POLITECNICO DI TORINO

Master's Degree in Mechatronics Engineering

Master's Degree Thesis

Multi-purpose high precision tactile interface

Supervisors

Prof. Marcello CHIABERGE

Candidate

Massimo ALFIERI

Academic year 2023/2024

Summary

The scope of this thesis was to develop a device that could bring tactile feedback in the spectrum of already available sensorial interfaces but keeping the system easy to implement, cheaper to realise and adaptable to a vast range of applications. The idea that started it all is the same as the VR helmets adopted: to render an image 360° around the user, instead of building a screen so big the user could fit in, it is the helmet that mounts just two screens and senses the orientation of the user to render on the screens only a portion of the full image, giving the user the illusion of the immersion. The concept of this device is very similar: the tactile surface, in contact with the user's finger, senses the movement imposed to it and reproduces the surface of a previously uploaded image that contains all the height information. The user will perceive a surface much bigger than the "display" under their finger, we will refer to this as virtual surface. The main body of the device is detached from the tactile surface and is stationary. Inside of it is contained the mechanical system that acts to adjust the height of every dot on the tactile surface. The dots are the smallest quantity of virtual surface the device can reproduce, similarly as pixels for digital images. The main body is connected to the tactile surface by a flexible bundle of tubes that allows the transmission of the mechanical movement to the dots without limiting the movement of the tactile surface.

Table of Contents

List of Figures

Chapter 1

Introduction

In the first stages of conceptualization of this device the pillars that were the foundation for the development were: small dimensions, inexpensive components, resemblance with human experience. Keeping the dimensions compact profoundly conditioned the design decisions, impacting on the mechanical solution implemented. Problems derived from this imposed goal required a number of trial and error to tune tolerances, adjustment on the design or even complete change of approach. And with dimensions also the cost of the component was something to deal with. Inexpensive components usually do not offer the best performances or have good ratio efficiency-dimension so it was predictable that this would also impact on the design. The components implemented in the end proven to be effective and functional but not without reservations, this are considerations that would lead to different and more conscious approach in future development. The user experience aspect, the resemblance with the tactile experience humans feel often without even realise its depth and complexity, was one of the most complex aspect to take into account. Make something work already offers its set of technical challenges but with user experience the technical aspects are useless if the final product is hard or unnatural to use. This is Masochist's Teapot concept: an object or tool that contains all its essential aspects, all working properly, but arranged in a way that makes it impossible to use for any kind of user. This was the biggest challenge to overcome, the one that goes beyond technical skills and instead asks for empathy and imagination.

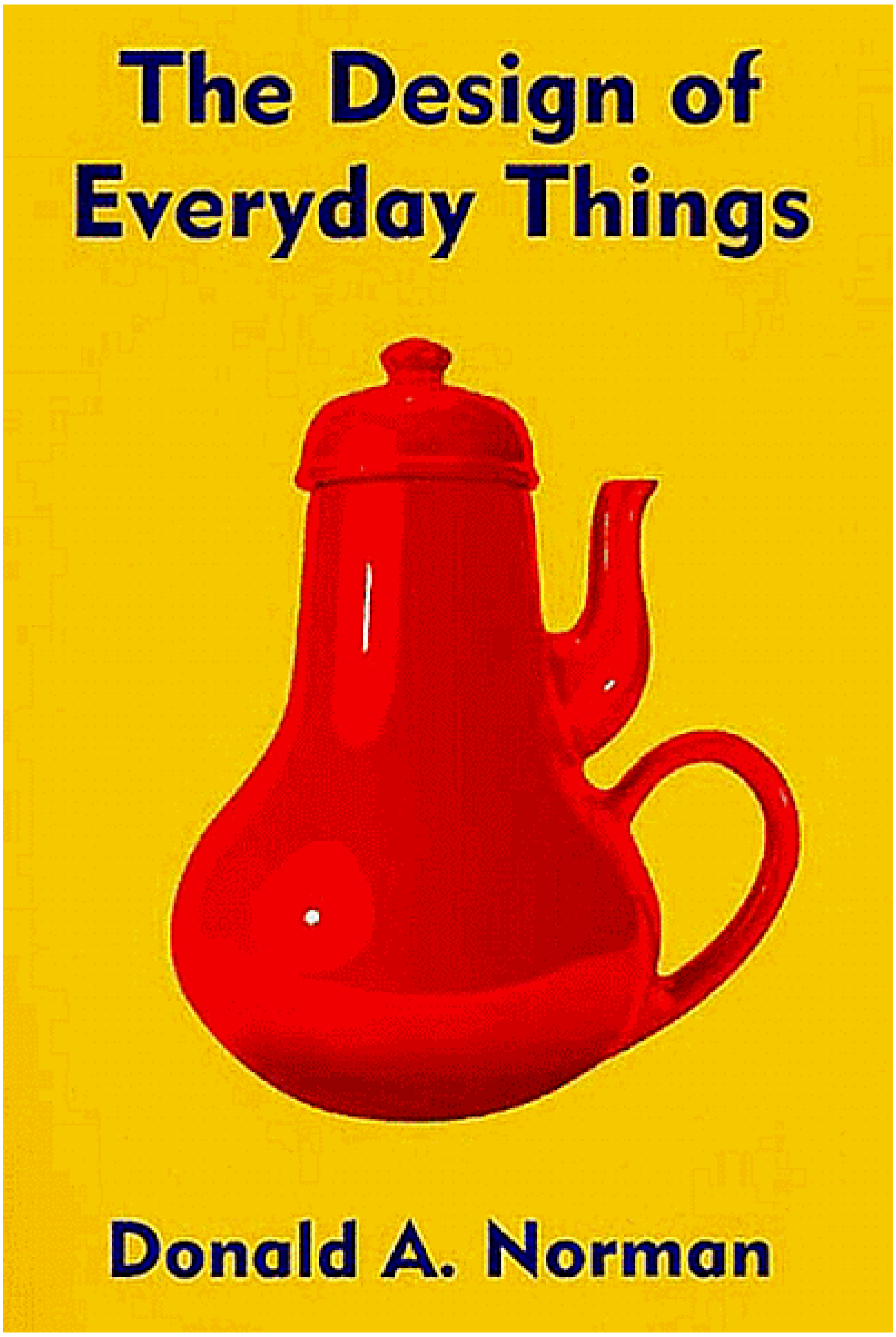


Figure 1.1: "The masochist's teapot" book cover

Chapter 2

State of the art

2.1 Existing technology

The most known and spread devices that use interaction with tactile sense are braille readers. While smartphones use just vibrations to interact with users, Braille readers are made of Braille cells that can modify their conformation to reproduce all the braille characters.

Figure 2.1: "Chameleon 20" by American Printing House

However these devices are very expensive (base price 2000 USD) due to the complex piezoelectric unit composing the cells. Even in their smaller iterations they are considered not very practical due to the limited number of cells they offer, the overall dimensions of the device and not many functionalities. Since vocal synthesis and vocal assistants have been well implemented in computers and smartphones, Braille displays have been substituted. Even though the text reproduction for visually impaired people has found cheaper and easier ways to be approached, there are no devices able to dynamically reproduce images, textures or large surfaces. There exist a number of academic projects and research[1] that, over the years, tried to tackle the problem but the results, although impressive, were not scalable, easy to implement or cheap. The actuation system has always been the main problem to all these studies due to the dimensions of the actuators, difficult to scale down and achieve in small form factor. An interesting device is the Optacon[2], commercialized for the first time in 1971 and discontinued in 1996. It was able to reproduce on a small finger interface what the probe the operator was framing on a surface, usually on a written page. The screen interface was able to reproduce only with two states, high or low with no values in between, and was placed fixed on the main body of the machine. One hand would have to stay still with the finger tip on the display while the other was moving the probe on the page. Despite the interesting concept it was hard to learn how to use proficiently for the users, there was no elaboration of the image so even imperfection would have been reproduced and testers reported it was very tiring and disturbing since it was continuously emitting high pitch buzzing noise.

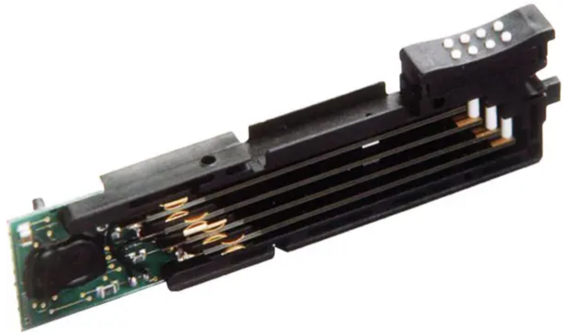


Figure 2.2: Braille module by HOERBIGER company

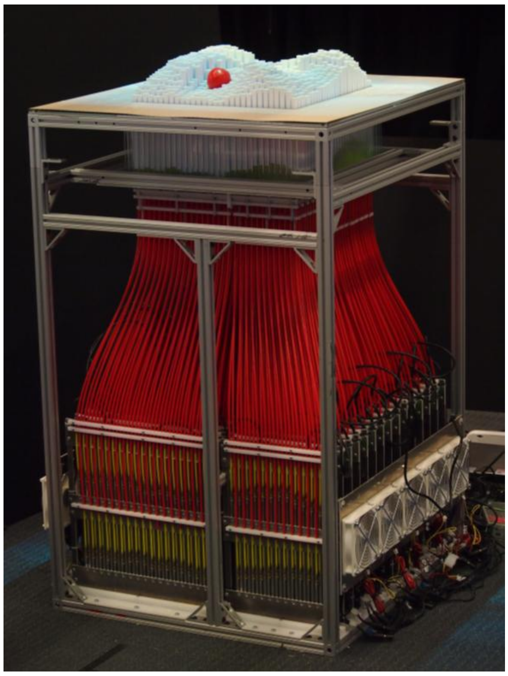


Figure 2.3: "inFORM" by MIT Media Lab

Figure 2.4: "Optacon"

What these past explorations made clear is the importance of having dots that are able to move vertically in a wide range of heights unlike the bistable positions the Braille cells devices are able to.

Chapter 3

Design phase

3.1 Dot visualization

As said in the introduction the dots are the minimum amount of data this device can "render". The matrix of dots is placed under the finger of the user and then renders a portion of the virtual surface.

3.1.1 Dot matrix

The dot matrix is the interface between the user and the device itself. The implemented matrix is composed of five rows and four columns (4x5) in a orthogonal grid but just nineteen dots are active over the twenty available slots. The reason for this decision has roots in the first conceptualisation of the prototype, nineteen dots arranged in a hexagonal pattern can reach a higher density matrix and so reproduce a higher definition virtual surface. Unfortunately the 3D printer used for the realization of the matrix was unable to reproduce such level of detail so the easiest way to accommodate for this was to rearrange the matrix from hexagonal to orthogonal. The result is a matrix that does not have the desired definition but is still able to give an idea of the functionality. Its dimensions are also quite similar to the standard arrangement of the dots in the braille cells, both six dots and eight dots cells.

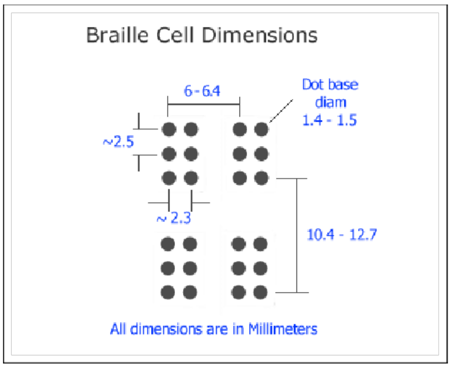
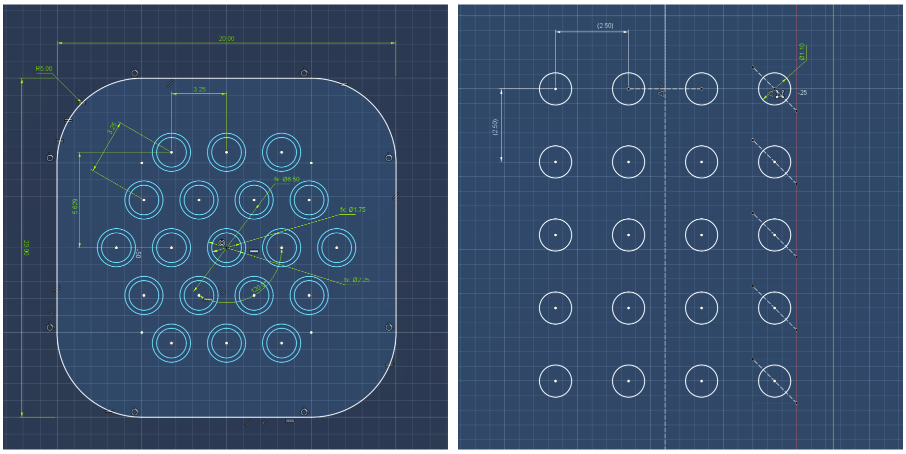


Figure 3.1: Standard Braille cell dimensions

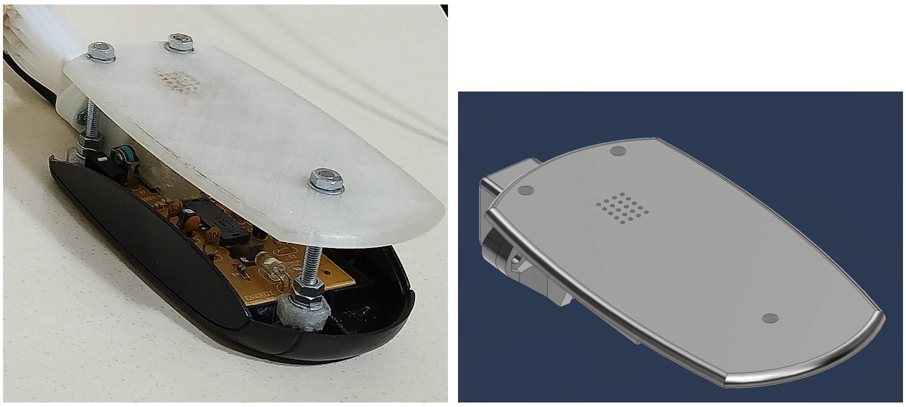


(a) Hexagonal dot matrix **(b)** Orthogonal dot matrix

Figure 3.2: The first (a) design compared to the final (b) design

3.1.2 Handled peripheral body

The matrix is placed above the peripheral the user will hold in their hand, placing the index finger over the dots matrix. The body of the peripheral is supposed to function as a common computer mouse, it is able to sense the relative movement over the support surface to determine which portion of virtual surface to render. For the realized prototype a wired computer mouse was used and modified just to achieve an approximation of this peripheral. On top of it a custom 3D printed shell was mounted to substitute the original one. This top shell has no button interface since they are not supposed to be used for this scope, instead in the center the dots matrix takes place approximately where the index finger should lie. The prototype was modeled just to prove the concept of the device so no particular attention was reserved for the ergonomic aspect of it. Although this would be something very important for a final product: comfortable shapes, pleasant materials, well balanced weight, custom dimensions and more are key aspects for a device that is supposed to be used, held in hand, for time spans that could vary quite a lot. The dots that go through the shell and go in contact with the index finger are nylon wires with a diameter of 1mm. Using nylon wire to interface the user was a practical decision, not comfortable but effective in simplifying the development. The wires are directly used as a mechanical transmission meant to connect the main body, where the active mechanics lies, to the interface peripheral. Later in this document will be explained how this choice was key to enable the decoupling of these two elements constituting the main feature of the device itself.



(a) Tactile mouse photo **(b)** Tactile mouse CAD model

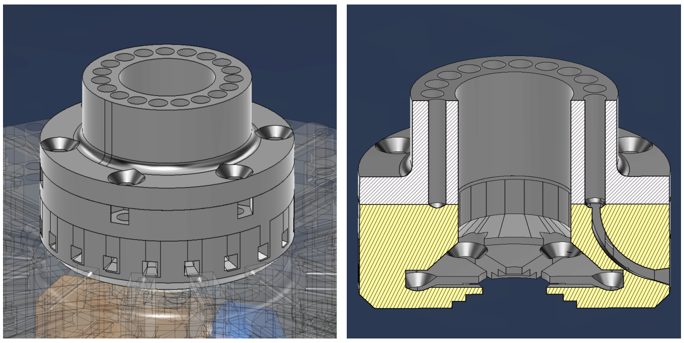
Figure 3.3: Real and virtual mouse side by side

3.2 Movement transmission

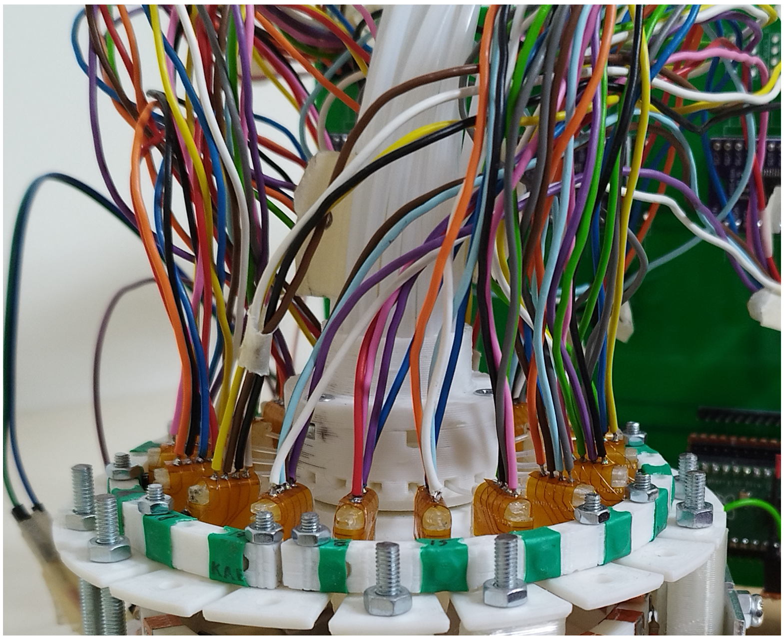
The nylon wires that on the interface side constitute the dots and on the other side are actuated by the main body, connects both parts through PTFE (teflon) tubes, one for each wire. Ideally the tube's internal diameter is as close as possible to the diameter of the wire, enough to allow it to slide through without constriction but not enough to have room for the wire to bend and accumulate over curves. A difference in the order of magnitude of hundreds microns would be the perfect scenario. If the tube has the ideal dimension and is fixed between two points, even if they are not rigidly connected to each other, the wire pushed inside one end should travel smoothly through it and come out to the other side with comparable amount of force and in the same amount of displacement. This is called a "bowden tube" and this principle is the same that bicycle brakes use to transmit the force exerted on the brake's lever to the pads against the tire. In this instance the wires are pushed by the actuators in the main body into the tubes escaping through the interface on the peripheral. This allows to detach the peripheral from the actuators keeping it small, lightweight and free to move in space, an essential feature for the device to work properly.

3.3 Round table

The role of the nylon wires has been already described, in this paragraph will be introduced how the actuation system works. Each of the nineteen wires is convoyed and collected in a cylindrical crown that redirects the wires parallel to the surface in line to connect to a cursor free to slide radially inside a slot on a circular plate.



(a) Wire crown (b) Wire crown (section)



(c) Wire crown 3D printed

Figure 3.4: Real and virtual wire crown side by side

The position of the cursor determines how much the relative wire protrudes on the interface, their ratio is proportional and with a ratio equal to one. To precisely manipulate the cursor a feedback position is needed, for this purpose a magnetometer is placed on it and, at the far end of the slot, a magnet is fixed in place. Nineteen permanent magnets are placed in a circle fixed on the plate

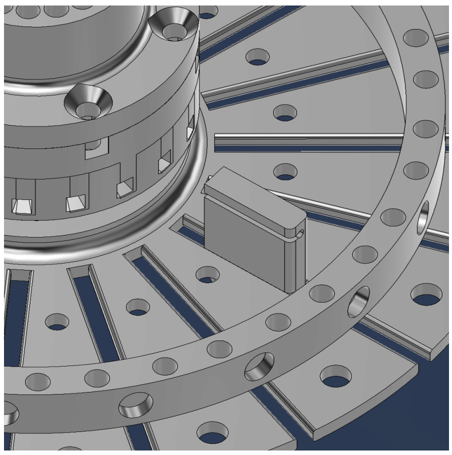
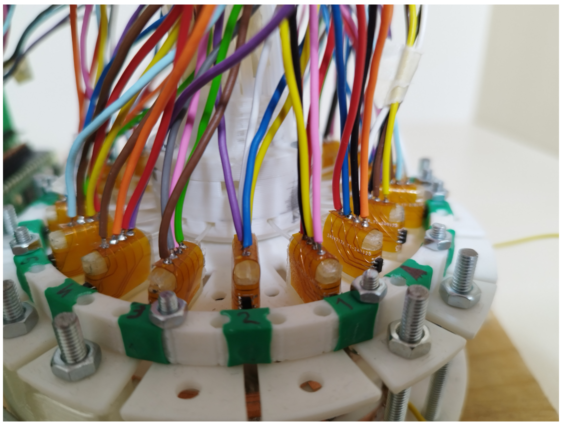
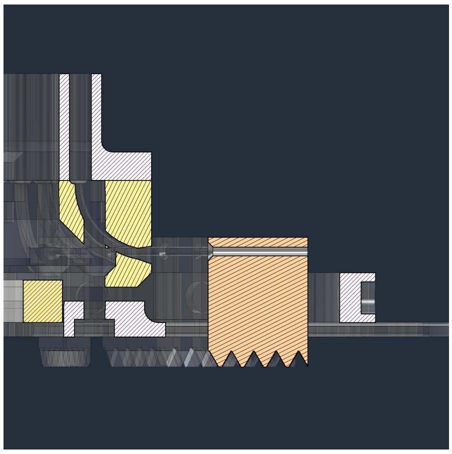


Figure 3.5: Cursor slot closeup in CAD

so that the magnetic field is stationary in time during the use, reliable for the magnetometer to determine their displacement. The slots allow an excursion of about 5 mm for the cursor and are all radial and equally spaced on the circular plate.



(a) Cursors on the round table



(b) Cursors on the round table in CAD (section)

Figure 3.6: Real and virtual cursors on the round table side by side

The magnetometers have a SMD packaging and are soldered on a flexible PCB that wraps around the cursors and is glued in place. Is important that the magnetometers are all about the same position on the cursors so that the measurements of the magnetic fields are all comparable. These are all digital sensors communicating to the microcontroller via I2C protocols. Since all the ICs have the same I2C address it has been necessary to utilise three eight channels I2C multiplexers to read from all the sensors using just one I2C channel of the microcontroller. It is worth mentioning that the multiplexers do slow the communication down but not in a noticeable way.

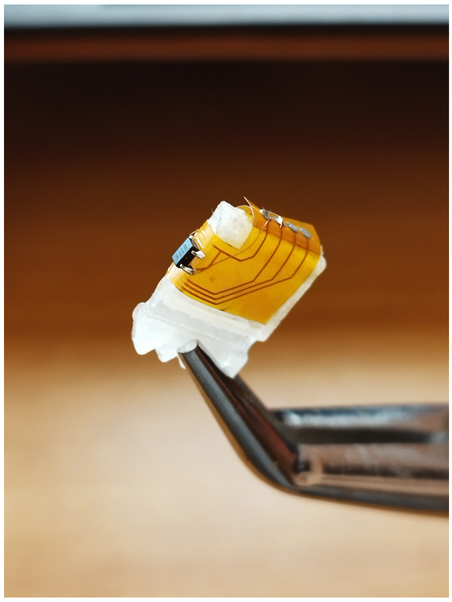


Figure 3.7: Selector unit in CAD

3.4 The selector

Underneath the round plate lies the actual actuation apparatus: the selector. The selector is a contraption that is coaxial with the centre of the round plate and is able to rotate about this axis by means of a DC motor. A second motor is placed on the selector, orthogonally oriented to the previous rotation axis.

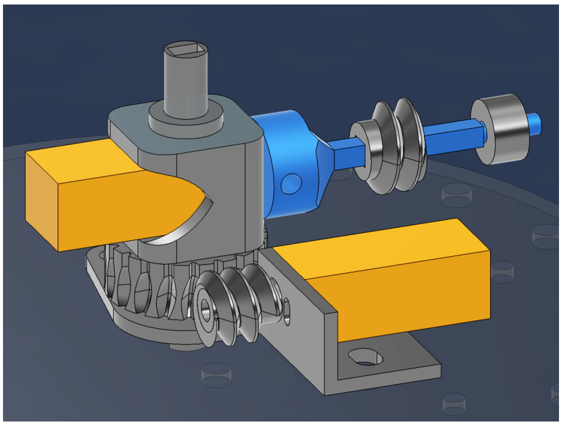


Figure 3.8: Selector unit in CAD

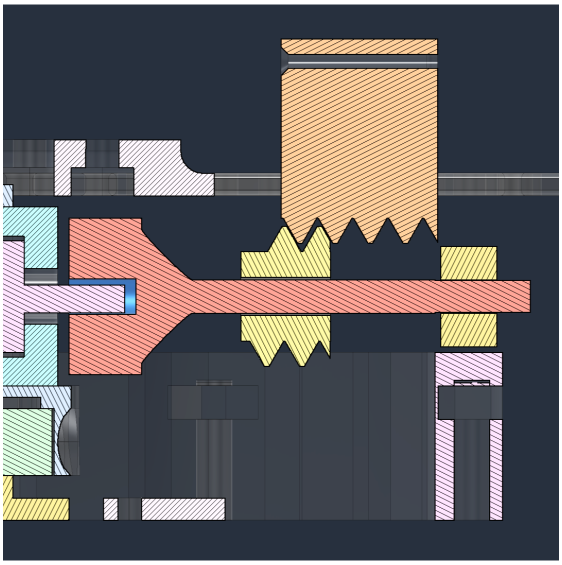


Figure 3.9: Selector unit, working principal, in CAD (section)

The motor shaft is extended with a fixed plastic shaft with square section. The peculiar section allows the worm gear placed on it to rotate when the motor is activated and slide along the shaft to allow small accomodations in the matching of the gears. The lower part of the cursors is constituted by a gear rack that matches the worm gear when placed parallel underneath. So when the horizontal motor is aligned with the slot above the worm gear matches the cursor, rotating the worm gear will force the cursor to move along the slot modifying the displacement of the wire.

The selector rotating engages the chosen cursor. To facilitate the matching of the worm gear to the gear rack of the cursor the square section shaft was kept as a separate component, the worm gear rigidly turns with it but still being able to freely move along. As mentioned before the whole system is able to rotate about the central vertical axis by a DC motor. As the one that moves the cursors this one too uses a worm gear. This enables to move the nineteen cursors with just two DC motors. The motor lays flat fixed on the bottom plate and its gear couples the selector corpus gear, this reduces the overall occupied space. The selector body spins around two ball bearings, a small one on top and a bigger one on the bottom. The bottom bearing is also used as electrical contact for the motor that moves the cursors, the other contact pass through the bearing and the bottom plate through a still rod catched by a brush. This allows the motor to spin continuously over 360° without tangling any wire. To determine the orientation of the selector a rudimental, yet effective, feedback position system has been implemented. To move a cursor the worm gear must be engaged and the shaft must be aligned, at the end of the shaft is mounted a small ball bearing that rolls over a plastic ring to sustain the load. On this ring, aligned with the cursors' slots, a couple of contacts have been placed, one is an analog input pin of the microcontroller and the other is a voltage level. To every position a different voltage level is assigned. When the ball bearing of the selector touches both contacts, closing the circuit, the voltage is read by the microcontroller and the absolute position identified.

Figure 3.10: Selector unit, photo

3.5 Electronics

The electronic apparatus of the prototype has been designed with the objectives to contain costs and to use components easy to hack, customize and upgrade. The following diagram illustrate the hierarchy of connection of the hardware. The same order will be used to explain the components.

The used micro-controller is a RaspberryPi Pico (RP Pico): a very cheap

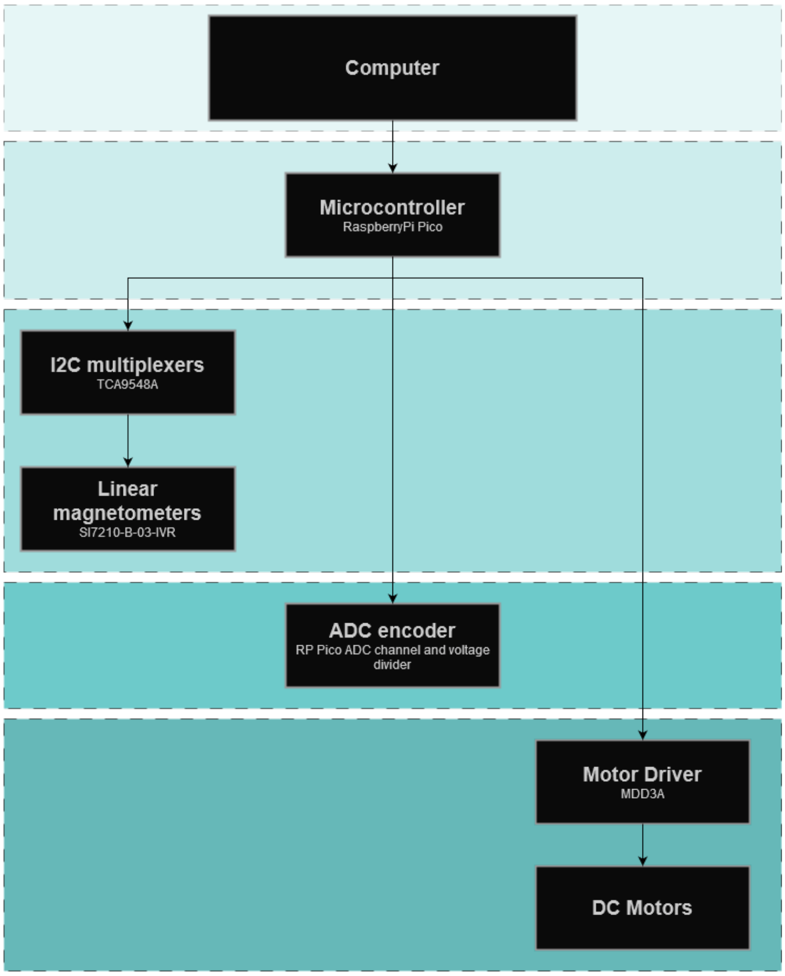


Figure 3.11: Electronics connection diagram

micro-controller, widely supported, with the possibility to run firmware in MicroPython, CircuitPython, C and C++. MicroPython has been chosen as main language as it is a very popular and easy to use high level programming language, maximizing the possibility of coworking and expansion in the future. The board mounts a RP2040 with a good number of I/Os (more than enough for the actual prototype) and a vast range of communication protocols compatibility (I2C, SPI, UART, etc...). More information can be found in the documentation [3]. The RP Pico is used connected the computer. A custom software (more information in the next section) through serial communication sends formatted arrays of numbers that represent the height of each dot. The micro-controller checks the actual state of the dots verifying the need of adjustments. As said before the positions of the dots are measured by magnetometers placed on the cursors.

The SI7210-B-03-IVR is an integrated magnetometer that presents a good performance-cost ratio in a small size, in fact is contained in a SOT-23-5 SMD package. The sensor has prove to be easy to use, the reading can be computed in few passages and communicated through I2C to the microcontroller with a maximum communication speed of 400000 Hz (although this was not the communication speed used in the prototype) in a 12 bit resolution. All these ICs are mounted on custom flexible PCBs wrapped around the cursors, helping the mounting and the connection to the RP Pico. Since all nineteen magnetometers use the same I2C address for the communication three I2C multiplexers (TCA9548A) were used on the same channel.

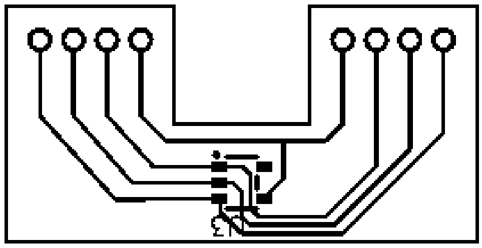


Figure 3.12: Flexible PCB schematic

Input A (M1A / M2A)	Input B (M1B / M2B)	Output A (M1A / M2A)	Output B (M1B / M2B)	Motor
Low	**Low**	Low	Low	Brake
High	**Low**	High	Low	Forward*
Low	**High**	Low	High	Backward*
High	**High**	High	High	Brake

Figure 3.13: Motor divers logic table

The other sensing layer is occupied by the position feedback system of the selector. It does not use any dedicated external sensor but instead the onboard ADC of the RP Pico. In correspondent position of the cursors the selector mechanically closes the circuit between two contacts. Each couples of contacts are independent but connected to the same analog pin of the microcontroller. Each couple of contacts is a voltage divider that, when closed, sends a unique voltage level to the ADC. This method allow to determine the position of the cursor without other sensors in a cheap but effective way. In the third layer lies the motor driver directly connected to the DC motors. This is a quite simple component that eases the control of the motors. It is in fact a double H-bridge circuit that step up the voltage and current sent to the motors. The controls of the motors are separated in two digital channels that follows the logic shown in the figure and it allows a PWM speed control. To power the motor an external power supply was needed and in the prototype a common Li-Po cell NCR18650D (5.5V output voltage) was used to ease the transportation of the prototype while demonstrating that a corded power supply is not necessary.

It was important to chose small lightweight motors to reduce the volume of the prototype. The gearbox that come with the motors increases the torque while keeping the rotational speed good enough for the application. The microcontroller, the three I2C multiplexers and the motors driver are all mounted on a custom PCB securing the connections between components and allowing replacement if needed.

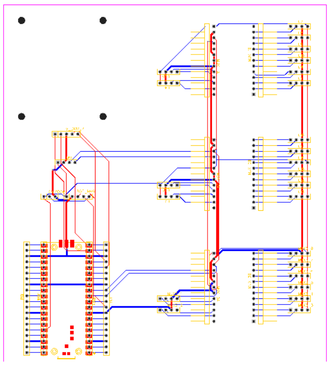


Figure 3.14: Main board PCB schematic

3.6 Software

On the software side two different scripts were written to work in parallel: one on the microcontroller and the second on the computer it was connected to. Respectively they were written in MicroPython and Python.

3.6.1 Computer script

The computer Python script handles the conversion of the input image to data to be sent to the microcontroller. All the image manipulations are executed with the OpenCV Python library. To be precise the script accepts images with PNG and JPG extensions and with any resolution or ratio. At first the image is converted to a proper gray scale format if it is not already, then it is scaled down or up, depending on the original resolution, to comfortably fit the computer screen. This process is automatic. Once the preprocessing is done a dedicated window is opened on the screen containing the image and three dial bar. At this point, in correspondence of the mouse cursor, will appear a small square through which the underneath image will be "pixelated" in larger pixel arrangements. This square lens follows cursor and its dimension (height and width)

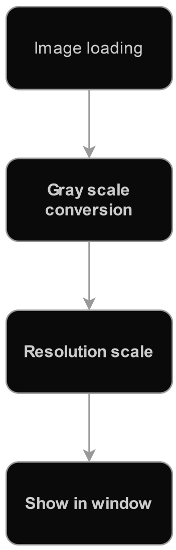


Figure 3.15: Image elaboration procedure diagram

and resolution can be adjusted by the three dial bars. The content of the square is raw data that will be send to the microcontroller. The number of pixels will constitute the length of the data array and their gray scale values (0 - 255), the values of the entries. Their order of placement in the array corresponds to the order of representation in the dot interface on the peripheral. To reproduce an image through the virtual surface it is possible to use ready made neural networks trained for monocular depth estimation[4]. These neural networks can take any picture as an input and return and return a height map gray scale image to feed the Python script and reproduce the virtual surface of the depth information of an image.

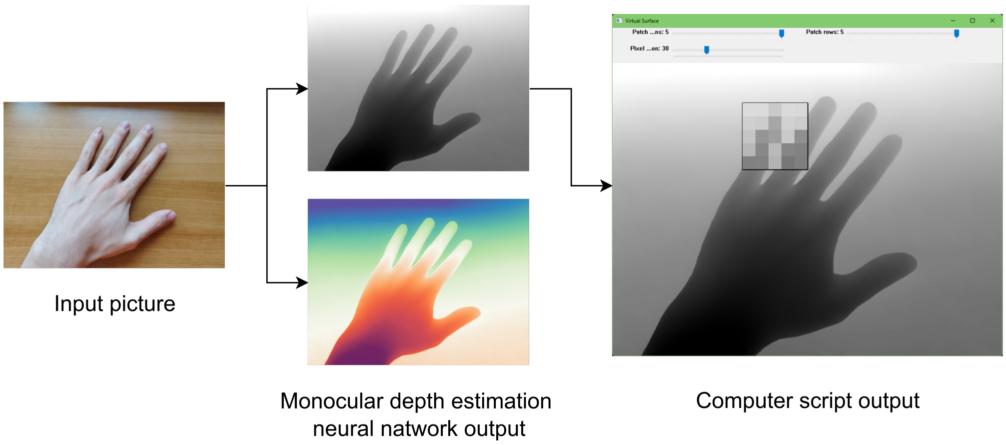


Figure 3.16: Image handling workflow

3.6.2 On board firmware

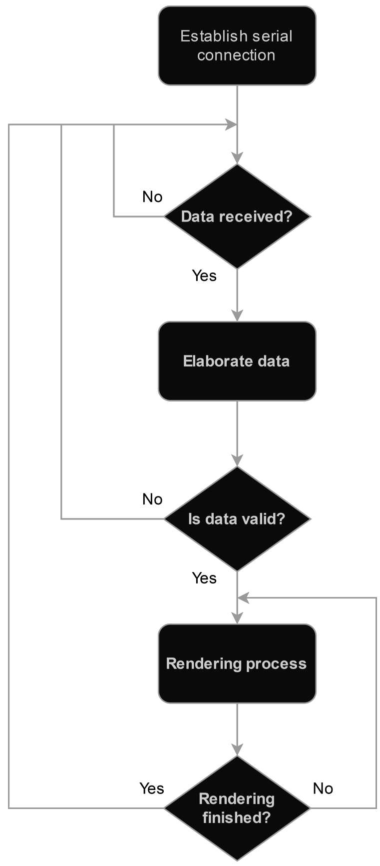


Figure 3.17: Firmware functional diagram

The on board firmware starts in the instant the board is powered on. It checks for the serial connection to be present and awaits still. If the connection is successful and the data is correctly received the actual elaboration starts. The received data should have the form of a string of numbers, each number going from 0 to 255, spaced by a comma sign. The firmware will act differently depending on the number of components of the array: six numbers will be identified as a braille cell 2x3 and represented accordingly in the top centre of the interface; eight numbers will be identified as braille cell 2x4 and represented in the centre; nineteen or twenty numbers will be represented in the full interface (the twentieth won't be reproduced); all other combinations or incorrect format of the data will be ignored leaving the firmware still waiting. The value of a single entry of the array represent the grey scale value of the original pixel. This number is proportionally scaled to be reproduced on the interface since every dot position lies in a spectrum that goes from 0 to 16384. Firstly, if the data is valid, the state of the dots is compared to the desired state, in case of mismatch the selector is moved to the closest dot to be modified. Once the selected dot is correctly updated the selector moves to the closest next one until all the dots are correct. The position of the dots allows a small tolerance imposed by code to overcome the noise that affects the sensors signal. After the update the firmware awaits the next data that will be reproduced only if it is valid or different from the last one executed.

3.7 Framework, 3D modelling and realisation

The mechanical design of the device has been handled in Autodesk Fusion 360. All the parts were modelled with an iterative process, since they have to fit and work together, they were 3D printed alongside with the modelling process then tested to check tolerances and adjusted back in the CAD.

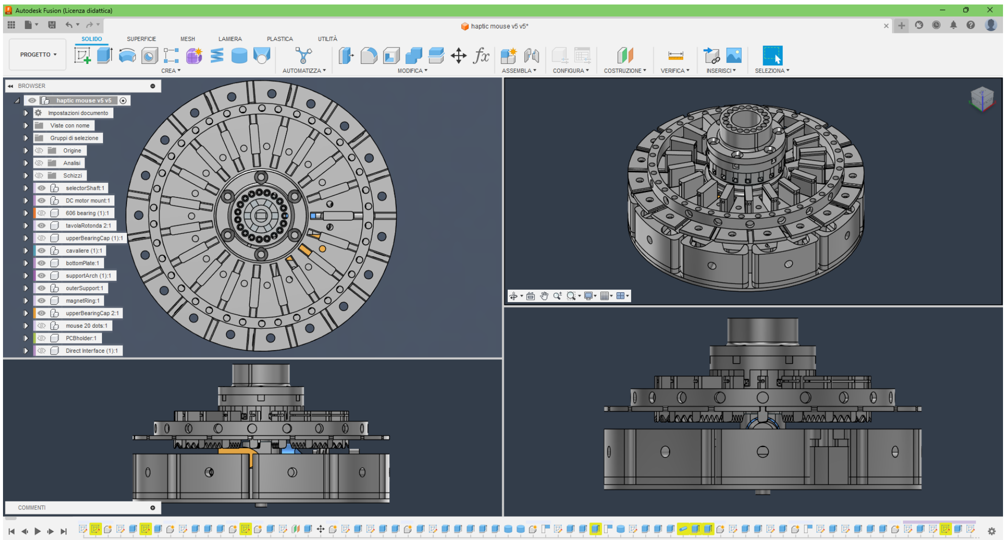
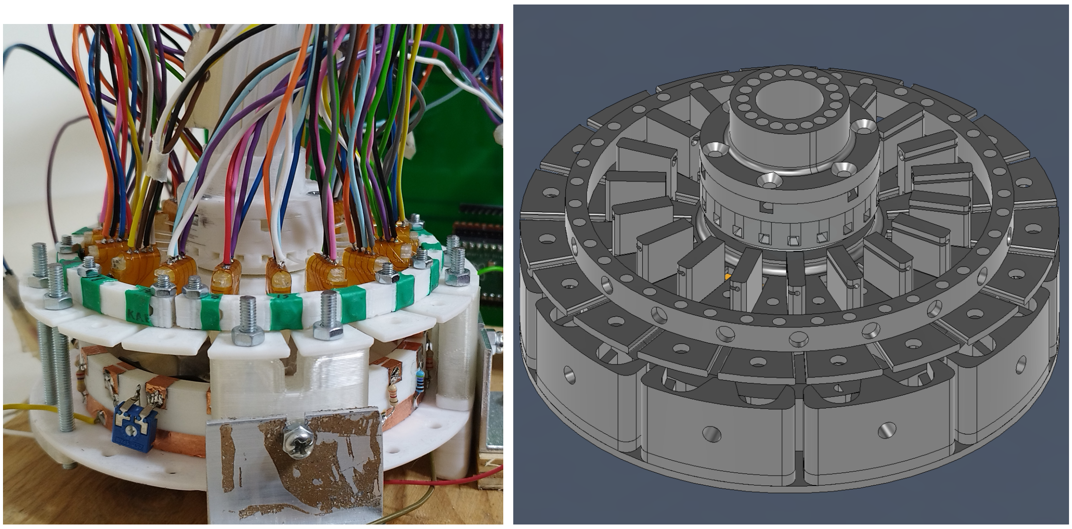


Figure 3.18: Design CAD screen

It is important to report that the ability to 3D print all the parts were crucial for the realisation of the mechanical components but FDM printing revealed to be not quite good enough to achieve precise small components. In FDM printing process the layer height represents the resolution of the final part. The smallest moving components of the design were printed with 0.2 mm layer height but this has been revealed to be not enough. This layer height was still comparable with the dimension of the parts, resulting in mechanical problems. Parts, gears in particular, tend to get stuck during matching stalling the whole mechanism. For the sake of this prototype and for tools availability, this ended up being one of the unsolved problems. In future iterations finer printing methods are advisable, like SLA or SLS 3D printing. The use of plastic, PLA, for all the parts constituted another problem. Shafts, crucial to the mechanism, tend to bend under pressure and torque and, the already mentioned, gears after all the testing show signs of deterioration.



(a) Main body, photo **(b)** Main body in CAD

Figure 3.19: Real and virtual main body, side by side

Chapter 4

Development problems

In this chapter will be reported some of the major problems encountered during the development and testing that lead to the final design described in the previous chapters.

4.1 Motion tracking

At first the handled peripheral was intended to determine its relative movement using a IMU sensor with 9 degree of freedom. The chosen integrated circuit was contained in SMT form factor. Without proper tool, such components are very difficult to if not impossible to solder and operate. I manage to solder all its pins by hands but in the end during the first test with I2C communication there has been no way to make it work, even though the I2C address scan seamed to work properly once accessing the registers there was no usable response from the IC. After the failed attempt with the IMU I tried to use image recognition and ARuCo markers. The initial tests seamed promising, the OpenCV library after some hiccups started to work properly but the noise of the position and the various problems in different light condition made this system quite unreliable without a lot more work to do. In the end I settled onto the computer mouse: an easy, reliable, cheap and effective solution.

4.2 Bowden system

I have already expressed some concern about the bowden tubes system. Two major problems raised around it and both involves precision in the material. First the tubes and wires: to achieve the ideal accuracy for the application the inner diameter of the tubes should be as close as possible to wire and both of them should be ideally flexible but inextensible. Second the components that match the systems:

the 3d printed parts have been revealed difficult to print properly. The machine at my disposal had a number of problem related with the geometry of all the parts that had to house the nylon wires. SLA or very fine FDM 3D printer nozzles should have been more indicated for this kinds of components that in the end introduced a lot of friction in the system.

4.3 Sliding cursors

4.3.1 Round table

After a number of tries I managed to find the optimal tuning for the round table to be efficient and easy to print. But, with the prototype fully developed I have some useful observation. First is the number of cursors the round table gather. Nineteen cursors has demonstrated to be a large number to manage, instead it would be better to reduce this number and achieve the required dots count with two bodies cooperating. This would increase the cost of the final device but exponentially decrease the problems. A second problem to note is the friction on the cursors. As said before, due to parts not precisely fabricated or impossible to find with the suitable requirements, there is a lot of friction exerted on the cursors and the wires and not in a consistent way, some cursors show friction concentrated on the handled peripheral, some on the opposite side where the crown redirects them, some on the cursors themself. Supposing to achieve a frictionless apparatus in a future iteration there will be the need to apply a controlled amount of friction to the cursors to keep them in place while the selector operates. In the current design is already present a mounting point to position a rubber o-ring that would have acted as dumper. This was left empty in the final device because of the problems already mentioned.

4.3.2 Cursors

Despite all the problems that plastic mechanical parts (gears, racks, shafts, etc...) inherit from the material, the match between gears and cursors racks works properly. However, in a next iteration, I would proceed making the cursors wider to ease the matching withe the selector gear. Even the magnetometers mount would benefit from this. In this version of the prototype flexible PCBs were use to mount the ICs and wrap around the cursors. This revealed to be a poor choice of design since the tight folds of the PCB break the connection with the soldering pads. In the end it has been necessary to solder by hand the pins bypassing the PCB tracks. Flexible PCB are also difficult to place firmly on the cursors making their position inconsistent between pieces. In a next iteration thin solid PCBs would be preferable to have sturdier connections.

4.3.3 Selector orientation

In the first half of development it was expected to be used a two axis magnetometer to determine the orientation of the selector. The magnetometer was placed under the wire crown, in the center of the round table while the magnet was embedded in the central shaft of the selector. Even though the IC was easy to use, configure, read from and the preliminary tests were positive, with the increase of complexity in the design error in the shaft started accumulating. A small wobble in the shaft was enough to cause the reading to be unusable. After few tries trying to fix the mechanical problem I decided to change approach drastically and implement the solution reported in the final design.

4.3.4 I2C communication

As extensively explained before, the magnetometers mounted on top of the cursors uses I2C communication. Since all of them have the same address three I2C multiplexers were used to handle the readings. A wide number of tests reveled that something in the chain that starts from the microcontroller and ends to the magnetometers causes a communication error. It was empirically proven that the communication frequency, if set to low, causes the ICs to go out of sync with the microcontroller. The crash of the program was happening quite often, immediately after the call of the reading, making the firmware impossible to execute. Various specific tests tracks the error on the microcontroller end. It is however difficult to determine if this behaviour is caused by the RaspberryPi Pico itself or the MicroPython image mounted on it. The effective solution found was, in the code, to catch the error when it occurs and restart the I2C communication.

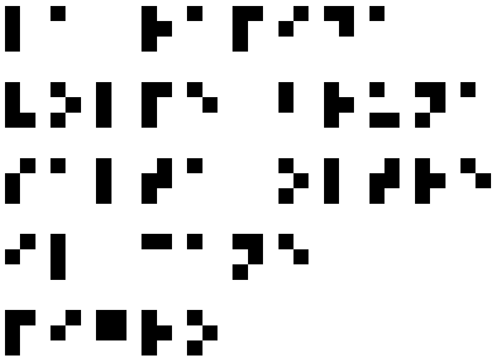
Chapter 5

Test results

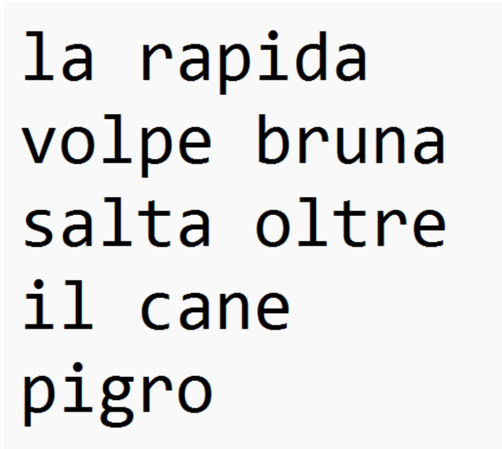
5.1 Beta testers experience report

Once the prototype reached a sufficient level of functionality to ensure the founding concept had some potential it was subjected to testing. In collaboration with UICI Siracusa it has been possible to set a testing session. Three testers were involved, president of the association Carmelo Fangano and two associates. Mr. Fangano and one of the two associates being braille readers were able to test the braille reproduction functionality, the third tester was not able to read braille but provided useful suggestions and questions due his experience with assistive technology. In the session two major tests can be identified: in the first part few braille letter were reproduced on the prototype interface and perceived by the testers; in the second part two "images" were reproduced on the interface to understand how suitable is the prototype to reproduce shapes. The reproduction of the braille letters was a mixed success. Most of the letters were recognisable but not without difficulties since the prototype at this stage is not accurate enough to set the dot's height in a consistent way. The raised dots should have had all the same height but they were inconsistent so the reading of the letters was troublous.

Another problem evidenced by the test was the orientation of the interface not being clear to the testers leading to misreading the dots positioning and so the letters. A proper braille cell's orientation should be intuitive to understand and this needs to be transposed in future iterations of the interface. Another important aspect of the interface was its rough texture. The natural roughness due to 3D printing was a big source of noise for the testers increasing drastically the difficulty of the reading. In this version of the prototype the dots are made of nylon wires. This decision, taken for sake of simplicity of realization, led to few reading problems. Nylon dots are too flexible and under the fingertip of the testers they bend adding challenge in the reading. They are also unpleasant to touch, a more noble material



(a) Braille text test

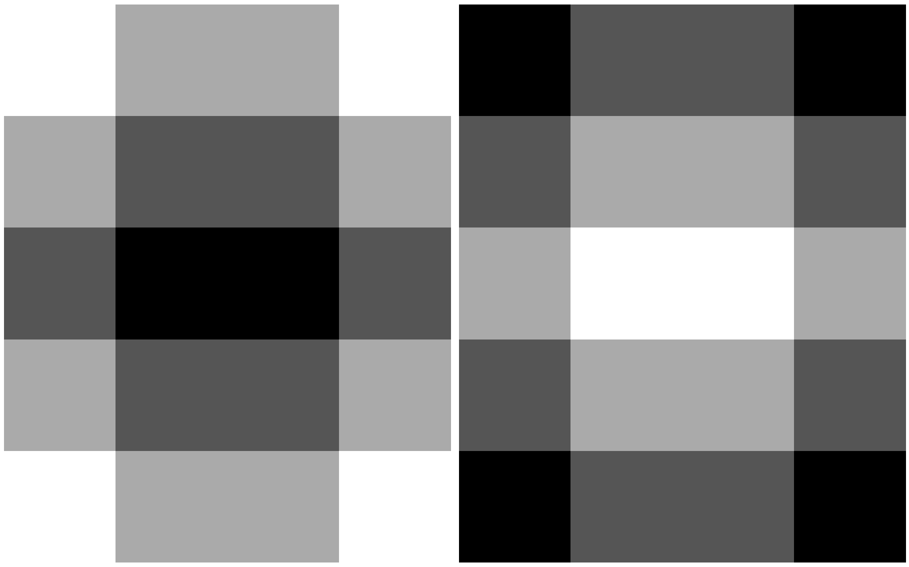


(b) Braille text test, decoded

Figure 5.1: Caption for this figure with two images

would probably be best at interfacing with the skin touch. Testers also reported that the current dot diameter is excessive, it is about 1mm, and so thinner dots would make the sensing easier. The second can also be identified as a not complete success. It consisted in the renderization of two images, one representing a dome shape that occupies the entirety of the matrix and the other the negative of the previous, a concave shape, but unfortunately the prototype ran into some error during the rendering of the second image so the submission to the testers was, in the end, impossible. However the render of the dome has been completed by the prototype and the testers recognised the shape, of course all the problems evidenced before were carried also in this instance.

In the end, in addition to the previous ones, there are other improvements to keep in consideration for the next iteration of the device. Render speed should be way higher to allow continuous reproduction of the virtual surface. A wider interface matrix would help the reading and orientation for the user. Adding a set of actuators to provide feedback to the user in edge cases would help during the sensing of the virtual surface.



(a) Dome shape test **(b)** Donut shape test

Figure 5.2: Caption for this figure with two images

Chapter 6

Future development

6.1 What further development could lead to

In this document it has been largely reported the development of a prototype with assistive technology finalities, even the conducted test involved the participation of visually impaired people. However this technology could be implemented in different applications since it essentially expands the range of sensorial feedback that can be reproduced to a human user.

Figure 6.1: Hololens 2 by Microsoft

As explained in the introduction this device shares with the Virtual Reality technology the same functional core concept so it would not be surprising to imagine it coupled with a VR or AR headset. Until now the reproduction of virtual environments has involved mainly the sense of sight: the VR/AR headsets put a double screen in front of the eyes of the user and substitute the seen reality with digital spaces achieving a level of immersion good enough to trick the human brain experiencing a whole new spectrum of emotions that were not present outside of reality. On the commercial side the gaming industry is leading the development of the VR technology but companies like Microsoft with the HoloLens [5] and Apple with the Vision Pro [6] have been investing in the development of AR sets destined to casual users and work environments. But, despite all the technology developed, these products still rely only on the sight as main stimulation, losing an unimaginably vast range of tactile interactions that could contribute to the reproduction of the virtual environment. This device could fill that empty gap. With a proper development the system could be implemented with those headsets: smaller dimensions could be achieved for portability and wearability, faster refresh rate could help immersion and representation fidelity, third dimension navigability could be implemented for 3D virtual spaces and a plethora of feedback could be integrated to unlock an exponential number virtual reproducible experiences.

The core concept of this technology is also stackable, putting together multiple mechanism a number of dots can be actuated, enough to cover an entire surface and build entire displays able to render 3D shapes. In the consumer pool an entirety of new experiences could born for the entertainment, the education, medical rehabilitation and working tools.

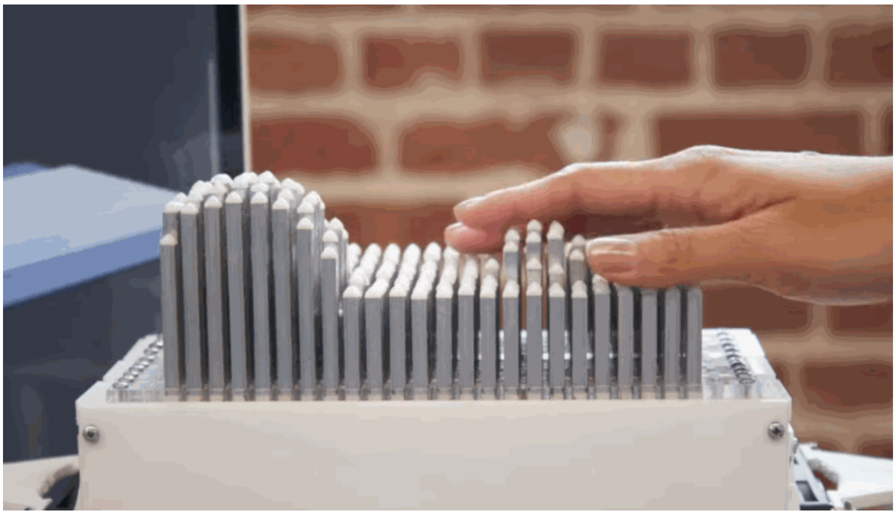


Figure 6.2: Tactile display by Stanford University

In a working context where the tactile sense would be useful but is obstructed by necessities, devices like this could lead to big improvements. For example space suits or underwater suits for obvious reasons need to shield the user from the external environment sacrificing big chunks of range of perception that could help during the normal operations. In a similar manner any operator that uses a machine as a tool or a means to interact with the world under certain circumstances or with specific empowerment would gain a sensibility in the interaction that until now has been forbidden. Terrain surface could gain interactive ways to perform expedition planning or scan analysis. Direct robot manipulation would be much similar to life experience. It has also been mentioned before how even the healthcare environment could benefit from iterations of this technology. Specialized personnel would have new tools to interact with patients in multiple scenarios and patients could have new tools for rehabilitation, testing, performance analysis.

To put it in a few words, the ability to interact with the virtual world with the tips of our fingers would improve our daily use of technology, having a profound impact on our lives. Work could be more efficient, assistive technology more effective and accessible itself, digital connections could now be physical.

Chapter 7

Conclusions

7.1 Insight from the development process

The work around the development of this functioning device was hard and long, the designs were changed multiple times, some choices were discarded in favor decision that would have been unexpected at the start of the project. Working around limitations and so restrictive pillars shaped quite lot the difficulty of the development and despite the result being not as effective as envisioned at first, the glimpse of potential appears to be bright. The learning process that this project required involved almost all the field that mechatronics could touch and this baggage of learning and experience will be valuable for any future development, even in different applications. In the end, with the first prototype fully developed and tested we can draw some conclusions. The prototype was a mixed success, it worked enough to prove the functioning principals that are key to its core concepts but technical limitations that afflict the device throughout the all device highlight the need to further development and better equipment. The testing was fruitful, the tester were able to recognise some of the reproduced sample and then return important feedback. Although the testers reported a number suggestions and critical aspects, some already known and others unexpected, the overall testing proved the validity of the concept, enough to justify further development in this direction. In the end this device was produced and then tested as assistive device. There is an immense amount of room for improvement with this application in mind and there is also a whole uncharted territory to discover when it comes to tactile feedback to be applied in VR, AR, work environment, space exploration, medical application or entertainment. As always, when technology aims to expand human capabilities, instead of substitute them, what is achieved is vast field of imagination ready flourish.

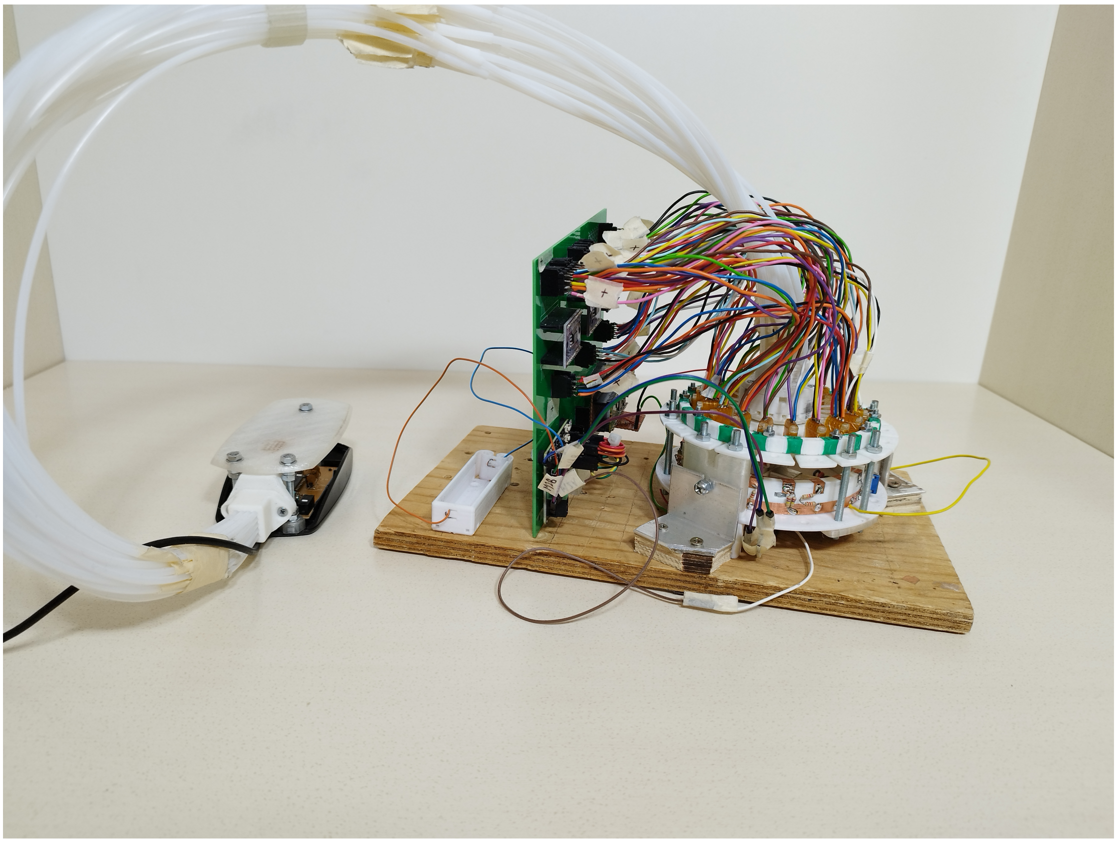


Figure 7.1: Full prototype, photo

Bibliography

[1] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. «inFORM: dynamic physical affordances and constraints through shape and object actuation.» In: *Uist*. Vol. 13. 10. Citeseer. 2013, pp. 2501–988 (cit. on p. 4).

[2] Mary W Moore and James C Bliss. «The Optacon reading system.» In: *Education of the Visually Handicapped* (1975) (cit. on p. 4).

[3] Raspberry Pi In. «Raspberry Pi Pico and Pico W - Raspberry Pi Documentation». In: (2024). URL: `https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html` (cit. on p. 13).

[4] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. «Repurposing diffusion-based image generators for monocular depth estimation». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 9492–9502 (cit. on p. 16).

[5] First Last. «Panoramica, funzionalità e specifiche di HoloLens 2 | Microsoft HoloLens». In: (2024). URL: `https://www.microsoft.com/it-it/hololens/hardware#documenta-le-esperienze` (cit. on p. 26).

[6] Applecontinuityicloud. «Apple Vision Pro - Apple». In: (2024). URL: `https://www.apple.com/apple-vision-pro/` (cit. on p. 26).