 2 optimized

As it was already explained for part 4, the radius of this part has also been extended to ensure a better grip. After optimizing, this part was made to protect and hiding the battery too. At the back of this component, a bracket was specially designed for a 9V battery. The 9Vbattery is now in and attached through a hook and loop fastener. Finally, to attach this component to the forearm, an elastic hook and loop strip was sewn.

6.9 Part 6 - Upper Arm Segment

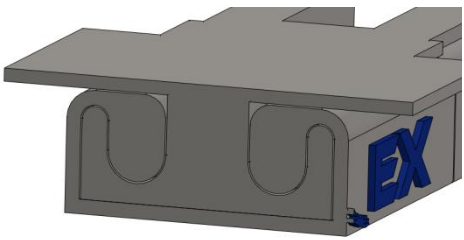


ill.20 Upper Arm Segment optimized



ill.19 Upper Arm Segment original

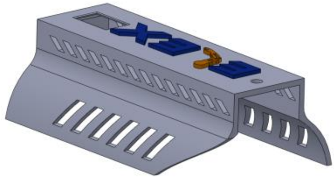
This component has been optimized throughout the project. The original component was used to connect the two upper arm cuffs and the motor adapter. The optimized version consists of two parts, which make sure to connect all other parts and was also developed as an adjustable one. You can now adjust the upper arm segment in 4 different sizes. To hide the Arduino board, a base plate was designed and printed. Using a self-developed guide system, the two components slide into each other and can be adjusted by a push button to the desired size. The push button is controlled with a spring. The ELEX logo was added so it can clearly be visible on the side.



ill.21 Guide rail

6.10 Part 7 - Lower Arm Protection

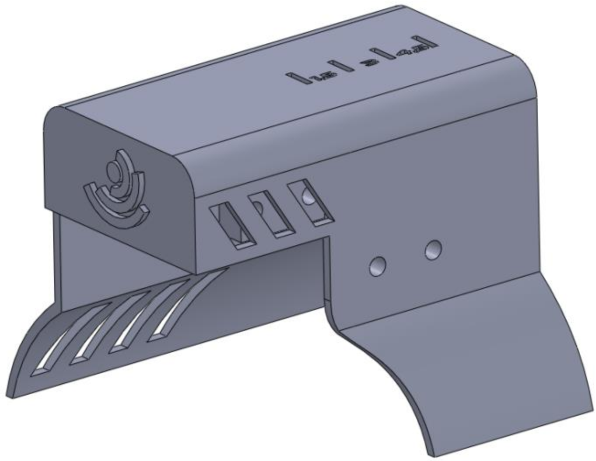
ill.23 Lower Arn Protection



ill.22 Lower Arn Protection

The Lower Arm Protection part has been developed to protect the screen and the electronic parts, as well as to provide a general protection for the other components underneath the sensor. The development and assembly proceeded smoothly. The arm was adjusted to the angle of the Lower Arm segment. Laterally, the Lower Arm Protection has been given a rounding to reach the desired 90 ° when moving without touching the other parts. To screw the arm, a hole was constructed on one side. In order to make the part more vivid, individual slits have been designed, which now decorates the arm with a futuristic look. It can also be seen that the HFU initials were attached to the front part of this part.

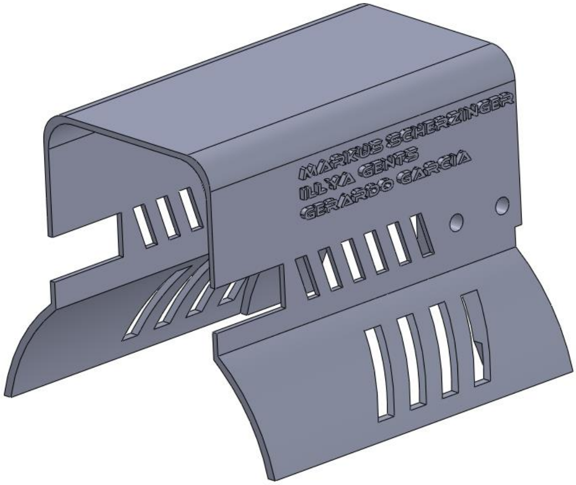
6.11 Part 8 - Upper Arm Protection 1



ill.24 Upper Arm Protection 1

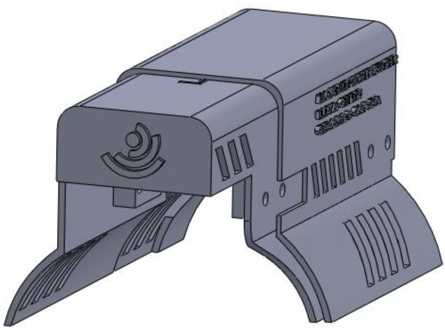
The Upper Arm Protection1 was made to protect the motor and part of the Arduino board. In order to reach the turning angle of 90° of the forearm, a radius was also calculated here. On the upper side of this component you can see a number scale, which should be used to recognize the size set. This component works like a rail system with the part 9. Due to the adjustable sizes of the protection it can also be unassembled, so if larger adjustments are added, it will ensure enough protection of the Arduino board. At the front side of this part is the HFU Logo to recognize the cooperation with the university.

6.12 Part 9 - Upper Arm Protection 2



ill.25 Upper Arm Protection 2

Just like with the other arm segments, this design allows you to see the design with the holes on the sides. Also, the names of the developers as well as the year of development (2017) were added. At the front face, two more holes can be seen. These were installed to prevent the contact with the screws in part 8.



ill.26 Complete Upper Arm Protection

7. Introduction Arduino

ill.27 Logo Arduino

For the development of the ELEX the Arduino Uno was used because it has the necessary functions and is not very expensive. The Arduino works like the 80s machines, the digital inputs and outputs can be on or off, and the analog inputs can return a value between 0 and 1024. The development environment is kept as simple as possible. Syntax highlighting is available, but the code is not completed. To try out smaller codes, the Arduino IDE is still suitable.

The original costs around 22 € on Amazon. But there are generic copies for under 5€ as well, where you have to use different software to use them.

ill.28Arduino Board

Technical data:

- ATMega 328
- Operating voltage: 5V
- Input voltage: 7 - 12V
- Input voltage (Limit): 6 - 20V
- Digital Inputs/Outputs: 14
- IDC 40mA/50mA (3,3V)
- Analog input: 6
- Flash Memory: 32 KByte
- SRAM : 2 KByte
- EEPROM: 1 KByte

8. Display

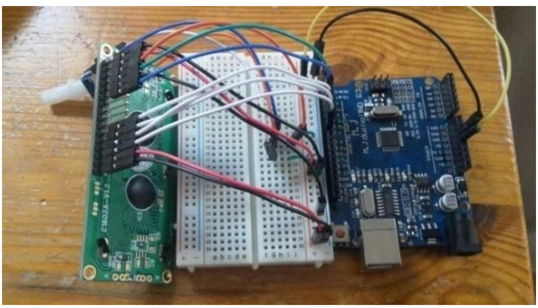
We wanted to attach a screen to see the state of the exoskeleton, for example battery level, acceleration angle, values over time, etc. This should have a low power consuming and be suitable with the Arduino computing power.

8.1 LCD Display

In the laboratory, an LCD Display 1602 was used and, before building the ELEX, it was tested out.

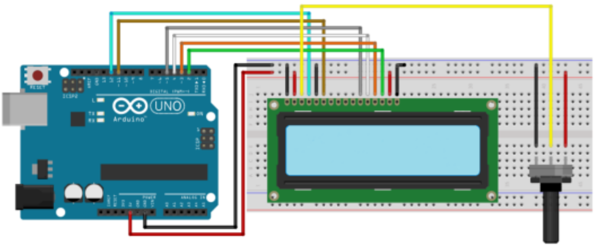
ill.29 LCD screen

1602 means 16 numbers on 1 line. It has a blue background with white characters, which can be adjusted with a potentiometer.



ill.30 Wiring

Because it was hard to work with, it was decided to save as much space as possible to make a more compact board.



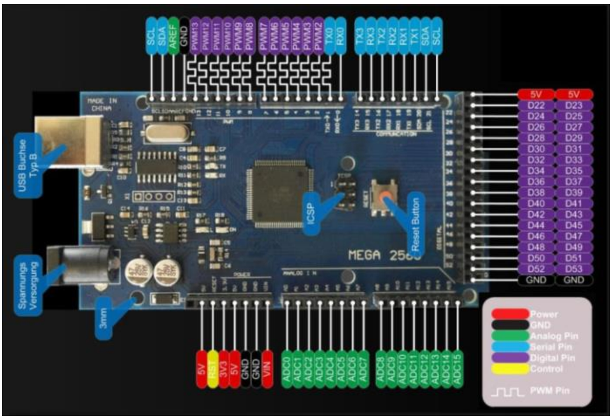
ill.31 Wiring diagram 1

Advantage: The output of standard texts is easy to program and this type of display is more stable than the thin OLEDs. The price of it is also cheaper because, nowadays, this technique is outdated.

Disadvantage: It is very large (80 x 35 x 9mm) and the display is only limited to certain characters. LCD consumes more power than OLED.

8.2 Touch Display Arduino Mega

Since our lab had acquired an Arduino Mega and the TFT 3.2 LCD 240x400mm screen, it was decided to use it because it has many more outputs. First, it was connected without a protection but it did not work and the connection consisted of about 20 cables. Afterwards, the protection V2.2 was bought and connected. On the big touch screen, you could see many things as well as additional functions display/control.



ill.32 Arduino Mega 2560

Advantage: Larger screen with touch display for controlling or visualizing data. After eradicating the problem with the covered pins, the Arduino Mega offers more options and has a higher power.

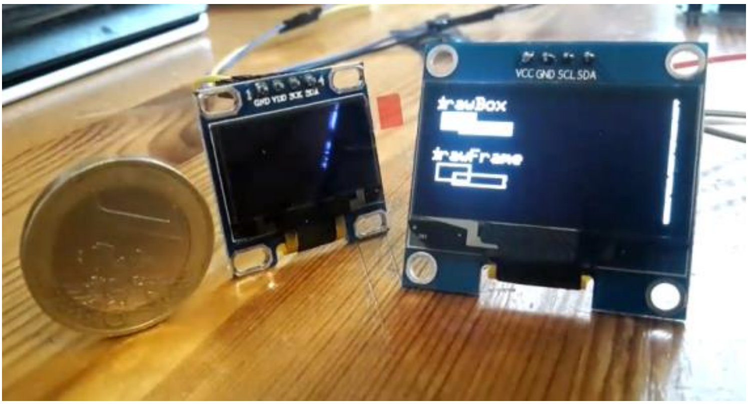
Disadvantage: The size was too big (10.5x5.5x3.5mm) and would have made the exoskeleton bigger. Most of the pins are occupied and the blank ones are covered by the screen, therefore it takes very long to connect.

ill.33 Screen

8.3 OLEDWhite Display 1,3”

The OLED SH1106 display can only show the color white and has 128x64 pixels. This screen is shown on the picture below (right side). After several attempts, this wasn’t totally achieved in terms of resizing because there was no guide but the smaller model next to it.

The biggest display is 1,3” big.



ill.34 OLED screen

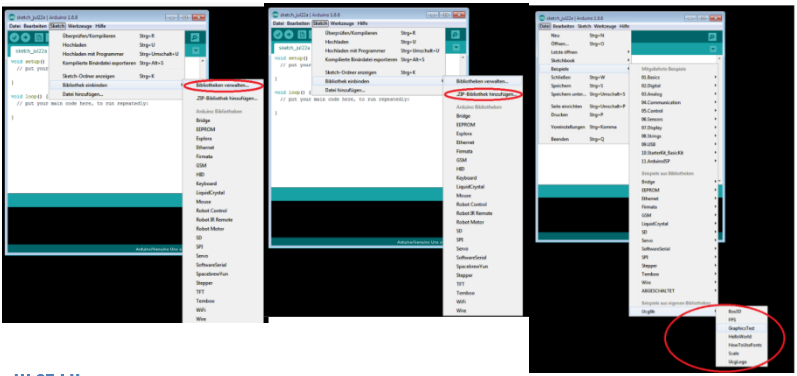
Advantages of OLEDs: Thin design, low energy consumption, therefore, good for portable devices, the response time is about 1000 times faster than LCD.

Disadvantages: Longer battery life than LCD, images can burn in and it is not as resistant to environmental influences as LCD.

8.4 OLED White Display 0,96”

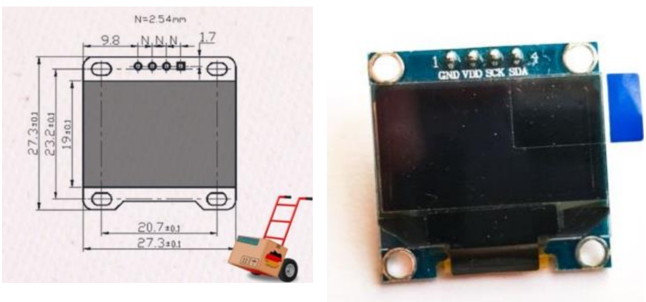
The 0.96” OLED screen was chosen because it uses less power and has a wider online library.

First, the U8glib library was downloaded and installed in Arduino.
The contact lettering on the display may differ.



ill.35 Library

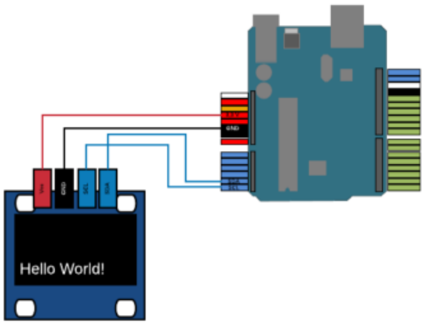
On the left, the dimensions can be seen. This screen is available for under 7 € on eBay; in Chinese shops, for under 3 €.



ill.36 OLED 0,96

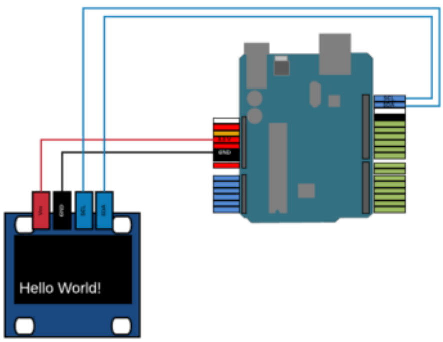
9. Installation des OLED´s

The display is then connected to the Arduino's I2C bus as following.



ill.37 OLED screen 1

If the Arduino has a separate I2C bus, make the connection as following:



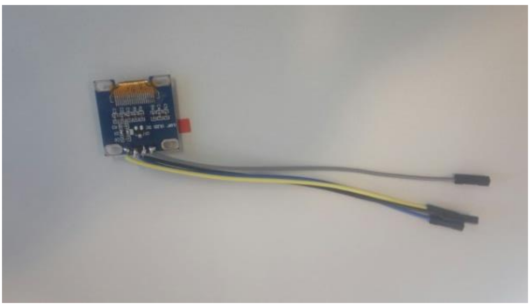
ill.38 OLED screen 2

Both options work fine. But the second one was used to reduce the number of cables on one side.

Information of the OLED

- Display type: OLED GM009605
- Angle perspective: > 160 °
- Resolution: 128 x 64
- Very low consumption at full intensity: Display 0,08 W
- Operating Voltage: 3 V ~ 5 V
- Operating temperature: -30° C - 70° C
- Size: 27 mm x 27 mm x 4,5 mm
- Driver modul: SSD1306
- I2C-Address: 60(0x3C)(Possible determination through tool)
- Control: I2C / IIC / TWI
    - GND
    - Vcc
    - SCL
    - SDA
- Display color: white
- Contrast ratio 2000:1

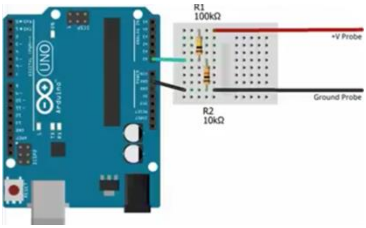
To save about 1,5 cm, the pins were detached and then soldered directly. Therefore, it's not quite modular and cannot be separated so quickly, but in this case, it did not matter because it is permanently installed.



ill.39OLED screen 3

10. Battery level

This is the scheme found on YouTube[1]

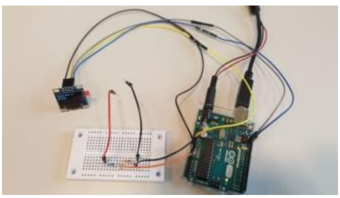


ill.40Wiring diagram 2

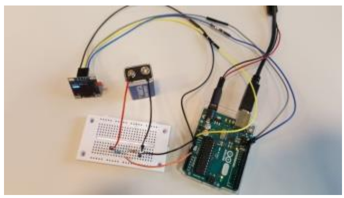
After following the instructions and integrating our code, the following conclusion arrived.

ill.42 Screen 1

ill.41 Screen 2



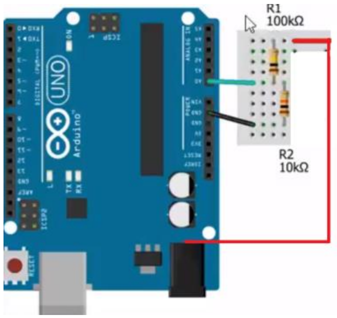
ill.44 Screen 3



ill.43 Screen 4

[1] https://www.youtube.com/watch?v=9gjJIFzXSAE&t=178s

After optimizing, the schematic looked like the following diagram. The positive pole is soldered to the Arduino and the negative pole can be neglected. With this, some cables were saved.



ill.45 Arduino Board wiring

In order to save more space, everything was placed on a board and soldered the cable directly to the resistors.
Tests were then carried out with the voltmeter. A deviation of 0.1-0.3V was found, but it turned out irrelevant to our structure.

ill.46 Wire 1

11. Servomotor

After many tests, the gear of the original motor was worn out, so it was overrunning when the arm was moved. This meant that the path cell did not show the desired force. So, the arm could only be moved if a steady force was applied. In addition, there was then the spring back effect which occurred when the arm has moved quickly in one direction. We tried to find the bug or error in the program but it was easier to use a more stable engine.

Advantage:

The new motor MG 996R had a metal gearbox and 0,343 N/m more torque, which enhanced its functionality.

Disadvantage:

It weighs 9g more and it is 0,01 sec slower in its work speed. It also had no feedback cable, which is the white one in the original.

Conclusion:

In our project, the cable turned out to be irrelevant. As the advantages were bigger, it was decided that a new engine which is also 10 € cheaper.

Specs original motor:

Weigh- 46g
Stable torque 0,637 N/m
Working speed 0.21sec / 60degree (4.8V)

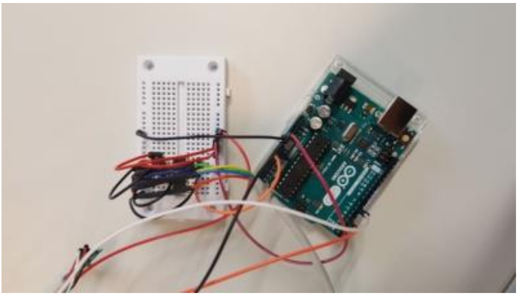
Specs new motor

Weigh- 55g
Stable torque 0,98 N/m
Working speed 0.20sec / 60degree (4.8V)

ill.47 Motor comparison

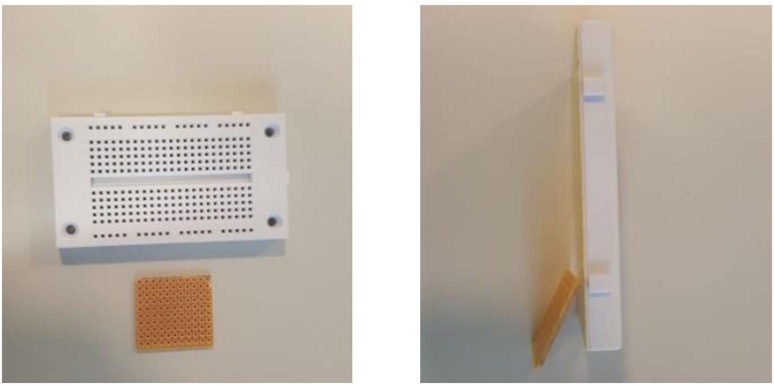
12. Board

Since the breadboard was too big and half of its space was not used, it was optimized.



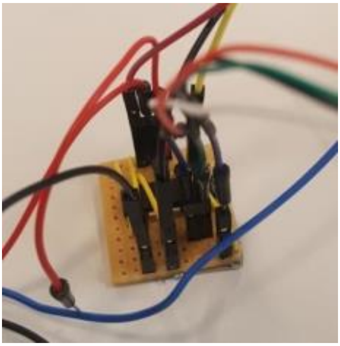
ill.48 Wiringboard and Arduino

A small board was used which was soldered and then filled in because the board per se was porous.

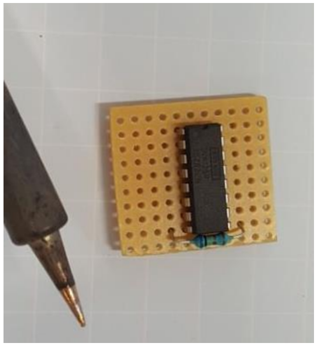


ill.49 Comparison wiring board

The chip and the leads were soldered on the back. The patch cords were disassembled and soldered to the board to allow a modular design in case something should be added later. The "slots" were fixed with superglue.



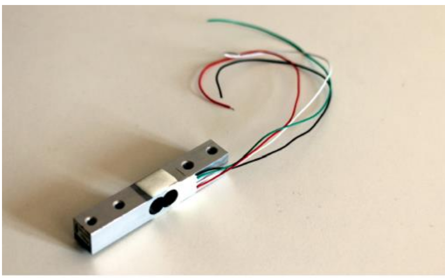
ill.50 Wiringsmallboard 1



ill.51 Solderingsmallboard

Clean and steady work here is very important as well as a third hand is a great advantage. The chip must not be too hot!

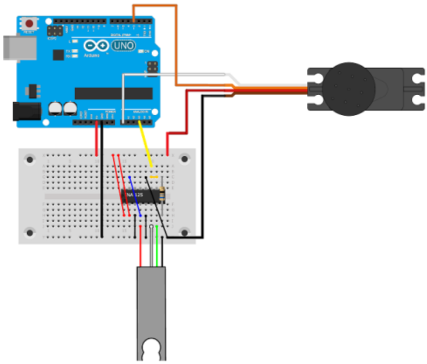
The force sensor was then attached as in the original.



ill. 52 Load cell

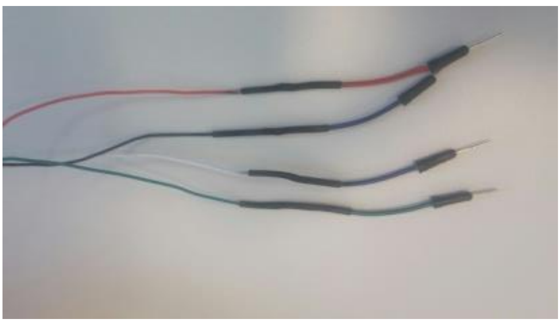
Board with load cell and motor

In the original instructions it is not shown how to connect the two parts together. The plus and minus poles of the motor are "drawn off" from the board. The white cable of the engine can be neglected because the new motor was used.



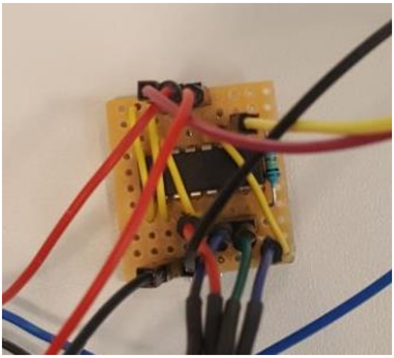
ill.53 Wiring diagram Motor/Arduino

The cables were soldered to the connectors and then insulated.



ill.54 Wire 2

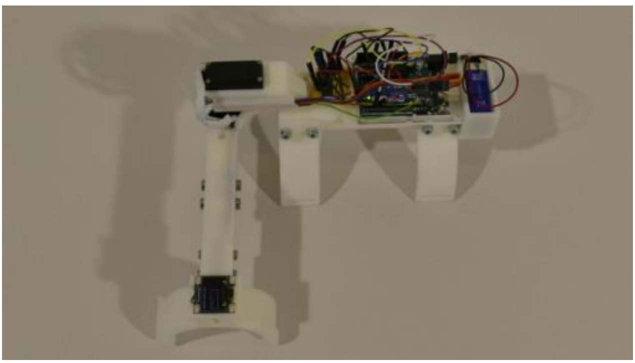
The cables were isolated with glue.



ill.55 Wiring small board 2

The board was finally attached to the arm with superglue and this made the arm ready to use!

We had our precautions because sometimes the cables detached, meaning that another type of gluing might as well be used. We highly recommend the use of superglue for lab practicing since it's the easiest way to do it.



ill.56 Finish wired part

13. Code

```
#include <Servo.h>
#include "U8glib.h"
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE|U8G_I2C_OPT_DEV_0);
 Servo myservo;

 int forceAnalogInPin = A3;
 int forceIs;
 const int forceOffset = 102;
 int pixelpercent = 0;

 void setup(){
 Serial.begin(9600);
 myservo.attach(3);
 }

 void loop(){

 forceIs = analogRead(forceAnalogInPin);
 forceIs -= forceOffset;
 Serial.print("Force: ");
 Serial.println(forceIs);
 delay(10);

myservo.write(122);
int servopos = 122;
```

III. 57 Declaration of variables

First the libraries and the display are initiated.

The analogue input is defined and the values are set to 0 with the variable "forceOffset". If the arm weighed more, this value should be changed. This variable will help assign the force as 0, so it can be easier to use.

In setup, the start and servo are both initiated.

The loop determines what is displayed in the serial monitoring and runs the code per se.

```
int n= 0;
while(true)
{
  if ( n%10000 == 0 )
      BAT();
  n++;
  int summe = 0;
  for ( int i=0; i<2; i++ )
   {
   forceIs = analogRead(forceAnalogInPin);
   forceIs -= forceOffset;
   summe += forceIs;

   }
   forceIs = summe/2;

   if ( forceIs > 5 and forceIs < 50)
   {
      servopos +=3;
   }

   if (  forceIs < -5 and forceIs > -50 )
   {
     servopos -=3;
   }

   if (  servopos <=40 )
       servopos = 40;
   if (  servopos >=122 )
       servopos = 122;
```

III. 58 Movement definition

Every 10000 loops, the screen is updated, otherwise it would interfere with the Arduino's processing speed and with its main function.
An average is performed to get a more accurate result.
If the force is greater than 5 and less than 50, the servo will move 3 units in the opposite direction.
The maximum freedom of movement is from 40° to 122°.

```
    myservo.write(servopos);


    Serial.print("Force: ");
    Serial.println(forceIs);
    Serial.print("Pos: ");
    Serial.println(servopos);

  }
}
void u8g_prepare(void) {
  u8g.setFont(u8g_font_6x10);
  u8g.setFontRefHeightExtendedText();
  u8g.setDefaultForegroundColor();
  u8g.setFontPosTop();
}

void u8g_r_frame(uint8_t a) {
  u8g.drawStr( 0, 0, "EDUEXO ELEX HFU");
  u8g.drawRFrame(0, 28, 80,28, 2);
  if (pixelpercent == 0)

      u8g.drawStr( 10, 32, "X X X");

  else
      u8g.drawRBox(4, 32, pixelpercent*2,20, 2);
}
```

III. 59 Screen display

In this code, it's explained how the Hello World Code was taken and changed.

The basics in this part are the initialization of the screen.

The screen will output the text "XXX" if no battery is connected.

```
uint8_t draw_state = 0;

void draw(void) {
  u8g_prepare();
  u8g_r_frame(draw_state&7);
}
void BAT() {

   int sensorValue = analogRead(A0);
    float voltage = sensorValue * (9 / 1023.00)  *68;
    Serial.println(voltage);
    double voltpercent = voltage/9.0;
    pixelpercent = (int) (voltpercent*38.0);
    Serial.println(pixelpercent);

delay(1000);
  u8g.firstPage();
  do {
    draw();}
  while( u8g.nextPage() );
    draw_state++;
  if ( draw_state >= 9*8 )
    draw_state = 0;

}
```

III. 60 Drawing and Battery Functions

The A0 input can detect values from 0 to 1023. When it's stored in the "int sensorvalue" variable, it is then divided by 1023 and multiplied times 68 so voltages from 0 to 9V can be read. It was then tested with a multimeter and got a deviation of 0.1V which is insignificant to us.

The do-while function in this part of the code was used to call multiple times the draw( ); function. This prepares the screen for display.

14. Extras

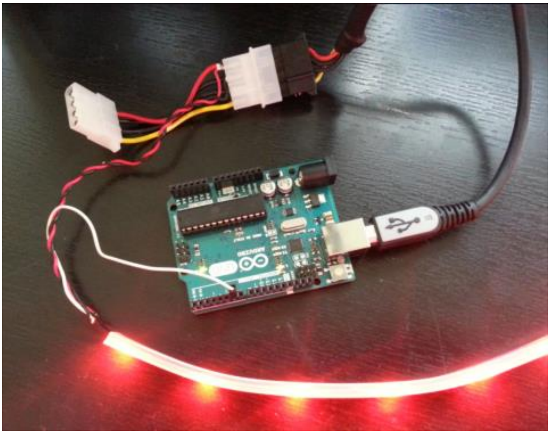
There are many ways you can attach things to the arm. But since our time was limited, the original ideas were not performed.

One was a pulse sensor that cost about 7€ and can read the pulse through the arm.

The problem we found was that maybe the Arduino would have been overloaded.

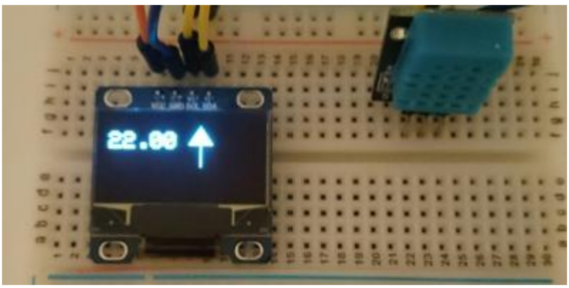
ill. 61 Pulse sensor

A second idea was a light strip was attached to the Exoskeletts. The problem was that this could mean higher power consumption.



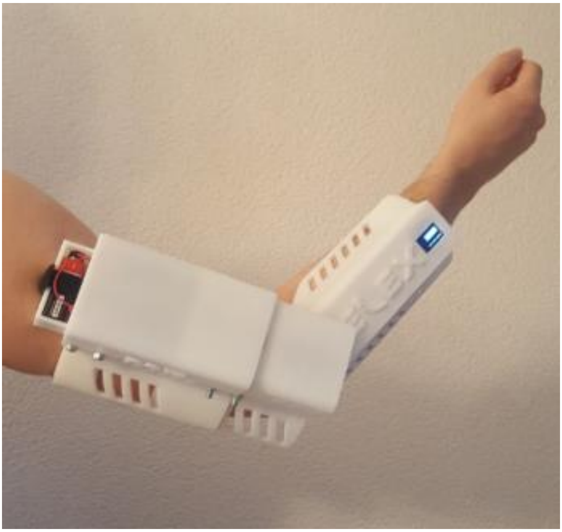
ill. 62 LED light

Our third idea was a temperature sensor, which could be nice to solve the other two problems, previously mentioned (overheating).

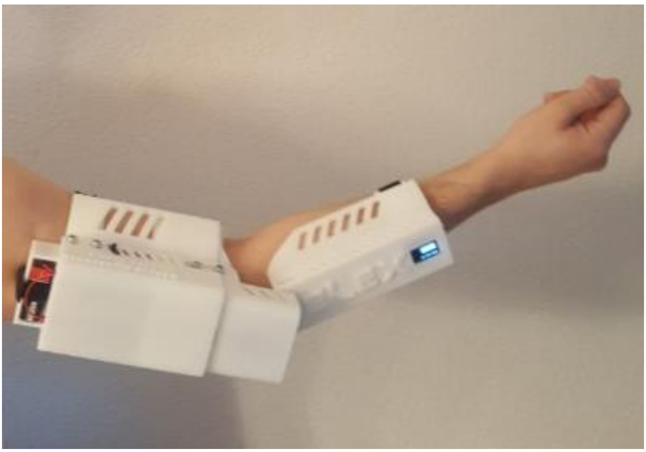


ill. 63 Screen

15. Finished elbow´s exoskeleton



ill.64 Finished elbow´s exoskeleton 1



ill.65 Finished elbow´s exoskeleton 2

16. Conclusions

As a team, we saw that each and every part was built with our own expertise, depending on the area. Although, we couldn’t have made it without the help of each other.

We developed our knowledge and deepened on the basics of the fields and learned from other fields we didn’t exactly knew.