Low-Cost Force Sensing System Prototyping for Upper-Limb Rehabilitation

Jessé de Oliveira Santana Alves[1]
Professor/Advisor: Maciej Bednarczyk and Bernard Bayle
[1] Université de Strasbourg

Abstract

Currently, robotics-assisted devices, such as exoskeleton and collaborative robotic arms, have been used for rehabilitation therapy. Where one of the biggest causes of the physical disability is stroke. However, despite the high benefits and potential of using these devices, the clinical application remains limited due mainly to the complexity and the high cost. Therefore, different strategies are used to overcome these limitations, such as the use of alternatives sensors and materials. In this context, this article presents the prototyping of a low cost and compact bracelet to detect the upper arm contraction using Force Sensing Resistor (FSR) and then, send the data from a cheap microcontroller to Robot Operating System (ROS2) environment.

Keywords: Rehabilitation, Post-Stroke, Robotic, FSR, Low-Cost.

1 Introduction

1.1 General Context

The stroke is defined by the World Stroke Organization as the interruption of the blood supply to part of the brain (Martins 2023) and is one of the main diseases responsible for long-term physical disability in adults (Azhar 2022). This interruption can cause different damages depending on which part of the brain is affected and how quickly it is treated. One of the most common damages are the lost of motor mobility and speech.

The Hemiparesis, defined as the weakness or the inability to move on one side of the body, is a common stroke consequence that causes difficulty in completing everyday tasks, such as the coordination, fine motor skills and movement precision. These motor system damages are directly related with changes in muscles architecture and the shortening of muscle fiber length, generating, in this way, less force and precision by the members of the human body.

Corresponding author(s): Jessé de Oliveira Santana Alves, Université de Strasbourg, jesse.alves@etu.unistra.fr

In this context, the post-stroke rehabilitation is usually done with a combination of physical and occupational therapy, being advised by the National Clinical Guideline for at least 45 minutes of physical therapy daily (Physiopedia 2022). These activities going from treadmill and exercises aimed at muscle strengthening to the use of high technology, such as virtual reality, electrical stimulation and robotic rehabilitation. This last one is a research trend in the field, where the robotic assistance can be used actively, passively and with haptic feedback. In (Lee 2020) these robots are classified in two main groups, the end effector model and exoskeleton. The last one is attached to the patient at multiple points, in order to match both join axes, the patient and the robot.

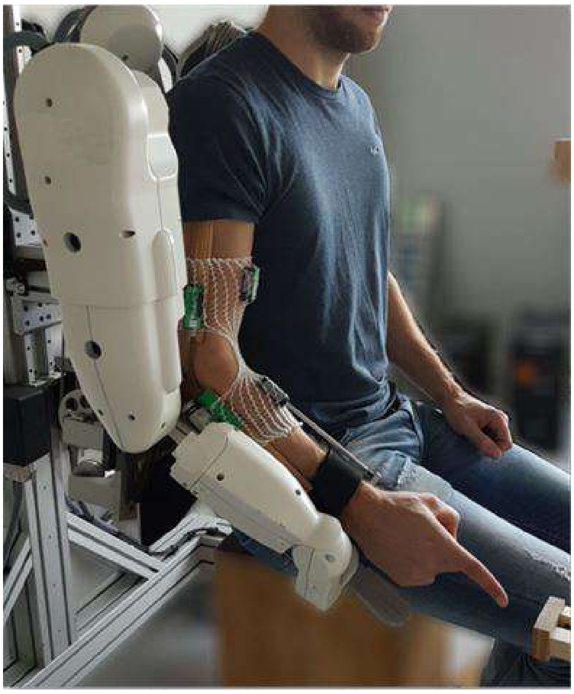


Figure 1: Exoskeleton Robot by (Saclay 2018).

The Figure 1 shows this configuration applied to the shoulder, upper-arm and forearm. In the present project this robot architecture is used, together with sensors to detect the muscle contraction from the patient, or at least its intention, and then, perform the physical therapy exercises and/or the day-to-day activities, such as grab objects and picking and place, giving them assistance while maintaining their autonomy (Azhar 2022).

Regarding the use of devices to detect the muscle contraction - object of contribution from this internship - different technologies can be used to measure the muscle activity, such as the surface electromyography (sEMG), inertial measurement unit (IMU), pressure sensors, force sensing resistors (FSR), electroencephalography and strain transducers. Among them, the electromyography (EMG) - considered the gold-standard technique used for the evaluation of muscle activity and contraction (Casaccia et al. 2015) - is the most common technology in the robotic application, due to the high number of important parameters present in the signal acquired. Because of this, it is possible to find important contributions in this field, such as the control for a hand exoskeleton rehabilitation (Wege & Zimmermann 2007; Mulas et al. 2005), the lower-extremity exoskeleton with one actuated degree of freedom in the knee joint (Fleischer et al. 2006), etc.

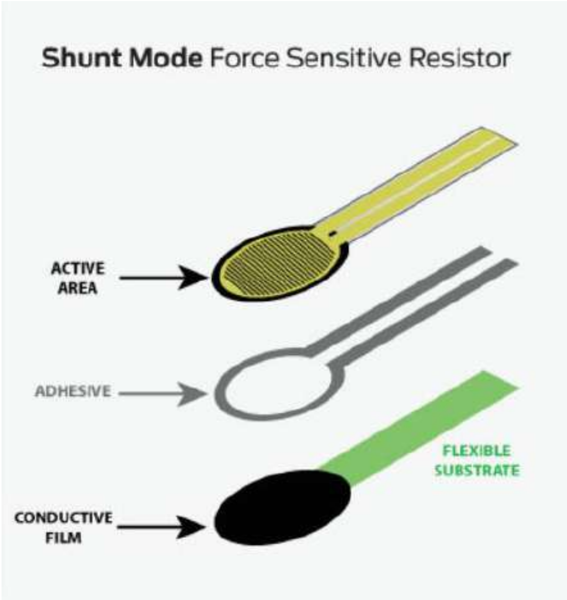
However, the EMG sensors are expensive and not easily available for individuals (Bawa & Banitsas 2022), being more used in rehabilitation clinics with specialized professionals, requiring a high cost for patients and medical insurances. For these reasons, there are also research branches in this field to seek, test and develop low-cost alternative sensors and devices that offer adequate performance for rehabilitation robot applications. Both for design low-Cost EMG Sensor compared to a commercial-based system (Bawa & Banitsas 2022; Prakash & Sharma 2021; Toro et al. 2019) and alternative low-cost sensors such as Force-Sensing Resistors (FSR) (Prakash et al. 2020; Castellini & Ravindra 2014).

1.2 The Research Goal

In this way, the present research aims to project a parsimonious device, both exoskeleton robot and muscle contraction measuring device, for upper-limb rehabilitation and force and precision compensator, similar to that shown in Figure 1. For that, low-cost materials are intended to be used in each part of the device. Therefore, it is necessary to prove scientifically that each parsimonious device part have a sufficient performance for these applications. Thus, regarding the muscle contraction measuring device, a cheap bracelet using FSR sensors was chosen to be tested and compared with EMG data and FSR commercial device, by Delsys brand.

1.2.1 The working principle of Force Sensing Resistors (FSR)

EMG sensors utilize electrodes to detect the electrical signals transmitted from the nerves to the muscles, thereby detecting muscle contractions. On the other hand, FSR sensors act as variable resistors that modify their resistance in response to applied force, being, in that way, easier to treat the voltage signal mapped between a range of resistance. The Force Sensing Resistors (FSR) used in this project are of the Shunt Mode type (Figure 2), where there are three layers, the top layer, called active area, is printed with conductive silver arranged into sets of interdigitating fingers, two sets, but not connected directly. The bottom layer, called conductive film, is a PET film made with a conductor material, usually carbon. Then, the middle layer, is an adhesive non-conductive spacer film that isolate the top layer from the bottom layer. Therefore, in this architecture, the unload state of the sensor can be interpreted as a infinitive resistor (in practice more than 10 $M\Omega$), and when a force is applied, the semi conductive traces of top layer shunts the traces on bottom layer reducing the resistance until 1 $k\Omega$ or 1Ω, depending on the manufacturer.



(a) The layers

(b) The sensor

Figure 2: The FSR sensor Shunt Mode (Tekscan 2023).

1.3 The Research State of the Art

In (Azhar 2022), some initial experiments were made, showing that, within your limitations, it is possible to map the biceps contraction using FSR sensors inside a closed bracelet (Figure 3), through the variations of volumetric and stiffness of the muscle which exert radial pressure. Among these limitations and constraints, it is possible to highlight:

- The static drift - a phenomenon where the output sensor reading changes over time without any input modifications - of 21.4% for interlink brand and 6.2% for FlexiForce brand, after a constant load application during 20 hours observed in (Dabling et al. 2012) study.
- The non-linearity relationship between low force applied (range of 0-4N) and the voltage level (Sadun et al. 2016).
- The uncertain and unreliable results using the FSR sensors directly on the skin, thus demanding the use of some adapter between the sensor and the skin, where one type of adapter was originally designed for this application in this internship.
- And finally, the necessity to calibrate each sensor when these are used inside a closed bracelet, since the minimum and maximum value read on the FSR sensor will depend on the force used in the manual fixation of the bracelet on the upper-arm. This internship work will also present, as part of the prototyping, an efficient and intuitive way to calibrate these sensors.

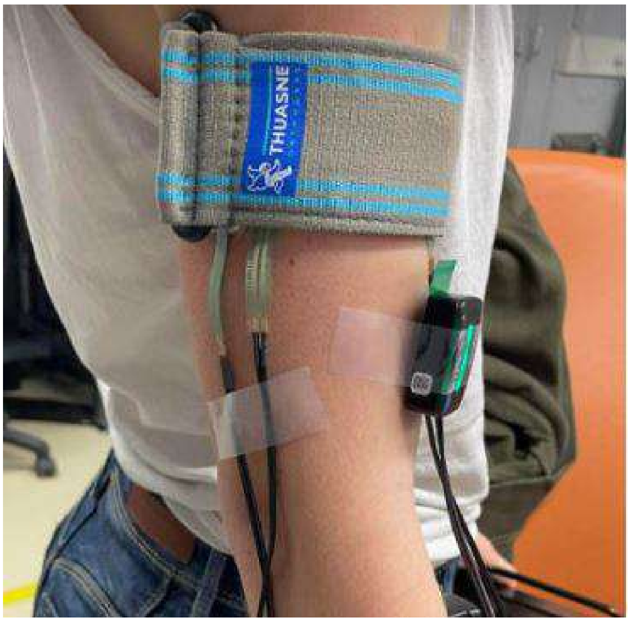


Figure 3: The FSR bracelet used in Azhar experiments (Adapted from Azhar (2022)).

1.4 The Present Internship Objectives and Specifications

In that context, the present internship aims to build a low-cost and compact prototype of a bracelet - the structure part, the hardware and the software - using FSR sensors and a microcontroller to detect the upper-arm contraction and send this information to ROS2 Environment. This prototype should take into account all the limitations described in the subsection 1.3 and be intuitive to manipulate for the user, since its goal is to accomplish the same and similar experiment describe in (Azhar 2022), replacing the setup shown in the Figure 3, in order to prove - with more accuracy - that a cheap bracelet has a sufficient performance to control the exoskeleton robot.

1.5 Work Environment

To achieve this task, the prototyping was made using both laboratories from Icube at Institute Surgery Guided Par L'image (IHU) and from École d'Ingénieurs Télécom Physique Université de Strabourg. Where the main research team involved in the internship was the Robotics, Data science and Healthcare technologies (RDH) team. As supervisors, the professor Maciej Bednarczyk and professor Bernard Bayle guided and followed the entire prototyping process of the internship.

1.6 The Internship Management

In order to project, build and test the prototype during 8 weeks of internship, the work has been managed following the Japanese Kanban workflow management method, where all the tasks to accomplish the project are divided and subdivided in cards and dynamically categorized as Backlog, To Do, Doing, Pending and Done. This method was learned in the Digital Economics core taught in the HealthTech Master M1 program and applied using the Trello tool (Figure 4).

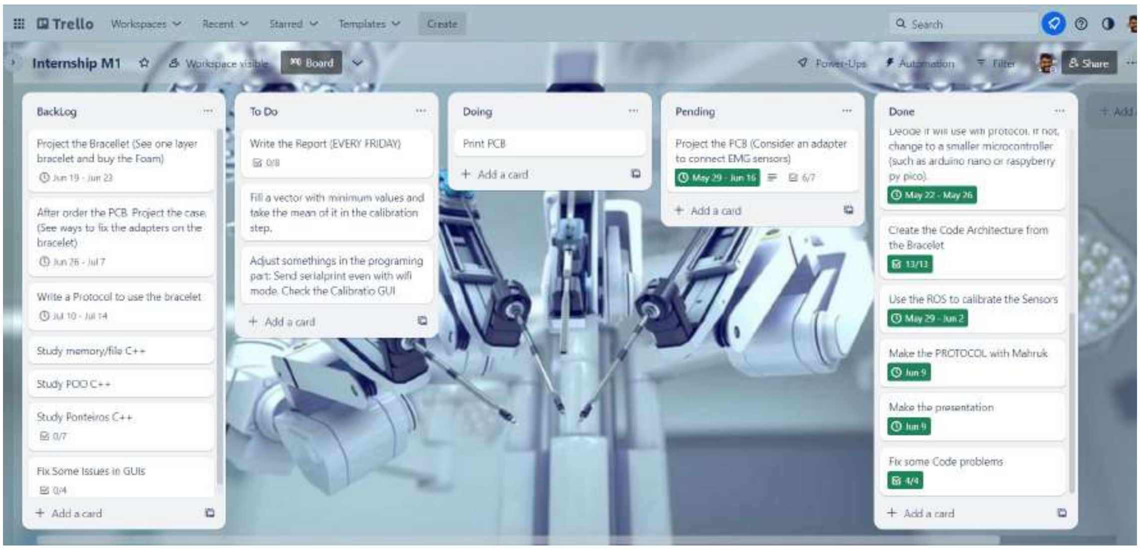


Figure 4: The Kanban used in Trello.

In that way, it was defined deadline for each macro task, as it is shown in the Table 1 below. It is also important to mention that this present report has been written during the prototyping process.

Table 1: The Macro Project Schedule

Tasks	Period
Write the code for the microcontroller, the Graphical Interface and ROS2 environment.	From 22/05/2023 until 06/06/2023
Project the printed circuit board (PCB) and select components from RS website.	From 06/06/2023 until 16/06/2023
Project the cases and test the circuits.	From 16/06/2023 until 23/06/2023
Mount all the parts on bracelet and test the prototype.	From 23/05/2023 until 15/06/2023

2 The Bracelet Prototyping

This section will present the methodology used to project, build and test the bracelet prototype. Therefore, it will be presented the methods used to validate each part of the device: the software, the hardware and the physical part. However, this report also presents a tutorial section to explain step by step how to reproduce the prototype in the Appendix A, more objectively and directly. Furthermore, it is presented in Appendix B an intuitive tutorial on how to use the prototype, once it is built.

The bracelet projected here is composed of three main parts, the box (Figure 5(a)), the sensors (Figure 5(b)) and the band (Figure 5(c)). The box is composed by a 3D printed case with the hardware part inside, the sensors are composed by the Force Sensing Resistors (FSR) inside the 3D printed adapters and the band is the tissue part responsible to attach on the upper-arm and gather all the other parts. As a final product, it is expected to reach a compacter version and with more features of the first prototype developed previously (Figure 6).

(a) The box. (b) The sensors (c) The sensors

Figure 5: The parts of the Bracelet Prototype.

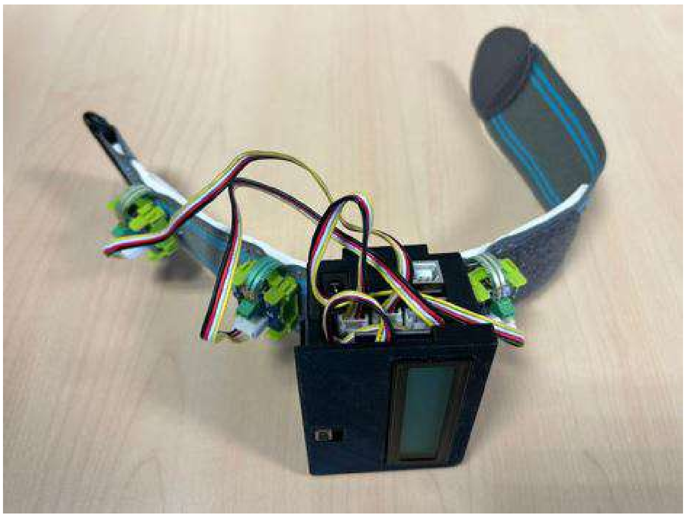


Figure 6: The first prototype.

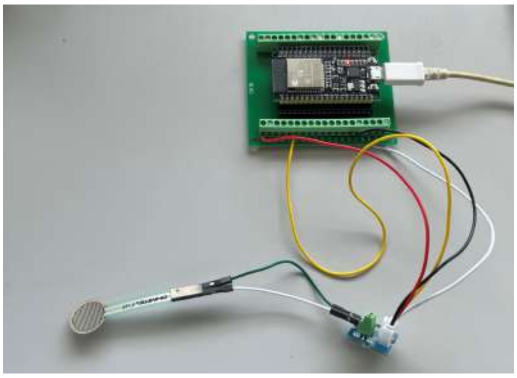
Therefore, the process to build the new bracelet was:

1. Program and test the reading of the FSR sensors using the ESP32 microcontroller.
2. Program the sending from ESP32 and receiving in a python code the sensor reading via Serial and via Wi-Fi.
3. Program a Graphical Interface to choose and save the type of communication.
4. Program the Calibration Process.
5. Stream the reading data via ROS2
6. Read this topic (Subscribe Node) and make a 2DOF planar robot mimic the arm contraction.
7. Project the PCB.
8. Test each part of the circuit separately.
9. Project the sensor adapters.
10. Project the 3D printed cases.
11. Fix all the parts on the bracelet and test.

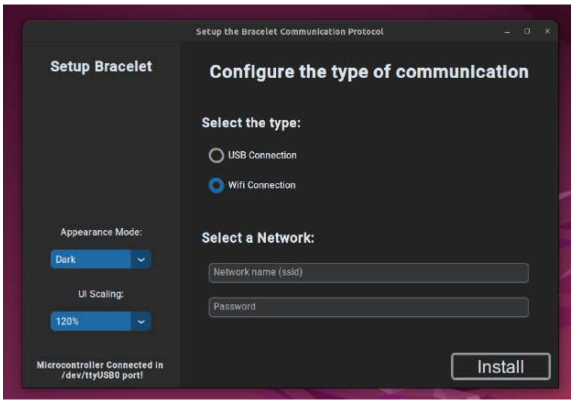
All these steps will be detailed in the following subsections.

2.1 The Software

The first part developed in this project was the software, starting for the microcontroller code, where the Groove Kit was used to test the analogical reading by the ESP32 (Figure 7(a)) together with the Arduino IDE. This board has a 12-bit analog-to-digital converter (ADC) for its analog input ports, thus, when the voltage level from the sensor varies between 0V and 3.3V, the number of discrete levels has $2^{1}2$ or 4096 distinct levels. Once this part was validated, the sensor data was programmed to be sent via serial from the ESP32 and be read in a python code. The same process was developed via Wi-Fi. Then, in order to be intuitive and practical the use prototype, a graphical user interface (GUI) in pyhton code, using the *customtkinter* library was projected to decide and save the type of communication (Figure 7(b)). In that way, the user needs to run this setup only the first time you configure the prototype, since this process saves these settings in the microcontroller.



(a) The sensor reading test.



(b) The Type of Communication Setup GUI.

Figure 7: The software part.

2.1.1 The Calibration Process

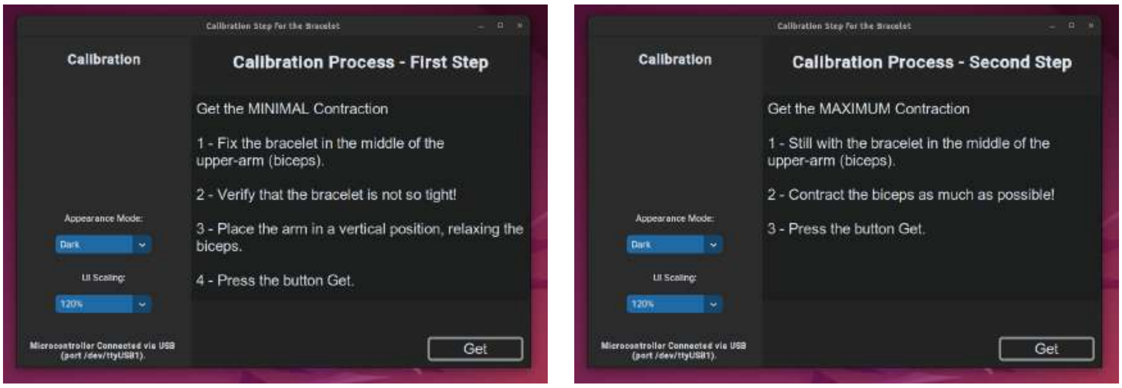
Once both type of communication, serial and Wi-Fi, were validated, a calibrated process was developed to map the sensor reading between 0% of contraction and 100% of contraction. This procedure was projected to work as follows:

1. Fix the bracelet on the upper-arm.
2. Relax the arm, putting in a vertical position.
3. Get the minimum value read from each sensor. This process is done, filling a vector with the sensor data for each FSR during a custom time defined (such as 5 seconds or 10 seconds). Then, take the average of each vector and assign each vector separately its minimum value.
4. Now, contract the arm as much as possible and keep this position during the same time defined. The same process is used to get the maximum value.
5. Once these two steps is done, each sensor has its value mapped in terms of percentage between the average of the minimum value read and the maximum, following this expression:

$$FSR_{mapped} = \frac{FSR_{read} - min}{max - min} \times 100\% \tag{1}$$

6. Lastly, the FSR data sent is the average of all sensors attached on the bracelet, being possible to choose between 3 or 6 sensors.

All these process is also guided from an intuitive and objective way for a Graphical User Interface (GUI) written in python (Figure 8) in order to facilitate the use of the bracelet.

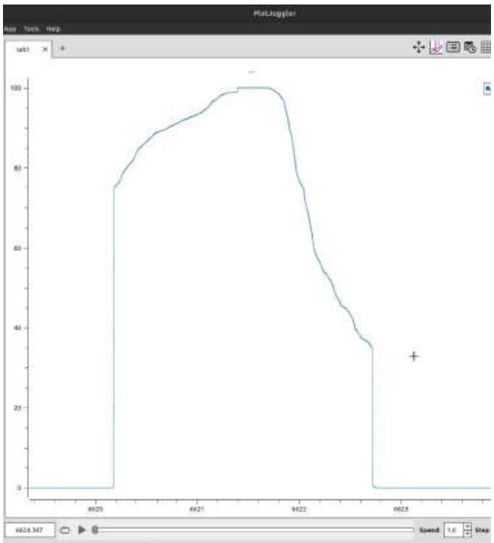


(a) Get Minimum Value. (b) Get Maximum Value.

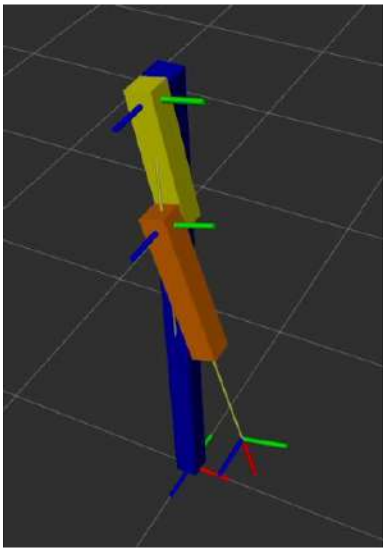
Figure 8: Calibration process GUI.

2.1.2 The ROS2 streaming data

Finally, the FSR data mapped received in a python code is published as a topic in ROS2 environment (Figure 9(a)) with until 1500 Hz stable publishing via USB port. And, in order to test the quality and performance of the signal streamed in ROS2 environment, it was used a 2DOF planar robot to mimic the arm contraction (Figure 9(b)).



(a) The FSR topic published in Plot Juggler.



(b) The 2DOF planar robot.

Figure 9: The ROS2 environment.

2.2 The Printed Circuit Board (PCB)

Upon completing the software part, a Printed Circuit Board (PCB) was designed (Figure 11 and 12) and developed (Figure 10) to assembly the microcontroller, the sensor connectors, the resistors and capacitors to read and filter each sensor, a battery charger modulus to operate wireless, a power switch, a tension regulator for 3.3V and a push button to change the type of communication, such as change the WiFi SSID or password, change from WiFi to USB serial, etc.

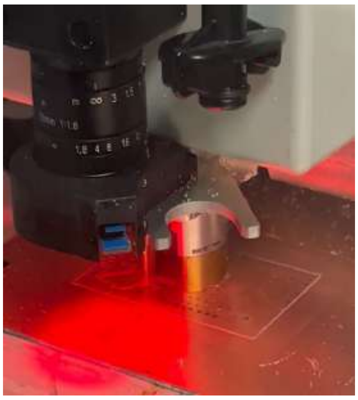
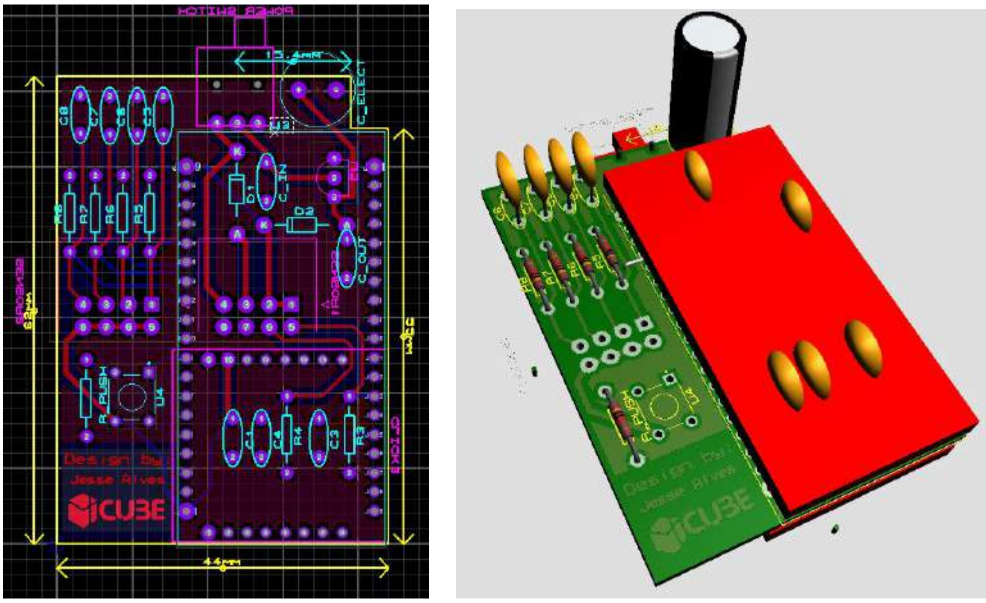


Figure 10: The PCB manufacturing at TPS lab.



(a) The PCB layout. (b) The PCB 3D visualization.

Figure 11: The PCB project.

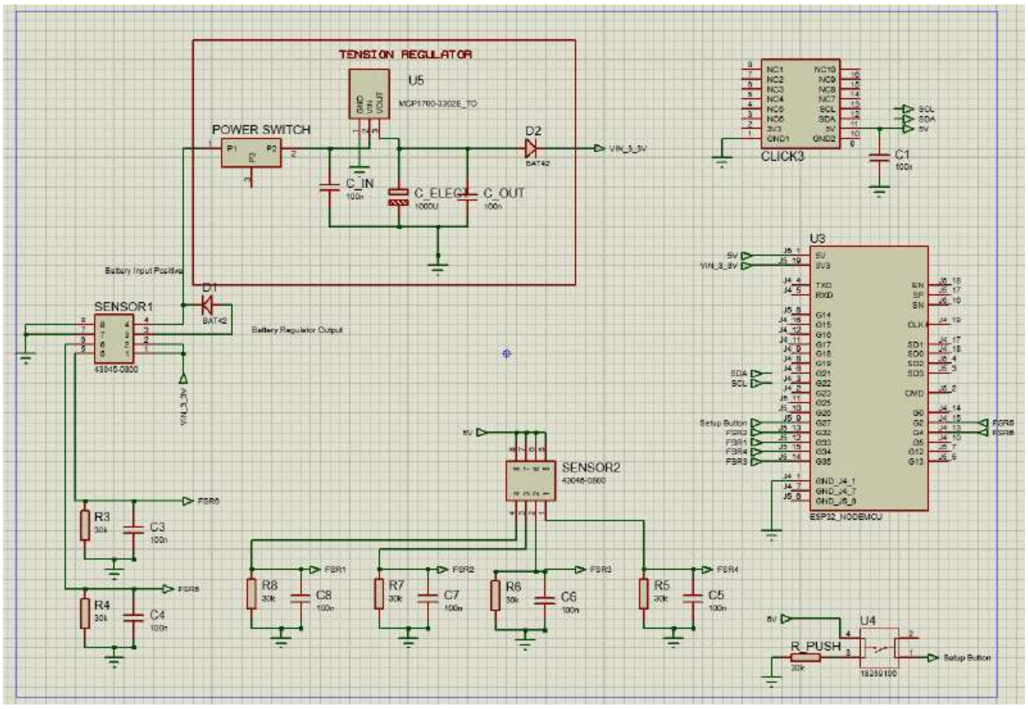
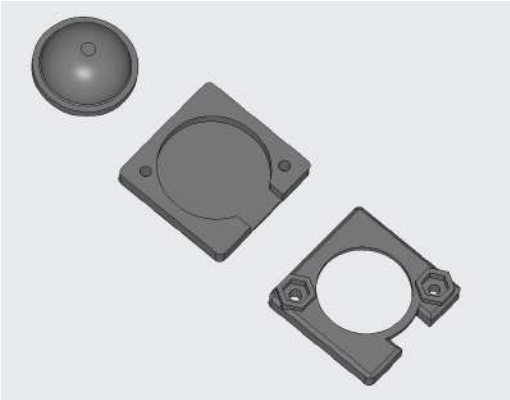


Figure 12: The PCB electrical circuit.

2.3 The Sensor Adapters

The CAD model of an adapter was projected (Figure 13(a) and 13(b)) to allow the correct contact between the FSR sensors and the skin (Figure 5(b)) and at the same time fix the sensors on the bracelet (Figure 14). The spherical part of the sensor increase the pressure on the arm skin, allowing the sensor to read with more precision the increase in muscle volume and stiffness. At the same time, the case keeps the sensor inside a flat surface, fixing and protecting it. Once the sensors were fixed on the bracelet with the adapters, the electrical connection with the PCB was made via custom wires (Figure 15), also built as part of the internship work, between the two bracelet layers (Figure 14).



(a) The CAD project.

(b) The Adapters Printed.

Figure 13: FSR Adapters.

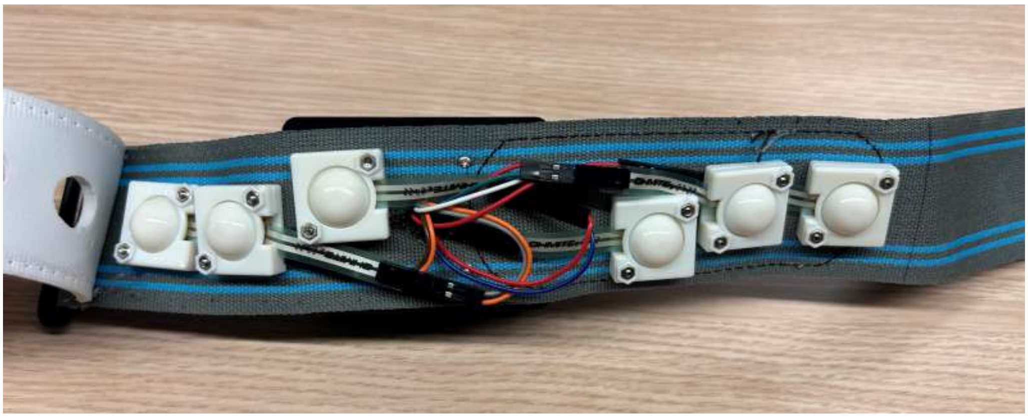


Figure 14: The Sensors with Adapters fixed on bracelet.

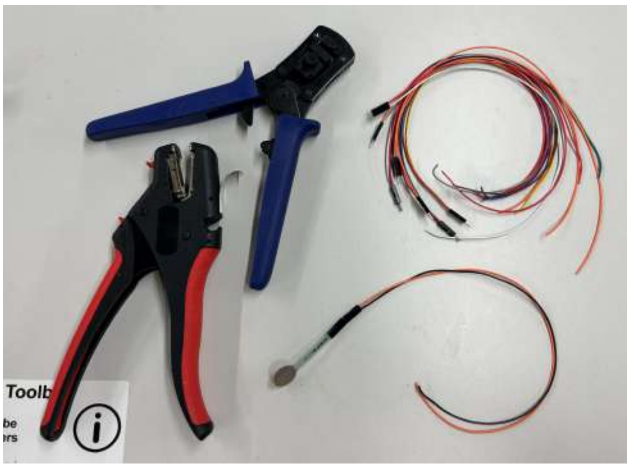
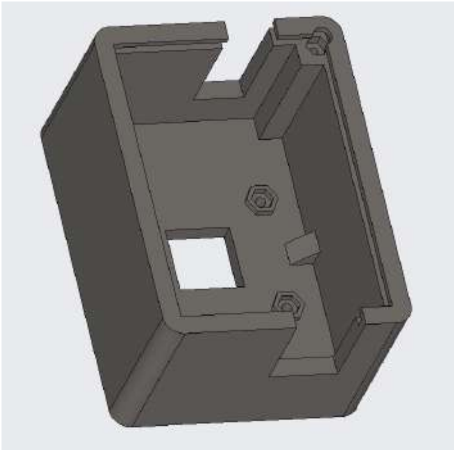


Figure 15: The FSR wires.

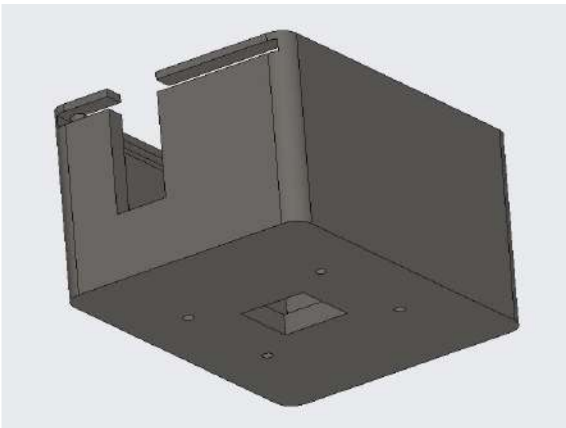
2.4 The Circuit Case

A 3D printed case with a sliding lid also was projected on CREO PTC software to hold and fix the PCB and wires on the bracelet (Figure 16).

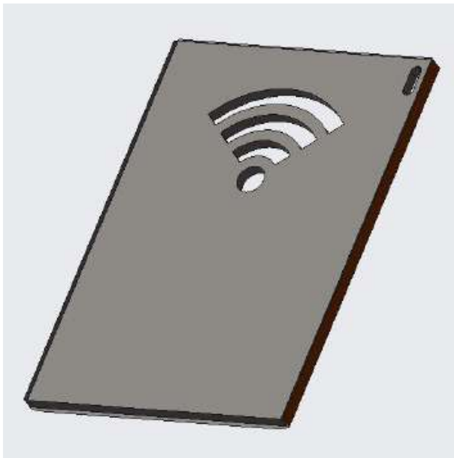


(a) The base

(b) The base



(c) The base

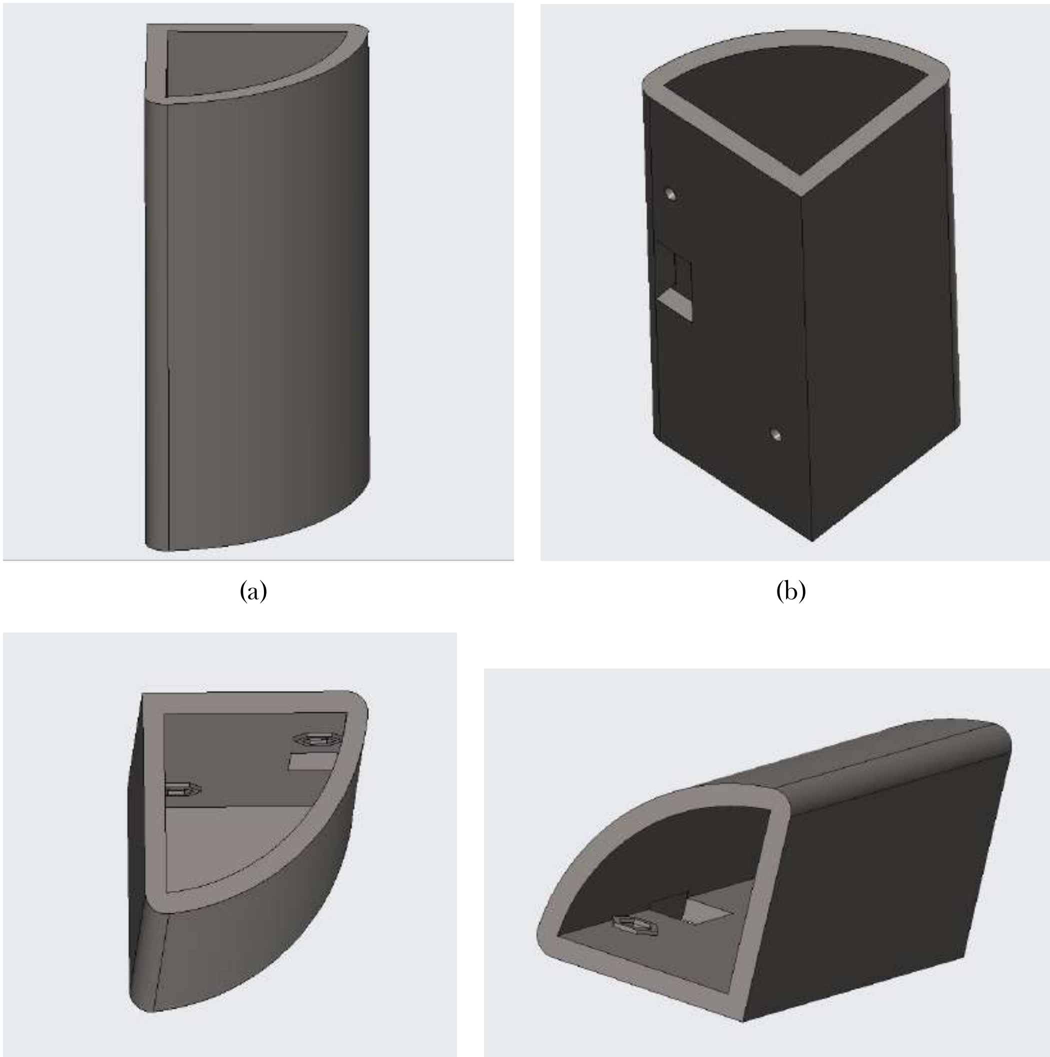


(d) The Lid.

Figure 16: The Case.

2.5 The Battery Case

Finally, since the bracelet was projected to be used with a lithium battery, a battery case, shown in the Figure 17, also was projected to hold the battery and fix it on the bracelet.



(a) (b) (c) (d)

Figure 17: The battery case.

3 Results

3.1 Preliminary Tests and Results

3.1.1 The Hardware

Before to test the entire bracelet, partial tests were performed. Thus, regarding the PCB, the circuit projected was tested via numerical simulation in Proteus software, then was tested each part separately on a breadboard, in order to validate on the practices the circuit projected. Therefore, the voltage level of each sensor was tested (Figure 18) and retested at this point (Figure 14) in the prototyping, as well as the battery charging circuit (Figure 19). Once these steps were validated, the PCB soldering started.

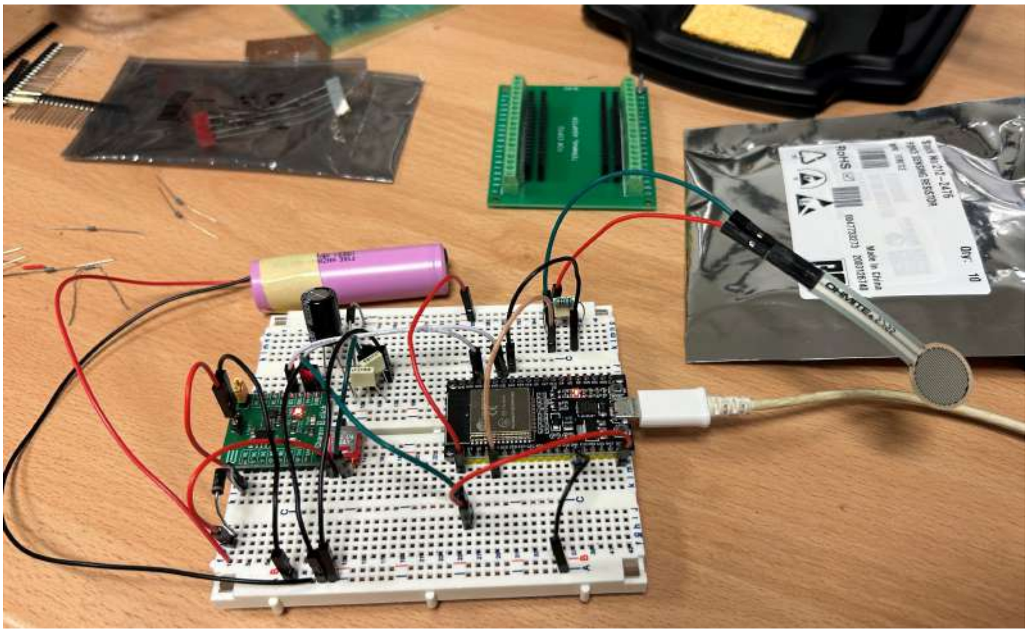
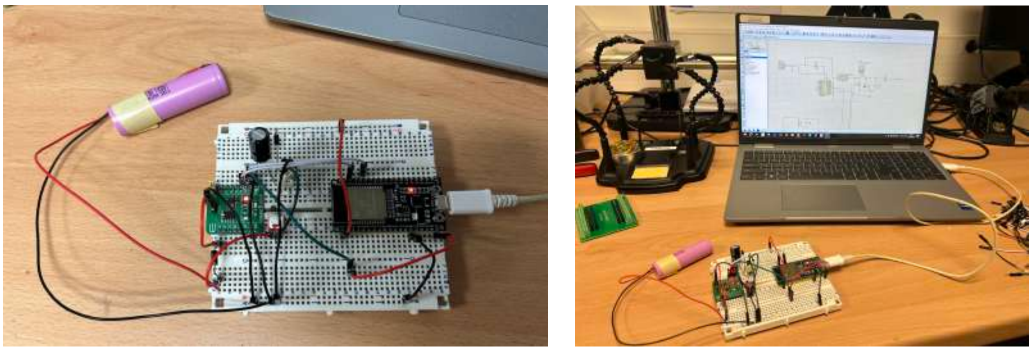


Figure 18: The voltage test of each sensor.



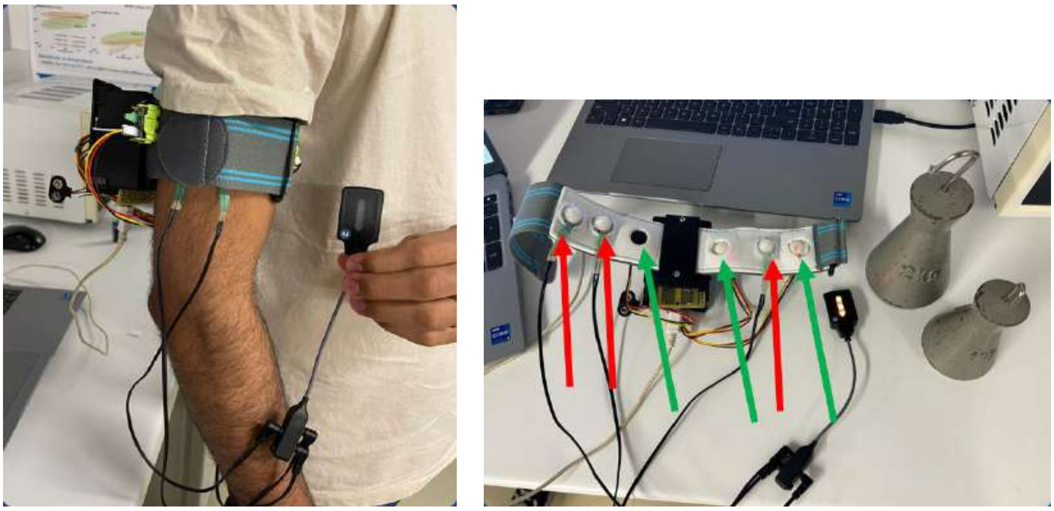
(a) The CAD project.

(b) The Adapters Printed.

Figure 19: FSR Adapters.

3.1.2 The Software

The same preliminary test was done in order to validate the software. Thus, the data sent from the bracelet was recorded in the ROS2 environment and compared with the FSR commercial sensors, by Delsys brand. The test protocol followed was similar to that performed in (Azhar 2022). Where the data average of three sensors connected to the bracelet prototype was compared with the data average of three sensors connected to the Delsys device (Figure 20). The experiment first followed the calibration process described in section 2.1.1, then perform three slow and three fast contractions holding a 1 kg weight in a 45 seconds interval. The test was repeated with a 2 kg weight. The percentage of contractions for 1kg in terms of time is shown in the (Figures 21 and 22).



(a) The Prototype and Delsys device on the arm.

(b) The experiment setup.

Figure 20: The Experimental Test to validate the bracelet software.

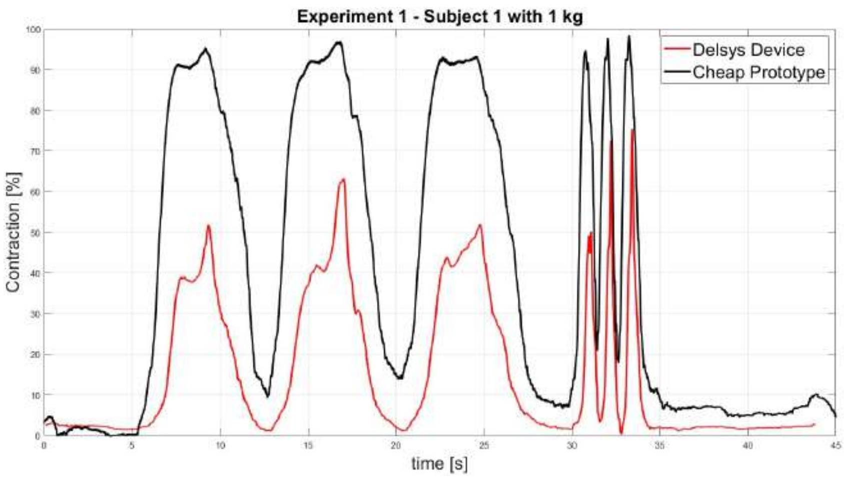


Figure 21: The 1 kg test.

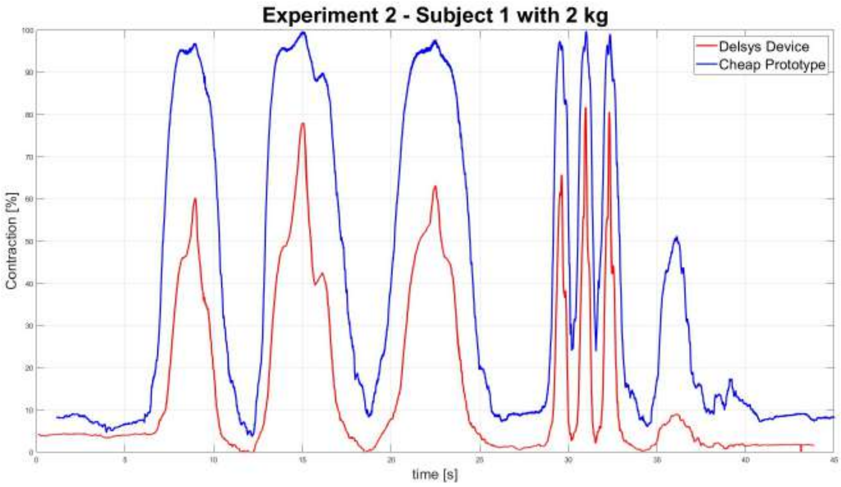


Figure 22: The 2 kg test.

3.2 The Final Prototype

After the preliminary tests and validations, the bracelet was mounted (Figure 23). The box is positioned on the external layer (Figure 24(a)), while the sensors - fixed between the layers with the adapters - have the spherical, responsible for the contact with the skin, part on the internal layer (Figure 24(b)). The case was projected with a sliding lid (Figure 25) and the band was adapted in order to increase its size, to fit in arms with larger circumferences (Figure 26). The battery case was printed but it was not fixed on the bracelet, the reason is explained in the discussion section. A video showing the working of the bracelet prototype is available at the link: The Bracelet Operation.

Figure 23: The bracelet closed.

(a) The front of the bracelet.

(b) The back of the bracelet.

Figure 24: The Bracelet opened.

(a)

(b)

Figure 25: The Bracelet Box.

(a) Original.

(b) Size Increased.

Figure 26: The Bracelet Size Adaption.

4 Discussion

A bracelet prototype was projected, developed and tested during this present internship. Thus, this section will describe and discuss some limitations, challenging and advantages of the final device and of the internship itself. The most limiting and challenging factor of this stage was certainly the time, where project, develop and test a new prototype in 8 weeks caused some unsolved problems. For instance, the printed PCB that was delivered has a short clearance between the pads and the traces. This caused a great deal of difficulty in the process of soldering the board, in addition to causing a short circuit that burned the battery charging traces on the PCB. For this reason, the prototype delivered does not have the battery case attached to the band. Another perceived problem in the mounting process was the position of the sensor connectors associated with the box case size, not allowing the circuit to enter inside the designed case, which had to be modified with a Dremel micro-grinder. This problem should be solved with both upgraded box CAD and PCB project delivered, however, for time reason, this new PCB and the box was not printed. Regarding the band, the short width caused problems to fix the sensors between the two layers, in addition to reduce the flexibility of the bracelet after the sewing (Figure 14). For this reason, it is highly recommended to use a wider custom band, where the sensor adapters can be positioned side by side with a high distance between each pair, as it is shown in the Figure 27(a). Finally, concerning the software, the calibration process was updated in the last internship week, because the first version worked getting a unique value read by each sensor and set as the min and max. Now, the calibration process considers the min and max of the data average during a defined time. After this software upgrade, the Wi-Fi communication is not working correctly, thus, it is necessary to debug this part of the code.

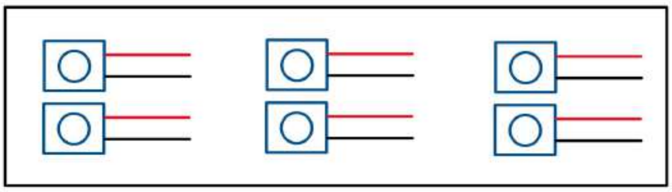
Despite these problems, even the prototype delivered presented good advantages, in terms of price, since it was built with cheap materials, but also in terms of performance. Since the device is compact, suitable for different users, intuitive to perform, well conditioned and tested. The sensor adapters projected solved different problems concerning the limitation of the FSR sensors, such as its fragility when folding, the contact with the skin, providing the correct reading and its firmness in the band fixing. In addition, concerning to the software, the prototype was capable to streaming via USB with stability the data in ROS2 environment with until 1500 Hz, what is more than enough to its application (probably 500 Hz). The prototype also was projected to work wireless, increasing the bracelet mobility. Finally, the simple and intuitive graphical interface for the calibration process here developed also solved the offset problem caused when FSR sensors are fixed on a closed bracelet. Since both minimum and maximum sensor value will depend on how much the user will tighten the bracelet on the arm initially. Thus, this procedure will map the contraction between these two read value. Thereby, the delivered prototype is ready to perform tests with Commercial devices, in order to compare, understanding better the limitations itself and probably proof with different experiments that a cheap bracelet has a sufficient performance for this application.

5 Conclusion

Despite all the limitations described in the discussion section, for a internship of 8 weeks, the prototype developed has good features and preliminary results, having in this way a high potential to be used in this application. Thereby, it is necessary to perform test comparing with validated commercial devices and also directly with robots.

6 Perspectives

For future work, it is possible to upgrade the prototype developed and perform more accurate experiments. Starting for the upgrades, as it was already mentioned, a wider bracelet is recommended for the next prototype (Figure 27(a)), increasing the flexibility of the bracelet.



(a) Recommended sensor layout.

(b) Delsys connector.

Figure 27

In addition, it is also possible to project a compacter version of the PCB presented in this work, where Surface-Mount Devices (SMDs) can be used to save space on the board together with smaller connectors for the sensors. Finally, depending on how the bracelet will be fixed on the exoskeleton, the device can use the robot power source energy, reducing more the PCB size.

6.1 More accurate experiments

Regarding the experiments, it is possible to test the bracelet with a real exoskeleton, performing the rehabilitation tasks in order to evaluate the prototype performance in the practices. Despite this, it is also possible to compare the prototype and the Delsys devices in two different ways: asynchronously and synchronously.

1. The first one can be done recording the prototype data following the experimental protocol presented in (Azhar 2022), then unplugging the sensor connector from the PCB and plug in the Delsys adapter, in order to compare the same sensors without changes in the bracelet position. For this, it will be necessary to create a cable where one end is connected to this Molex and the other connected to this Delsys male plug (Figure 27(b)).

2. The synchronous way is to reprogram the prototype to be trigger together with the Delsys device at the same time, where the bracelet will use three sensors and the Delsys the other three sensors. To do that, this device should be used.

Reference

Azhar, Mahrukh. 2022. Parsimonious technologies for sensing upper limb muscles activation. *Université de Strasbourg, Télécom Phisique, HealthTech* .

Bawa, Anthony & Konstantinos Banitsas. 2022. Design validation of a low-cost emg sensor compared to a commercial-based system for measuring muscle activity and fatigue. *Sensors* 22(15). 5799.

Casaccia, Sara, Lorenzo Scalise, Luigi Casacanditella, Enrico P. Tomasini & John W. Rohrbaugh. 2015. Non-contact assessment of muscle contraction: Laser doppler myography. In *2015 ieee international symposium on medical measurements and applications (memea) proceedings*, 610–615. doi:10.1109/MeMeA.2015.7145276.

Castellini, Claudio & Vikram Ravindra. 2014. A wearable low-cost device based upon force-sensing resistors to detect single-finger forces. In *5th ieee ras/embs international conference on biomedical robotics and biomechatronics*, 199–203. doi: 10.1109/BIOROB.2014.6913776.

Dabling, Jeffrey G, Anton Filatov & Jason W Wheeler. 2012. Static and cyclic performance evaluation of sensors for human interface pressure measurement. In *2012 annual international conference of the ieee engineering in medicine and biology society*, 162–165. IEEE.

Fleischer, Christian, Andreas Wege, Konstantin Kondak & Günter Hommel. 2006. Application of emg signals for controlling exoskeleton robots .

Lee, Park G. Cho D.Y. et al, S.H. 2020. Comparisons between end-effector and exoskeleton rehabilitation robots regarding upper extremity function among chronic stroke patients with moderate-to-severe upper limb impairment. *Sci Rep 10, 1806. https://doi.org/10.1038/s41598-020-58630-2* .

Martins, Sheila. 2023. World stroke organization. https://www.world-stroke.org/world-stroke-day-campaign/why-stroke-matters.

Mulas, M., M. Folgheraiter & G. Gini. 2005. An emg-controlled exoskeleton for hand rehabilitation. In *9th international conference on rehabilitation robotics, 2005. icorr 2005.*, 371–374. doi:10.1109/ICORR.2005.1501122.

Physiopedia. 2022. Hemiplegia. https://www.physiopedia.com/index.php?title=Hemiplegia&oldid=297252.

Prakash, Alok, Neeraj Sharma & Shiru Sharma. 2020. Novel force myography sensor to measure muscle contractions for controlling hand prostheses. *Instrumentation Science & Technology* 48(1). 43–62.

Prakash, Alok & Shiru Sharma. 2021. A low-cost transradial prosthesis controlled by the intention of muscular contraction. *Physical and Engineering Sciences in Medicine* 44(1). 229–241.

Saclay, Universite Paris. 2018. Faculte des sciences du sport. https://www.faculte-sciences-sport.universite-paris-saclay.fr/recherche/presentation-unite-de-recherche-ciams/mhaps/axe-1-facteurs-humains-interaction-humain/machine.

Sadun, AS, J Jalani & JA Sukor. 2016. Force sensing resistor (fsr): a brief overview and the low-cost sensor for active compliance control. In *First international workshop on pattern recognition*, vol. 10011, 222–226. SPIE.

Tekscan. 2023. How does a force sensing resistor (fsr) work? https://www.tekscan.

com/blog/flexiforce/how-does-force-sensing-resistor-fsr-work#:~:text=Shunt%20mode%20force%20sensing%20resistors,a%20special%20textured%2C%20resistive%20ink.

Toro, Sergio Fuentes del, Silvia Santos-Cuadros, Ester Olmeda, Carolina Álvarez-Caldas, Vicente Díaz & José Luís San Román. 2019. Is the use of a low-cost semg sensor valid to measure muscle fatigue? *Sensors* 19(14). 3204.

Wege, Andreas & Armin Zimmermann. 2007. Electromyography sensor based control for a hand exoskeleton. In *2007 ieee international conference on robotics and biomimetics (robio)*, 1470–1475. doi:10.1109/ROBIO.2007.4522381.

Appendix A - Tutorial How to Build the Bracelet

A The Tutorial Goal

In order to allow and facilitate the reproduce of this bracelet, the present section will describe step by step of how to mount the prototype based on the deliverable software, hardware project, CAD projects and the list of materials. All the bracelet project deliverable is available in this link.

A.1 List of Materials

Therefore, starting from the list of materials, to build one prototype, the Table 2 list each material used, as well as the quantities. In addition, each component name has a hyperlink with its reference.

List of Materials	Quantity
The band (Taille 2)	1 unit
The 3D printed box case	1 unit
The 3D printed battery case	1 unit
The FSR adapter	6 units for 6 sensors
FSR sensor	6 units
ESP32 NodeMCU Microcontroller	1 unit
Double-Sided Copper PCB Board	1 unit
Wires	$\approx$ 3 meters in total
Male Arduino Connector	6 units
Molex Connector 1	2 units
Molex Connector 2	2 units
Charger Modulus Board	1 unit
Bolts and Nuts M2-H10mm	18 units
Capacitor 100nF	9 units
Capacitor Electrolytic 1000 μF	1 unit
Resistor 30k Ω	7 units
Push Button	1 unit
Diodes Schottky (max 5V and 2A)	2 units
Tension Regulator MCP1700-3302ETO	1 unit
Power Switch	1 unit
Micro controller Adapter	1 bar
Lithium Battery 18650	1 unit
Battery Connector 1	2 unit
Battery Connector 2	1 unit
Black Velcro used in the Figure 26	$\approx$ 20 cm

Table 2: The List of Materials to build the Prototype

A.2 The CAD projects

Starting for the CAD models, it is possible to begin the project printing the pieces. This prototype uses three different types of 3D printed pieces: the adapters for the sensors, the box case and the battery case. Below it is shown the figure of each piece and the respective link to download the CAD file project made using CREO PTC software.

A.2.1 The adapters

The sensor adapter is constituted of three parts: the base, spherical part and the top (Figure 5(b) and 13(a). Therefore, to use one sensor is necessary to print the three pieces shown in the figures. For this project, six sensors are used, so six set of adapters should to be printed.

The CAD Project is called ”fsr_adapter_v2.prt” and is available in this folder.

A.2.2 The box case

The box case is constituted for the base and the sliding lid (Figures 16, 25).

The CAD Project for the base is called ”case_battery.prt” and the for the lid called ”lid_bracelet.prt”. Also, available in this folder.

A.2.3 The battery case

Finally, the battery case (Figure 17) is just one piece called ”case_battery.prt”. Also, available in this folder.

A.3 The Printed Circuit Board (PCB)

While the CAD pieces are printing, it is possible also start the PCB printing. The Proteus Project, composed by the circuit schematic and the PCB layout, is available in this folder. However, to print the PCB using the CNC machine, only the Gerber files, available here, should be sent to the manufacture.

A.3.1 Suggested welding order

Once the PCB is printed, and you already have all the components listed in the Table 2, the Table 3 suggests the welding order of each component. It is possible to use this files as a guide to solder each component in the correct spot.

Note: **It is also important to note each board side has the copper trace connected with the respective component, in other words, there are electrical components placed on the top of the board linked with a copper trace also on the top of the board, hence, requiring to be welded on the top.**

Electrical Components	Order
Microcontroller Adapter	1°
Resistor 30k Ω	2°
Capacitor 100nF	3°
Capacitor Electrolytic 1000 μF	4°
Diodes Schottky (max 5V and 2A)	5°
Push Button	6°
Charger Modulus Board	7°
Molex Connector 1	8°
Tension Regulator MCP1700-3302ETO	9°
Power Switch	10°

Table 3: The Suggested Welding Order

A.4 Mounting the Bracelet

A.4.1 The battery cable

Two wires, positive and negative, should be soldered to the battery and the another tip connected to this Molex 2 poles connector.

A.4.2 The sensor cables

Regarding the sensor cables, there are two cables:

1. One cable is responsible to connect 4 sensors. To do that, one tip is linked to this Molex connector, therefore eight wires are used. And the another tip is connected with a 2 poles male Arduino connector, shown in Figure 15.

   Note: The Molex connector presented on the PCB was projected to connect each sensor, as is shown in the Figure 28. No matter the sensor position, since it is basically a resistor.

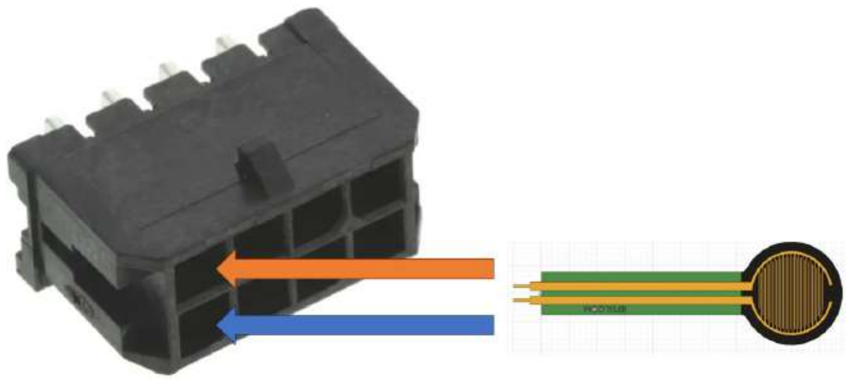


Figure 28: The sensor connection in the Molex connector.

2. The other cable also has one tip linked to this Molex connector, however, the another tip is divided: 4 holes of the connector are dedicated to the 2 FSR sensors that is missing, thus using the same Arduino male connection (Figure 15), but the another 4 holes should is dedicated to other power connection, as is shown in the Figure 29.

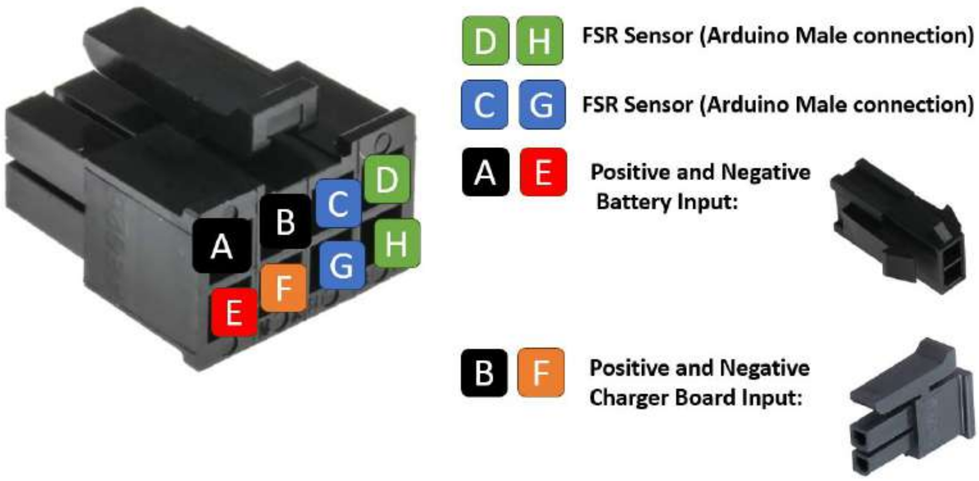


Figure 29: The second sensor cable.

Thus, after to build this cable, first verify again order of connection shown in Figure 29 and if it is possible confirm looking to the PCB cooper traces, to avoid any electrical short. Then, connect the (A-E) wire in the Battery Connector (section A.4.1) between the two layers of the band. And the (B-F) connector into the battery white plug from the Charger Modulus Board.

A.4.3 Fixing on the band

To fix the pieces on the bracelet, the following steps were performed:

1. Open the bracelet, cutting the sewing of the band, splitting, in this way, the two band layers.

2. Then fix, with the screws, the sensors together with the adapters inside the bracelet, as is shown in Figure 14.

   Note: **It is advised to use a wider bracelet, allowing in this way to position two adapters in parallel, in order to make the bracelet more flexible and be easier to organize the wires inside the bracelet.**

3. Then fix the box case and battery case on the bracelet, using the screws as well.

4. After that, the other band layer must be drilled in the center of the sensor adapter (Figure 30), in order to keep the sensor as is shown in 24(b). It was used the Dremel tools to make these holes.

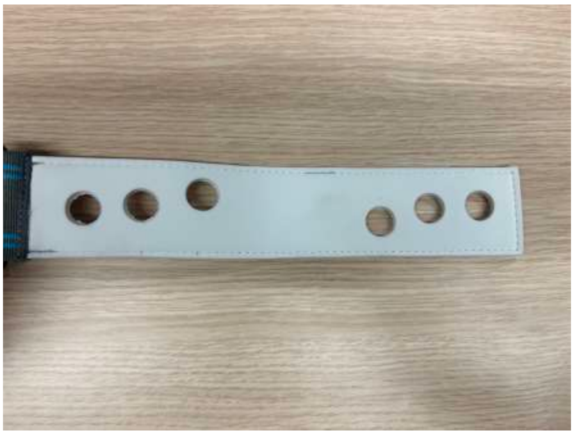


Figure 30: The white layer drilled.

5. Finally, the band should be closed, sewing both layers.

A.5 The software part

A.5.1 Prerequisites

Before to upload the codes, it is necessary to download and install some external libraries, as is shown below.

1. Installing ESP32 Board in the Arduino IDE.

   To install the ESP32 Board in the Arduino IDE, follow this Tutorial.

   Note: If you use Linux, probably, the command below should be run in the terminal, as explain in the Espressif Documentation.

```
$ sudo apt install python-is-python3
```

2. Arduino IDE External Libraries

   Open the Arduino IDE, and go to ”Sketch” -> ”Include Library” -> ”Manage Libraries”. In the ”Library Manager” window, search and install the follow libraries:

   (a) SPIFFS.h
   (b) ArduinoJson.h
   (c) WiFi.h
   (d) AsyncUDP

3. Python Code For the python code, it is necessary to install the follow libraries:

```
$ pip install tk
$ pip install customtkinter
$ pip install pyserial
```

4. ROS2

   Since the bracelet was projected to send topics into the ROS2 Humble environment, it is necessary to have the ROS2 installed in the Linux system. In the ROS2 documentation website there exist an Installation Guide.

A.5.2 Uploading the Code

Once the prerequisites are all installed, it is necessary to upload two codes (folder) in the right order into the ESP32 microcontroller. Thus, connect the bracelet into the USB computer port and upload these codes in the right order:

1. Upload first: initial_setup.ino
2. Then, upload: main_code_bracelet.ino

Done! The prototype is completely built after all these steps. Finally, to verify the working and use the prototype, the Appendix B is a section dedicated to explain step by step how to use the bracelet, once it is already built. The final result should be as demonstrated in the video: The Bracelet Operation.

Appendix B - Tutorial How to Use the Bracelet

B The Tutorial

This tutorial will explain how to use the bracelet prototype built in this internship work. However, this Appendix presupposes that the prototype is already done by following all the steps describe in the Appendix A. It is also important to note that to operate the bracelet with the default configurations, the computer used should have all the **Python** and **ROS2** prerequisites described in the section A.5.1. In addition, before to modify any parameters inside of the ESP32 microcontroller code, the **Arduino IDE** prerequisites 1 and 2 described in the section A.5.1 must also be met first. All the codes and project are available in this folder.

B.1 The Initial Setup

Since this bracelet prototype was projected to be use via USB serial or Wi-Fi, the first step to take is define which type of communication the bracelet will use. This step should be done once, ONLY IN THE FIRST TIME. After that, the microcontroller will save this information.

Thus, to set up the type of communication:

1. Connect the bracelet via USB into the computer.
2. Run the setup_bracelet.py code.
3. Choose the type of communication: serial or Wi-Fi Figure 31. For the WiFi option, the name and the password should be typed. Be sure that the WiFi network is turn on and working.

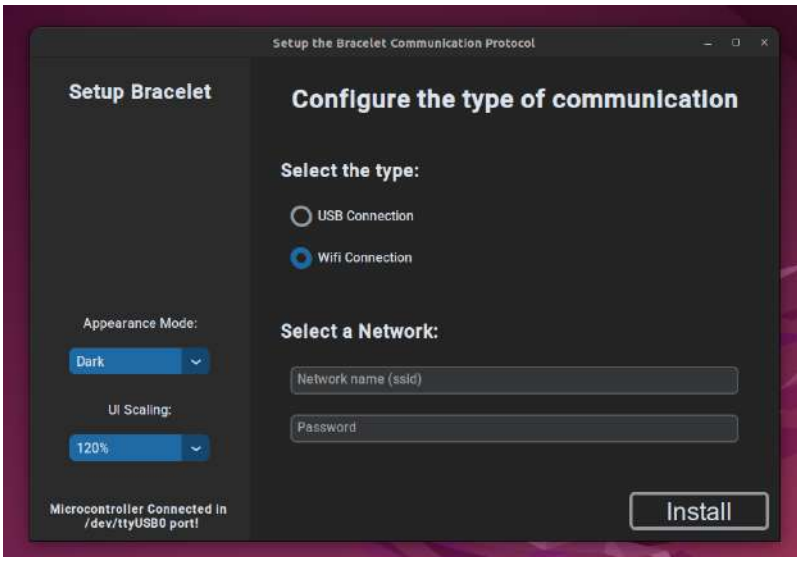


Figure 31: The Type of Communication Setup GUI.

4. Press Install and wait the Setup process to finish.

After that, the bracelet is ready to run following the section B.2.

B.1.1 The Type of Communication Reset Button

If the type of communication should be changed, such as change from USB to WiFi, change the WiFi password, etc. This prototype has a reset press button on the PCB board. To do that, just:

1. Connect the bracelet via USB into the computer.
2. Press the PCB button.
3. Repeat the process describe in the previous section, to set up again the type of communication.

B.2 Run the Bracelet Prototype

Once the type of communication has been set, to run the bracelet:

1. Connect the bracelet via USB into the computer if the communication is via serial, or turn on the battery if it is WiFi.
2. Open the terminal in the bracelet_ws/src folder.
3. Run these commands:

```
$ colcon build --cmake-args -DCMAKE_BUILD_TYPE=Release --symlink-install
$ source install/setup.bash
$ ros2 launch rrbot_bringup rrbot.launch.py
```

4. Follow the Calibration Process Instruction, described in detail in the section (2.1.1).
5. After the calibration the data is sending via topic in the ROS2 environment and the 2DOF robot should mimic the user arm contraction. As it is shown in this Video Simulation.
6. It is also possible to run only the node read_fsr.py in order to publish the topic into the ROS2 environment without launch the 2DOF robot. To do that, run these commands inside of the /src folder:

```
$ colcon build --cmake-args -DCMAKE_BUILD_TYPE=Release --symlink-install
$ source install/setup.bash
$ ros2 run rrbot_nodes read_fsr.py
```

Note: It is possible to record the simulation using the ROS Bag. Then visualize the simulation in the plot juggler, or using the plot juggler, export the simulation data to .csv file.