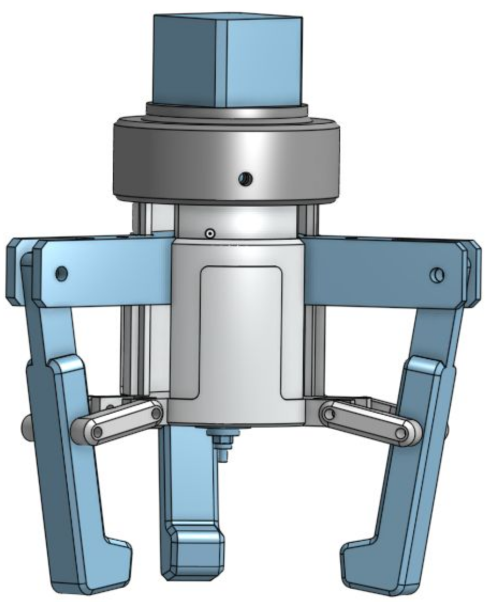
Aalto-yliopisto, Sähkötekniikan korkeakoulu
ELEC-D0301 Protopaja
2018

Final Report

Project #5
Futurice: Robotic touch



Jani Kiviaura
Joel Määttänen
Lauri Blomberg
Minh Nguyen
Ville Sinkkonen

Date: 31.08.2018

Information page

Students
Jani Kiviaura
Joel Määttänen
Lauri Blomberg
Minh Nguyen
Ville Sinkkonen

Project manager
Joel Määttänen

Sponsoring Company
Futurice

Starting date
4.6.2018

Submitted date
31.8.2018

Table of contents

Tiivistelmä

Projektin tavoitteena on suunnitella ja rakentaa robottikäsi, joka tunnistaa jos se on tarttunut kappaleeseen. Tämän vaatimuksen lisäksi robottikädelle tehdään tietokonesimulaatiomalli. Robottikäsien kehitys on tärkeää, koska tulevaisuudessa teknologia integroituu entistä tiiviimmin ihmisten kanssa. Esimerkiksi amputoitu raaja voidaan korvata robottiraajalla tai sosiaaliset robotit voivat työskennellä yhdessä ihmisen kanssa.

Projektitiimimme on rakentanut useita prototyyppejä robottikäsistä. Tiimi on perehtynyt robottikäsien tarttumiseen ja simulaatioon vaadittaviin materiaaleihin ja teknologiaan. Niiden kehitys tapahtuu projektin seuraavassa vaiheessa, koska kehitystyö käden mekaanisen suunnittelun parissa on saavuttanut tyydyttävän tason.

Projekti on edennyt siten, että prototyypin valmistumisen jälkeen tiimimme on analysoinut sen vahvuudet ja alkanut rakentaa seuraavaa, opitun tiedon perusteella. Raportissa vertaillaan prototyyppien ominaisuuksia, koittaen mukautua toimeksiantajan vaatimuksiin. Suunnitelmat robottikäden kehityksessä muuttuivat perustuen prototyyppien analysointiin. Kehityksessä jatkettiin analyysissä ilmenneitä ominaisuuksia, jotka täyttävät toimeksiantajan vaatimukset.

Toistaiseksi tiimimme on suunnitellut tai rakentanut 5 robottikättä. Olemme myös valmistaneet väliaikaisen version kontrollointiin tarvittavista piirilevyistä. Viimeisin malli robottikädestä on mäntä-malli, jonka toimintaperiaate tulee pysymään kehitystyön prioriteettina kurssin loppuun saakka.

Projektin aikana olemme käyneet läpi useita iteraatiokierroksia, ja olemme valmiita siirtymään robottikäden mekaanisesta suunnittelusta sähköisten toimintojen, sekä kontrolloinnin suunnitteluun.

Abstract

The aim of the project is to plan and build a robotic hand, that can recognize if it has grabbed an object. The requirements for the robot include developing a computer simulation of the robot. Robotic hands are important since technology will be integrated more tightly with humans in the future. For example, a amputated limb could be replaced with a robotic limb, or social robots could be working together in harmony with humans.

Our project team has built several prototypes of robotic hands. The team has conducted some research on the materials and technology required for grabbing and simulation. These will be under further development as we have reached a sufficient level for the mechanical design of the hand.

The project has followed a workflow, where we have built a prototype, analyzed it flaws and perks, and then started rebuilding another one based on what we have learned from the previous one. In this report we will compare the acquired results while trying to accommodate the requirements of the client. The project plan has changed in accordance with to the result we have obtained. All of the changes were made according to the information gained in the analysis of prototypes. In the development of the hand, we kept the features of the prototypes, which fulfilled the requirements of the client, and discarded the rest.

Until this point of time, we have designed and built five prototypes of robot hands. We have developed a crude version of the sense of touch for the robot, including PCBs and testing of sensors. The latest prototype for the robot hand is a piston-based model. The piston-based design will remain our main focus, until the end of the course.

During the project we have gone through many iterations of mechanical design for the hand, we are now ready to shift focus from the mechanics of the hand to sense of touch and a model for simulation.

1. Introduction

Research and projects on humanoids has increased extensively in recent years. A humanoid resembles the human appearance with various capability from simple walking to complicated interactions such as talking and understanding. Futurice has been interested on the topic for several years. Hence, Futurice has cooperated with Aalto University as a sponsor for this project as a part of the course Protocamp organized in summer 2018.

Futurice develops robots based on open source technologies for social environments. The tasks could be an educator, guide, artist, security guard, cleaner or nurse. This kind of fields will have an abundance of demand for jobs as population becomes older and generally, less willing to do manual labour. This kind of environment is also more challenging for a robot than regular assembly line work, because of the amount of information needed to take in from the environment and the required sensitivity in changing environment for safe operating.

Our project aims to create a robotic hand that can pick up a raw egg without breaking it. The hand needs to have enough gripping power to keep the egg stably and it is precisely control so that the pressure does not exceed the breaking threshold of the eggshell. The latter can be accomplished by sensors that measure the force applied by fingers and then the measured signal is returned as feedback to control the fingers so that they stop at the right moment. Our challenges are to design the hand, design the sensors to measure the gripping power of the hand, arrange the communication between the sensors and the motors controlling the hand, and create a digital twin of the hand if time allows. In our case, the digital twin is a computer simulation of the hand. Its purpose is to give accurate predictions of how the hand would perform in real life situations. Therein, we can run numerous simulations and allow the code to learn how to apply forces on the hand in different situations. The result of the project will be open source materials that help robots to function in this messy world around human life where we need to hold things in our hands firmly, but preferably not crush them.

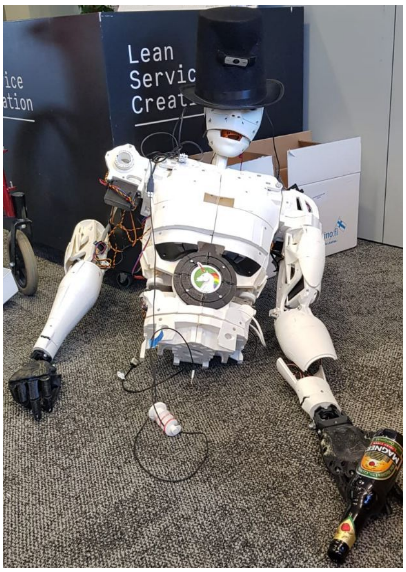


Figure 1. A robot made by Futurice.

2. Objective

The expected outcomes of this project are to create a software and hardware that equip a robotic limb with sufficiently precise haptic feedback loop so that it is capable of handling delicate objects without breaking them. The feedback loop is constructed from two major parts: (1) a hardware and software which adjust the force applied to objects, and (2) a virtual twin of the robotic hand which can be used to simulate and monitor the robotic limb movements.

Main use cases for this would be humanoid robots which interact with humans, such as simple greeters at a store. Considering how strong a force even simple servos can create, it is important for any robot that interacts with humans to have a sensing system installed in order to avoid painful accidents. Some more complex use cases are applicable from a future perspective. For example, cooking robots should be able to handle fragile materials such as egg or glass. Our goal is to create a robotic hand that can exert enough force to break an egg, and then implement a precise enough feedback system that it can gently grab and hold an egg without damaging it.

3. Results on Hand models

This chapter describes the hand models that the project group have produced and experiment. Each of the prototypes is presented with their strengths and weaknesses as well as the reasons why it has been continued developing or abandoned.

3.1 ADA hand

Figure 2. ADA Hand [1].

ADA Hand is a five-finger, humanoid hand, with the fingers being moved by strings (tendons) attached to linear servos (muscles) as shown in Figure 2. It is designed to be open source by Open bionics and can be easily 3D printed using flexible filaments. This hand was originally designed to mimic simple human hand gestures. In order to close the hand or control the fingers, the servos pull the strings forcing the fingers to bend. Once the strings are released the elasticity of the filament returns the fingers back to their original positions.

Our initial print of the ADA Hand was done using PLA (polylactic acid) filament found at Aalto workshop. While generally the hand looked and worked fine at the beginning, the filament was not adequately elastic, and stopped returning the fingers to their initial positions after some time. The material at the joints also started to show signs of wear and tear, indicating they would eventually break after a short period of use. Hence, we changed the use a more elastic material obtained from the sponsoring company. Additionally, we gained the experience to use soluble support material PVA (Polyvinyl Acetate) for printing for an ease when removing the it from the considerably tiny holes.

Figure 3. ADA hand assembly.

The second version shown in Figure 3 appeared to possess a better elasticity, yet another problem gained while controlling the fingers. The control given by the linear servos is inadequate to reliably grab an object, as the constraints of the servos prevent them to properly close the fingers tightly. Tightening the strings could relatively solve this issue. However, we have postponed the work on this hand because of some reasons. Firstly, we already had another promising prototype. Secondly, we only got two of the required five linear actuators for the model whereas our budget is limited meaning that we should consider a more feasible option. Thirdly, the design of the hand does not allow sufficient space for allocating and wiring the sensors. We decided to leave the model to the end of the project and work on it if there was still time remaining or if the other prototypes prove to be unsuccessful. Furthermore, the linear servos could only pull the finger to a 90 degree angle in relation to the palm which made improbable grabbing without some major alterations to the motor and strings pulling the fingers.

3.2 Silicone hand

One option we looked into in the beginning of the project was using soft (flexible) parts for the joints of the hand. Such flexible parts would provide softer touch, making breaking the objects less likely. One such design is a three-finger, single motor, string actuated model shown in the Figure 4. Similarly to the ADA Hand, the servo or motor pulls the strings, which forces the fingers to bend. After the strings release, the flexible parts reverse and open the hand.

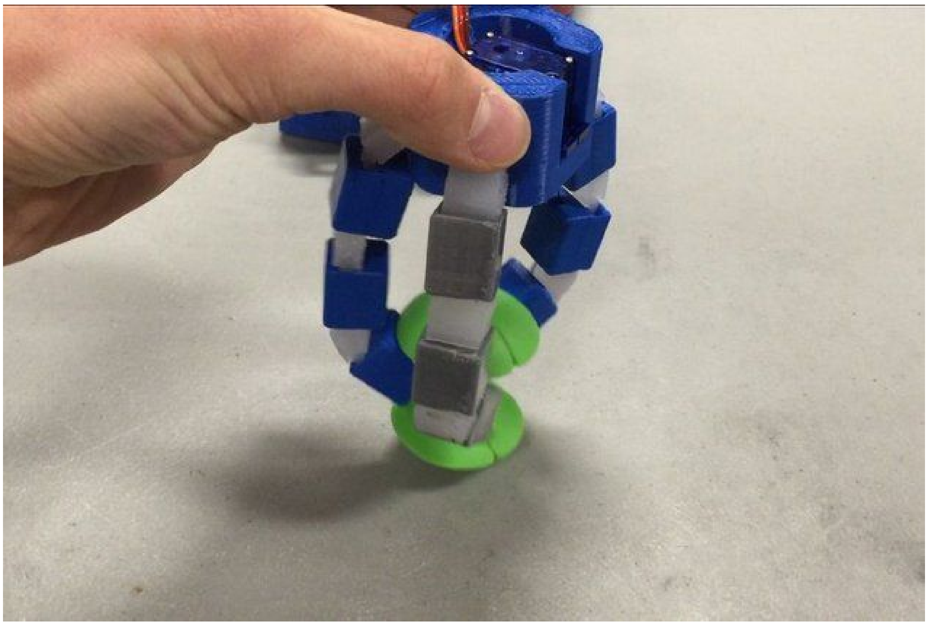


Figure 4. Silicone hand [2].

After some evaluation, we realized a problem with this design as it requires some specific types of silicone, that was both expensive and difficult to get in reasonable time in Finland. Besides, modeling such flexible joints for the digital twin would most likely prove to be challenging as well. We attempted the shown design with construction silicone instead of the intended one, as that was significantly cheaper, and available instantly from the local stores. We printed a mold from water soluble plastic - PVA, filled it with the silicone and allowed it to cure over a week (Figure 7). The resulting pieces were of the right shape, but the silicone could not match the required durability as it broke during the assembly of the hand.

Figure 5. Making of the silicone parts.

Figure 6. Bagged 3D-printed parts of the silicone hand.

3.3 3-servo hand

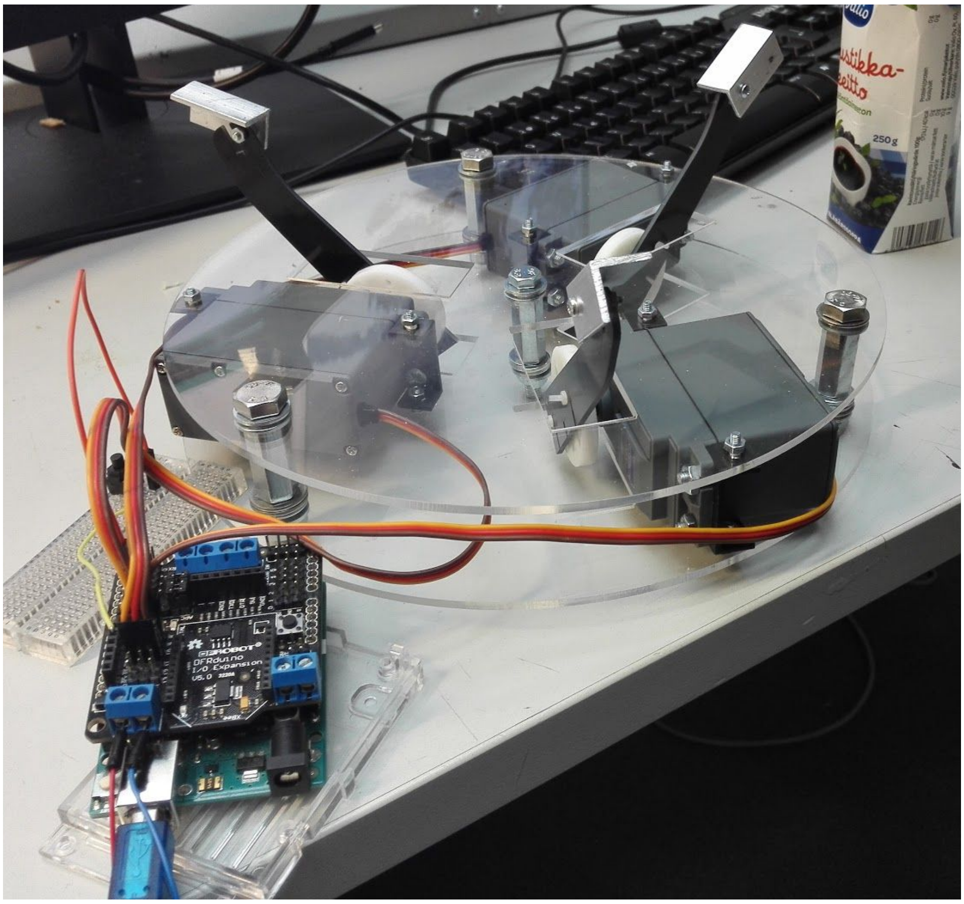


Figure 7. 3-servo hand

Another design we have put attempt on was a three-finger gripper with each finger powered by their own servo. The design includes basically two plates with all the components in between as a “sandwich” structure illustrated in Figure 7. It starts closing when a button is pressed, and uses a pressure sensor on each finger as a feedback to stop. It stops squeezing when a pressure threshold, specified in the code, is reached, and holds the grip until the button is pressed again to open the claw.

This design used HK15338 servos from the inventory of the workshop. Those servos are not sufficiently reliable, as they suffer from some level of stutter and inconsistent movements, most likely due to their age and prior usage. Besides, the size of the servos resulted in a substantially occupied space which was not appreciated by the sponsoring company. Several designs were constructed based on the same sandwich idea in an attempt to reduce the size of the structure but the dimensional concern still could not be solved completely. For example, some smaller servos were applied but provoked a problem of less gripping power. Then, a more promising prototype was achieved as a piston design.

3.4 Piston hand

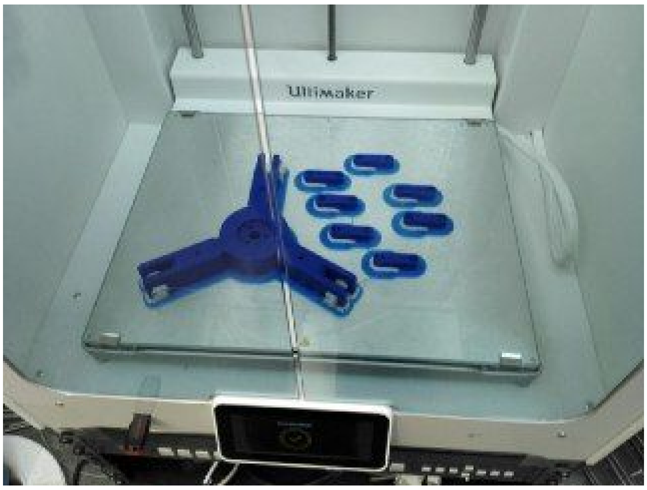


Figure 8. Printing piston

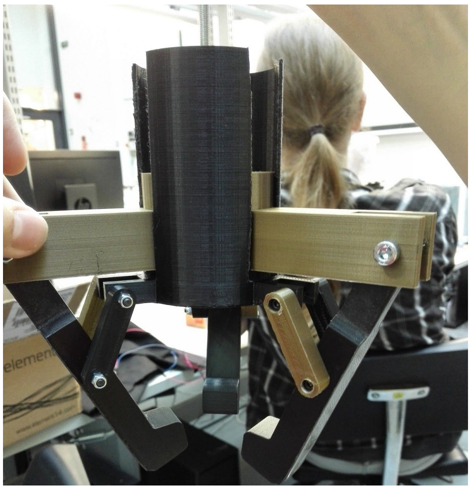


Figure 9. Piston hand version 1

Our final design is a piston hand which is an inheritance model of the prior three-finger sandwich. This model is more compact in size when compared to its predecessor, which was not appreciated by the sponsoring company. This prototype uses a stepper motor to spin a threaded rod that moves the piston (the big part in Figure 8) to which one end of the fingers are attached (golden ledges in the Figure 9). This motion moves all three fingers simultaneously. This model also advances as it can provide higher gripping force due to the friction between the thread and the nut. According to our observation, the opening and closing of the hand is at this stage quite slow. However, we have proposed a feasible improvement with a better rod and motor. The model was designed in OnShape CAD, which can export all parts as STL format for the digital model's meshes as well as 3D printing. It can also export as COLLADA, which would contain all information about the joints as well.

After the first prototype of this model, it was the most promising we have achieved. Then, we attempted to enhance the model by adding sensors to the top and bottom of the piston to ensure it will stop when it reaches the end position in order to prevent it from breaking itself. Additionally, the model was modified to be more versatile and yet stable shaft, and some minor details was reshaped continuously to improve the overall aesthetics.

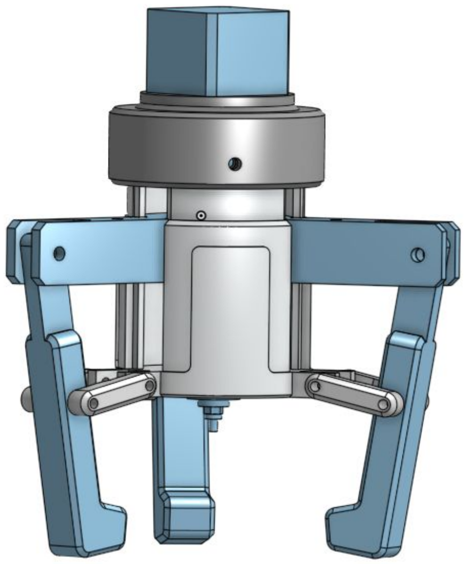


Figure 10. Piston design(25.7.2018)

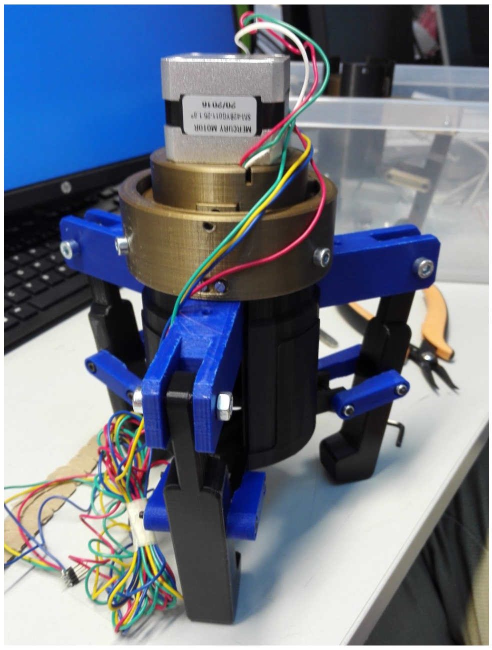


Figure 11. A reprint of the piston hand with improved design.

While designing the individual pieces, we did acknowledge about the assembly and the order how we could attach the parts. As some parts could not be attached if some other parts were in place, the structure had a definitive order for parts. The threaded rod had a bad connection to the motor, and got disconnected many times. Then every time we needed to take off many nuts and bolts to fix the connection. Also the connection of the wrist part that holds the electronics to the rest of the hand is improvised, because in the beginning we did not design the hand to be connected to the robots arm.

The problem with connection between motor and rod was fixed by several methods. First, a hole we drilled holes on the rod to support the connecting screws. Many other design flaws were needed to be handled by drill or dremel, and in the final version of onshape we fixed all the problems we found during the course. Additionally, we also produced the connection parts in Figure 12 from aluminium foil with drilling and turning. An experiment to use a printed part was failed because of the expansion and reduction of the printing material. The machined part is more stable and could fix the parts together in proper quality.

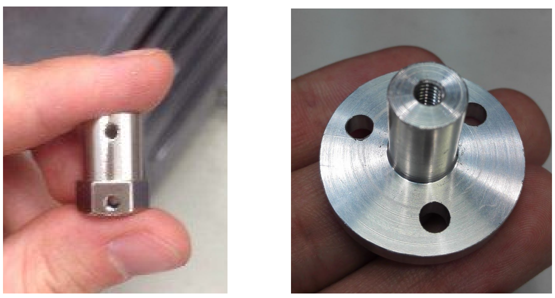


Figure 12. Connections pieces

4. Results on Feedback

4.1. Current sensor

Since the fingers of the hand are moved by a motor, a possibility for feedback measurement is the current flowing into the motor. When the motor runs without load, the current is normally small. When the motor is resisted, normally caused by a load or a force, the current increases. In our project, theoretically the increasing of the current indicates that the fingers is probably touching an object.

Figure 13. MAX4372TEUK.

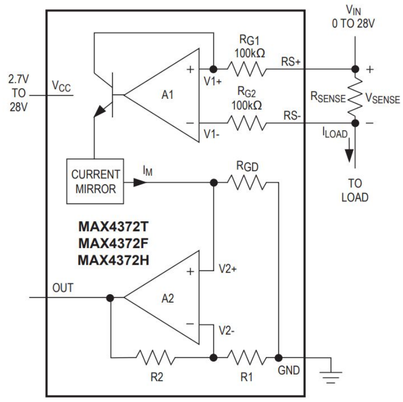


Figure 14. Current sense amplifier functional diagram.

For this feedback, we have planned to use a current sense amplifier (MAX4372TEUK, Figure 13). A current sense amplifier is typically used for measuring the working characteristics of devices in a circuit or protecting the overload in a circuit [3, 4]. The circuit can be set up in accordance with the diagram in Figure 14. An appropriate shunt resistor will be selected so that the output voltage at the OUT pin satisfies the measurement requirement. For instance, it does not exceed 5 V if it will be read by Arduino. The maximum allowable voltage at OUT is 15 V. However, this sensor was not implemented in this project since other sensors were sufficient for the aims of the project.

4.2. Pressure sensor

Pressure sensors have been researched and used in various applications for robotic hands [5]. Pressure sensors are utilized in this project as a target of the project is to measure the pressure between the robot fingers and the grasped objects. In the project, three type of sensors are being tests for an appropriate selection including two thin film pressure sensors (force sensing resistors) and a strain gauge sensor.

4.2.1 Thin film pressure sensor CP1-149NS

CP1-149NS (Figure 15) is a piezoresistive tactile sensor whose geometric stretching and compression results in detectable change in its resistance [6]. Figure 16 shows an equivalent circuit of the sensor. From Figure 17, it can be seen that the sensor response almost linearly when the force is between 0.5 and 100 N meaning that the pressure can be easily achieved for this range. However, the sensor has a better resolution in the range between 0.3 and 0.5 N which is not likely relevant to the purpose of this project but can be useful in touch detection.

Figure 15. CP1-149 NS

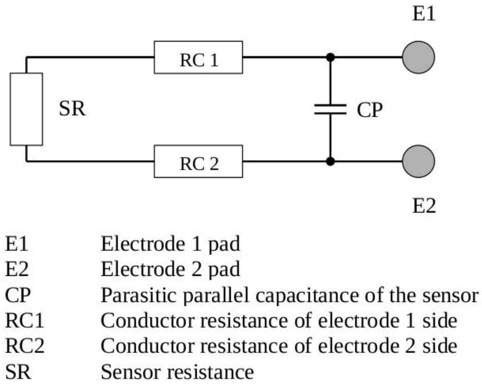


Figure 16. Simplified sensor equivalent circuit

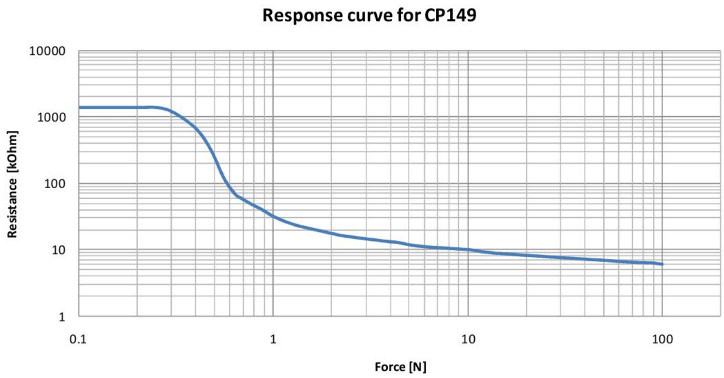
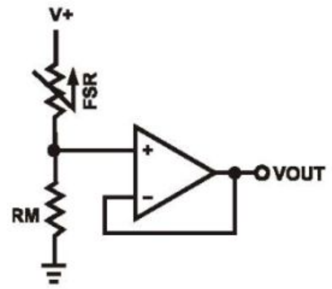


Figure 17. Response curve of CP1-149 NS sensor

4.2.2 Force sensing resistor FSR 400 (FSR)

Figure 17. FSR 402

FSR 402 is a force sensing resistor belonging to the family FSR 400 produced by Interlink electronics. This type of sensor is designed based on piezoresistive sensing technology [5]. It is another thin film sensor of which resistance decreases with the increasing of force. It can detect contact and touch between objects and measure a proportional change in force [7]. A simple force-to-voltage conversion circuit is usually used as in the below figures:



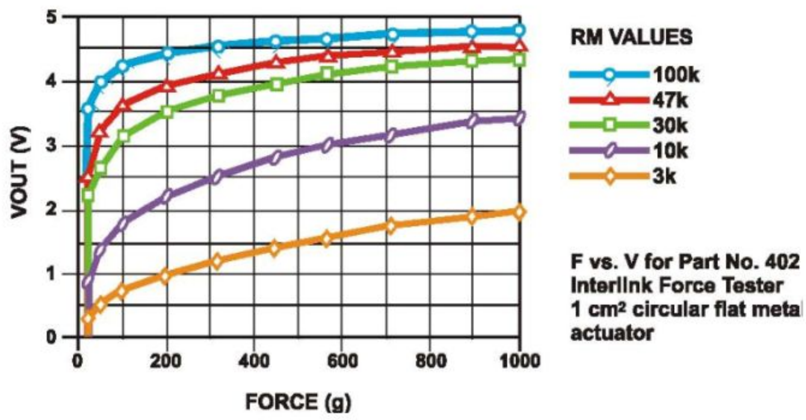


Figure 18. Force-to-voltage circuit Figure 19. Possible values of RM

The working principle of the circuit can be illustrated in the following formula:

$$V_{OUT} = \frac{R_M V+}{(R_M + R_{FSR})}$$

where Vout increases with increasing force,

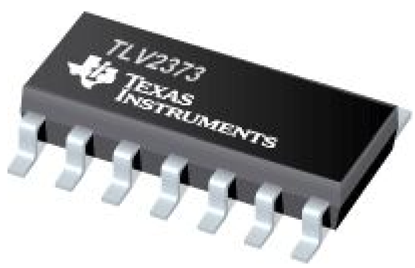
RM is a resistor of which the resistance influences the resolution of the measures output,

RFSR is the resistance of the sensor under application of a force,

V is the power supply voltage.

According to Figure 19, the value of the resistor RM should be 10 kΩ or 30 kΩ for good resolution of the output voltage.

The result from the FSR is measured and amplified through an amplifier. TLV2372 amplifier (Figure 20) was used in this project. Its circuitry can be seen from Figure 21. The outputs of the amplifier are connected to the input of Arduino board for voltage measurement as the feedback of the control loop.



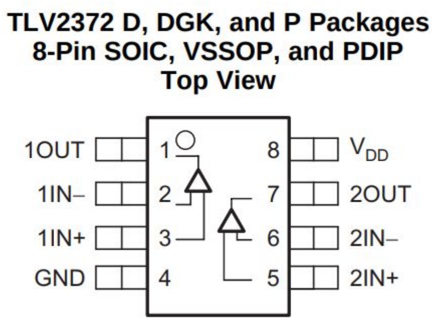


Figure 20. Amplifier Figure 21. Amplifier circuitry

4.2.3 Strain gauge sensor TF3-120K

The strain gauge functions based on the change of electric resistance of metallic foil when it elongates or contracts. The foil has a rate of resistance change proportional to strain with a certain constant [8].

Figure 22. Strain gauge sensor TF3-120k

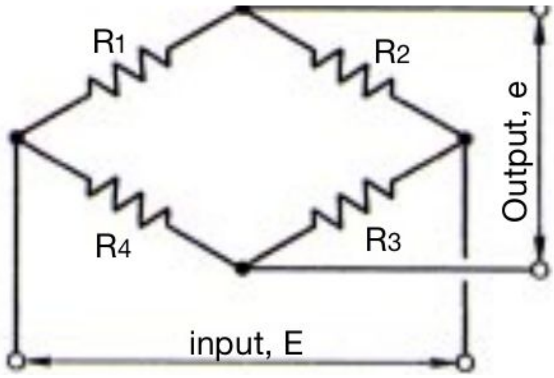


Figure 23. Wheatstone bridge.

The sensor is used in a Wheatstone bridge. Typically, the bridge is constructed from four equal resistors so that the output voltage, e, is zero in balance condition. When the resistance of the resistor changes, the output voltage also changes proportionally as

$$e = \frac{1}{4}\frac{\Delta R}{R}E$$

where e is the output voltage,

ΔR is the change of resistance,

R is the resistance of strain gauge,

E is the input voltage.

A gauge sensor can be used to replace one of the resistor as in the following figure.

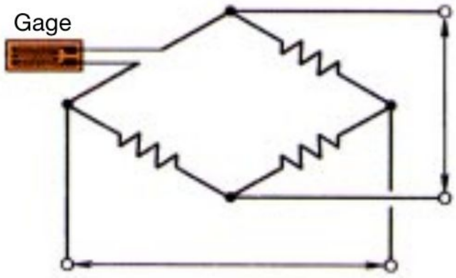


Figure 24. Circuitry with one gauge sensor.

Then, the voltage is equivalent to:

$$e = \frac{1}{4}K\varepsilon E$$

where ε is the strain,

K is gauge factor.

Then, the strain ε can be determined.

4.2.4 Pressure sensors comparison

The following table presents a brief comparison between the sensors [9]:

Table 1. Sensor comparison.

Features	CP1-149NS [x]	FSR 402 [x]	TF3-120K
Type	Thin film force sensing resistor	Thin film force sensing resistor	Strain gauge
Dimensions	6 mm diameter	18.28 mm diameter	5 x 7.5 x 0.06mm
Measurement range	0.5 - 100 N/cm2	0.1 - 10 N	
Rise time	2 - 3 ms	< 3 μs	
Resistance	1 M **Ω** > R > 2 k**Ω**		120 **Ω**
Requirement	Mounted in a smooth, even and hard support surface	Mounted in a smooth, even and hard support surface	
Merits	● The size of the sensor is more suitable for fingers which have similar sizes to human fingers ● Low cost ● The sensitivity of the sensor is almost linear between 1 N and 100 N	● Low cost	
Demerits	● Humidity sensitive	● The size of the sensor is quite larger than the size of the fingers ● Nonlinear response ● Temperature sensitive	● Susceptible to humidity and temperature ● Overload must be prevented

As the main aim to obtain the feedback from the hand, the sensors is mounted on the tip of the fingers. The signal from the sensor was transmitted to the Arduino board for controlling. The grasping hand stops when a threshold limit is reached.

4.3 Infrared proximity sensor

The robot hand has been improved with a infrared proximity sensor GP2Y0D805Z0F (Figure 25). It functions based on infrared light emission and reflection. Its working range spans from 0.5 to 5 cm. The sensor can quickly detect the presence of most opaque objects within the detection range. The sensor can be simply connected to the circuit using three pins Vin, GND and Signal.

The Signal pin is connected to Arduino board for controlling the hand. The infrared sensor is mounted on one of the hand so that it can detect an object but not the other fingers. First we attached the sensor under the tip of one finger. But when the finger got very close to the object, the sensor was closer than 0.5 cm from it so the sensor didn’t notice it anymore and the fingers started to open. Then when they got few millimetres farther, sensor noticed the object again and started closing. After we moved the sensor next to the finger farther from the object, this problem didn’t occur anymore.

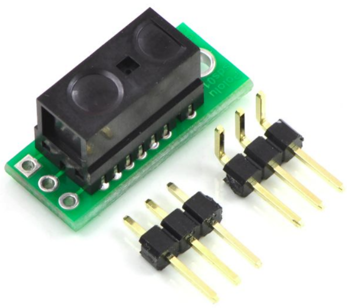


Figure 25. Infrared proximity sensor.

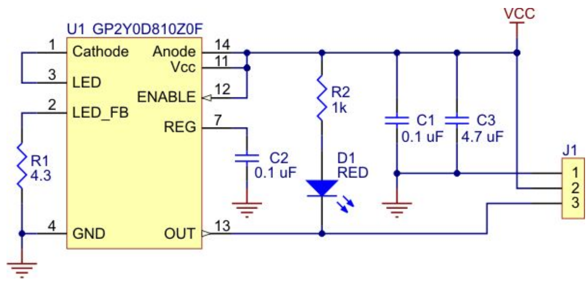


Figure 26. Infrared proximity sensor circuitry

5. Other elements

5.1. Motors

Finding three or more identical, working, and non-modified servos in the workshop has proven more difficult than one would have expected in the beginning of the course. We have tried using servo motors and stepper motors, the current design of robot hand uses one stepper motor. For the current hand prototype, it could be possible to use any kind of these three types of motors: servo, stepper or DC. AC motors and other bigger/more powerful possibilities were discarded as they would not meet the size or weight requirements of the project.

We are yet to test using a simple brushless DC motor for the project. This has not been done as a stepper motor provides more accuracy at the loss of torque. An idea of using a DC motor with an encoder has been presented, and could be worth a while to be looked into.

First prototypes used simple servos because they provided fair amount of torque and accuracy. As our project advanced we found out that the step-control-system would be more reliable and sophisticated than using a servo motor. Stepper motors are also easier to mount to a lead screw shaft as the 3D-Printing and CNC-communities have popularized the design of these types of systems.

Finally, we decided to use a stepper motor which provides flexibility in controlling as well as sufficient power for holding objects. For two prototype, two bipolar stepper servos were used including SM-42BTG011-25 which can provide a torque of 0.23 Nm and NEMA17-13-04SD-AMT112S which can provide a torque of 0.30 Nm. It is controlled by a motor driver DRV8255 from Pololu.

5.2. Digital twin

A digital twin is one of the main requirement of the project as it benefits the simulation and testing the robothand. It was suggested to be implemented using Robot Operating System (ROS). As such, creation of an URDF (Universal Robot Description File) would be ideal.

The piston hand, that is the final physical prototype, was created using OnShape CAD, which can export the models in COLLADA, which can be used in ROS robots. Unfortunately, OnShape exports these files in COLLADA 1.4.1, whereas ROS only supports COLLADA version 1.5. While some converters and conversion methods between the versions exists, none has been proven to be functional enough to solve the issue. Tried converters include go-engine collada converter, various online converters, URDF Editor GUI package for ROS, and a blender plugin, that allows exporting to COLLADA 1.5. Older versions of ROS were also attempted in the hopes that the older collada-to-urdf packages would support COLLADA 1.4. These older ROS versions have however reached the end of their life, and the package repositories have been disabled, making it impractically difficult to install older software packages. Those converters that can convert the

COLLADA files up to the point that ROS doesn't give errors due to incorrect version, are often incomplete, resulting in XML parsing errors instead.

As the automatic creation of the URDF files constantly failed, an attempt was made to create the URDF by hand manually. This process involved taking STL files from the OnShape model and writing each joint, their parent and child link, their positions and their angles manually. During this process a realization was made: our hand was impossible to model as an URDF, since URDF can not model kinetic loops, that are present in our design. This also explains the failures in the automatic creation attempts.

In the end, a proper digital twin was not made due to the aforementioned difficulties and approaching course end. However, the OnShape COLLADA files provide a good starting point, if anyone wants to give the digital simulation model another attempt in the future.

5.3. Wifi communication

At the end of the project, we would like to improve the result using Wifi communication to control the hand as well as sending the acquired data to a web server for better illustration. It may also be used for IoT in the future applications. The project group used the available ESP8266 from Electrical workshop:

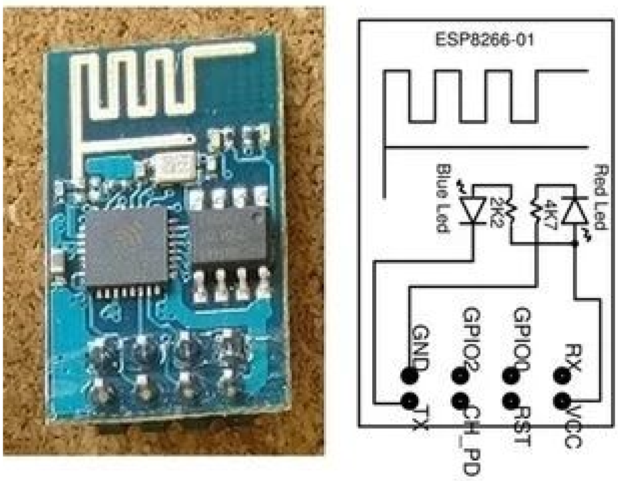


Figure 27. ESP8266 wifi module

In order to use ESP8266-01 wifi module, some steps need to be processed as follows:
i. In the Arduino IDE click File -> Preferences -> add the line
http://arduino.esp8266.com/stable/package_esp8266com_index.json as in Figure X.

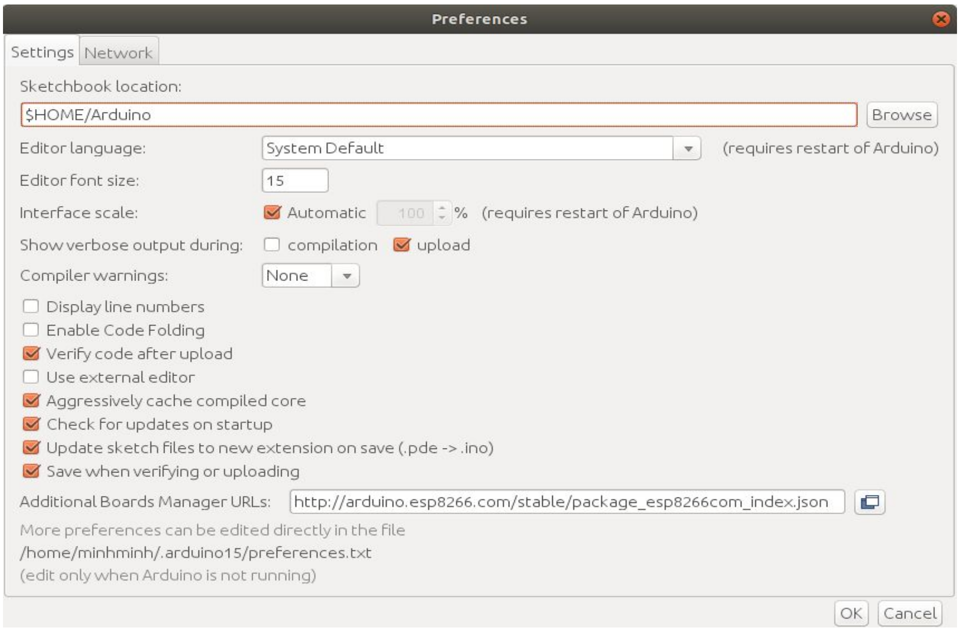


Figure 28. Including board manager

ii. Sketch -> Include library -> search for ESP8266 library and install it.
After that, the Generic ESP8266 Module board will be included in the list of boards as well as the example sketches will appear when the board is selected.

Since the available boards only include two GPIO pins, it would be difficult to send data from multiple types of sensors, it should be used in cooperation with an Arduino board which possesses more input/output pins. They are connected through wires and the Arduino board will send command to the wifi module using AT commands. Those commands are available after the AT firmware is flashed to the module. Several flashing tools are obtainable from open sources as follows:

- ESPflash tool for Windows user with an easy-to-use interface.
- Arduino tools (located at $HOME/.arduino15/packages/esp8266/tools/esptool/0.4.13/esptool)
- Esptool: https://github.com/espressif/esptool

In this project, we used the two available resources including Arduino IDE esptool and esptool from espressif. The latter tool can show more information relating to the chip such as chip type, MAC address. Additionally, it could flash the firmware (less than 30 seconds) faster than the esptool from Arduino IDE (proximately one minute) as observed from the flashing process.

Two versions of AT firmwares were flashed as follows:
i. v0.9.5ATfirmware

- Arduino tools in terminal:

```
>> $HOME/.arduino15/packages/esp8266/tools/esptool/0.4.13/esptool -vv
-cd ck -cb 115200 -cp /dev/ttyUSB0 -ca 0x00000 -cf
/path/to/the/AT/firmware/v0.9.5.2ATFirmware.bin
```

- Using esptool from espressif:

```
>> esptool.py --port /dev/ttyUSB0 --baud 115200 write_flash 0x00000
/path/to/the/AT/firmware/v0.9.5.2ATFirmware.bin
```

ii. Newest version of ESP8266 AT firmware

- Arduino tools in terminal (for both ESP modules):

```
>> $HOME/.arduino15/packages/esp8266/tools/esptool/0.4.13/esptool -vv
-cd ck -cb 115200 -cp /dev/ttyUSB0 -ca 0x00000 -cf
/path/to/the/AT/firmware/boot_v1.2.bin -ca 0x01000 -cf
/path/to/the/AT/firmware/user1.1024.new.2.bin -ca 0x7c000 -cf
/path/to/the/AT/firmware/esp_init_data_default_v05.bin -ca 0x7e000 -cf
/path/to/the/AT/firmware/blank.bin
```

- Using esptool from espressif (only worked for the 512 kb module):

```
>> esptool.py --port /dev/ttyUSB0 --baud 115200 write_flash 0x00000
/path/to/the/AT/firmware/boot_v1.2.bin 0x01000
/path/to/the/AT/firmware/user1.1024.new.2.bin 0x7c000
/path/to/the/AT/firmware/esp_init_data_default_v05.bin 0x7e000
/path/to/the/AT/firmware/blank.bin
```

  The memory allocation was achieved from the README.md file going along with the binary files.

Two ESP8266 wifi modules were used to test the connections including (1) ESP8266 302014 with 512 kb flash size and (2) ESP8266 302015 AI cloud Inside with 1Mb flash size. The flash size was determined with the esptool developed by espressif. The two modules possess some different in their properties. Module (1) can be flashed with both tools and both firmware. Module (2) has taken some uploading efforts but could only succeed when flashing the newer firmware version with esptool from espressif. This is not of our interest, so we did not progress it further.

In order to run AT commands, the flash pins must be disconnected. In order to join a network, the work mode of the ESP needs to be set to STA or STA+AP (AT+CWMODE=1 or AT+CWMODE=3). Other AT commands can be found from internet and official document of the manufacturer. They must be used without space otherwise the terminal will raise “ERROR”.

This functionality allows the wifi module transferring the information to a webpage. In this project, we tested it with Thingspeak and sent information from an Arduino board to the wifi module through Serial communication. However, we did not have enough time to test the printed circuit board for the ESP module.

6. Printed circuit board

For an organized demonstration, the electronics components were integrated in a printed circuit board as Figure 29. The schema of the board was designed using KiCad. Then it was manufactured using necessary chemicals in the workshop. Finally, the components were soldered on the board as Figure 30. Several versions of the board were produced since wrong connections were found during the testing phase.

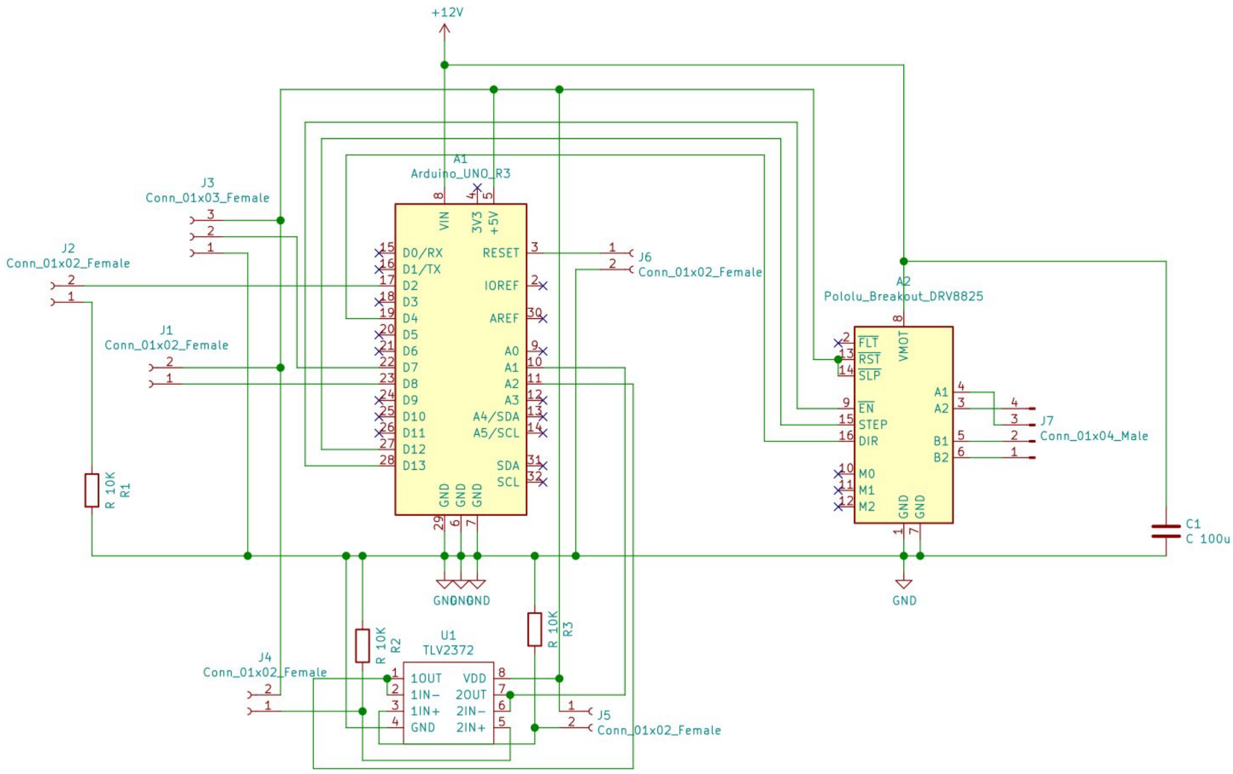


Figure 29. Printed circuit board schema.

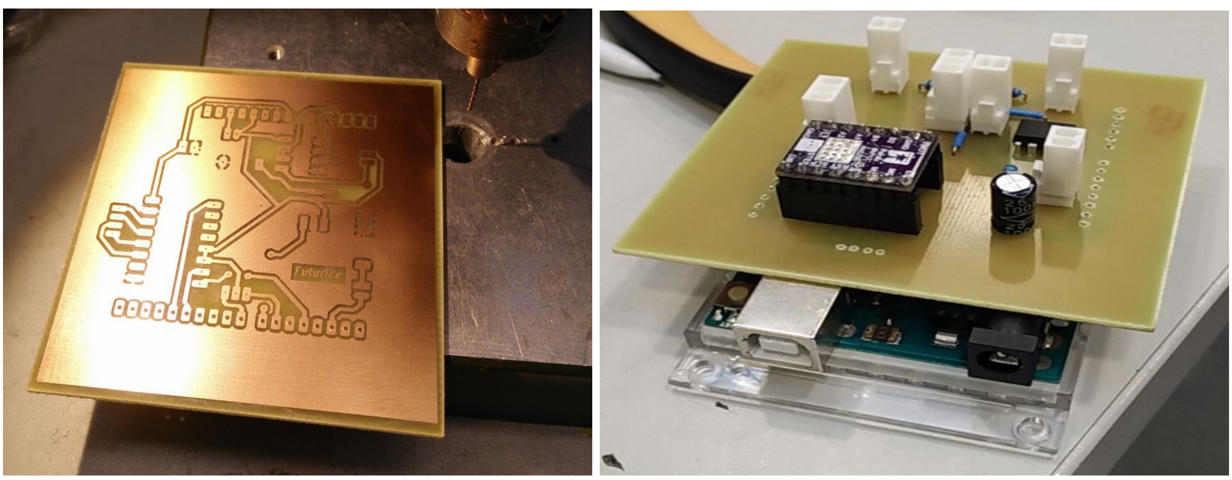


Figure 30. Printed circuit board manufacturing.

7. Software

The program code for the robot hand was created using the arduino IDE and libraries. The motor is controlled with a PWM signal.

The robot uses an infrared proximity sensor to detect if there is an object between the fingers, and automatically grabs it. The robot uses the pressure sensors to detect when the grabber has achieved sufficient gripping strength.

As the device only had one programmable button, originally intended for releasing a grabbed object. Some tricks had to be implemented to make it possible to control all the last minute features that were added to the robot. These features included IR sensor override and crushing of an object.

The IR sensor override allows the user to force the gripper close even when an object is not detected by holding the button after the piston had begun moving. The piston keeps moving regardless of the IR sensor value for as long as the button is being hold and no pressure sensor threshold or maximum piston position has been reached.

The crush mode can be activated after an object has been grabbed by holding the button for a second instead of quickly pushing it, which would release the object. For safety reason, the button needs to be released before the piston actually begins crushing, since the same button is then used for stopping the crushing if necessary. Crushing mode ignores all sensors, but not the maximum piston position to avoid the robot from destroying itself.

8. Reflection of the project

8.1 Reaching objective

We started with a detailed plan for 3 different prototypes from which we would choose a functioning one. The idea was to work in union on the three prototypes so that we could reach the results quickly. After choosing a prototype we would have started developing it further, so that in the end of the course we would have reached a viable prototype.

We started doing research on the prototypes, and figured out some key features we would like to have. At the halfway of the course we realized we could not use any of the prototypes, and started to look for more and different ideas for a prototype. We found a youtube video describing a crude mechanical design of a piston-based hand, and pivoted fast to make a similar design. For the visuals we took inspiration from a video game called half-life 2[31], where there is a gun called the gravity gun. Therefore, we decided that our new prototype would be called the “grabbity gun” as it went well with the PR material. At some point we found out our plan for the digital twin were not possible, and decided to stop working on it, and instead perfect the design of the mechanical hand.

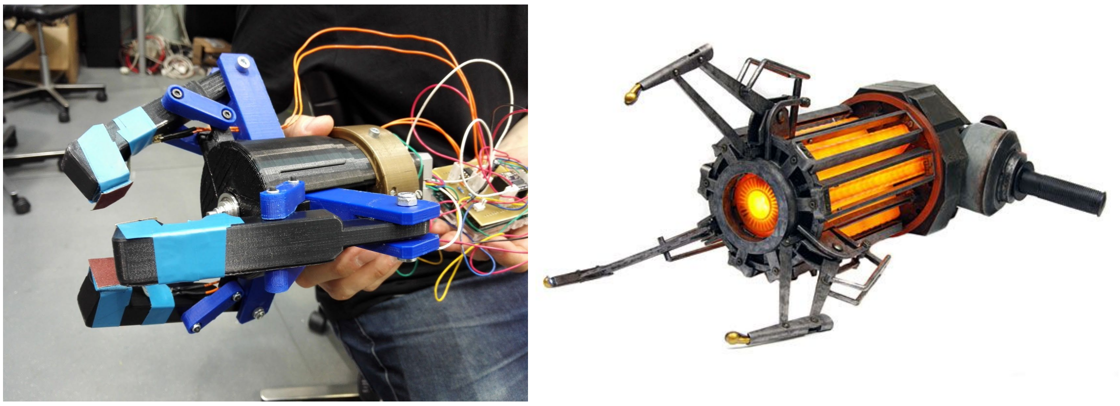


Figure 31. Comparison of the prototype and the gravity gun from Half-life 2.

With help from the assistants of the course, we were able to complete a working prototype within the deadline. The prototype was a hit on the demo day and at the company. Excluding the digital twin we managed to deliver on all the requirements made in the beginning of the course.

8.2 Timetable

Original timetable for the project had three parts:

1. Building multiple quick prototype designs in parallel. (2 - 4 weeks)
2. Testing said prototypes, figure out which one is the most promising. (1 - 2 weeks)
3. Further develop the most promising prototype, start working on the digital twin. (rest of the course)

This plan fell apart at part 2, since none of the prototypes we had created really worked well for our purposes. The ADA hand proved to not have enough gripping strength to be of use, and it would be a nightmare to make a digital twin because of the multiple joints driven by a string. Silicone hand required specific materials that we were hesitant to order because of the price, considering how useless the final prototype looked based on our research. In the end we started making more quick prototypes, first of them being the 3-servo hand. This design proved to be the easiest to make a digital twin for, and was easy to combine with the idea to use a threaded rod driven by a stepper motor to move parts. We decided to go forward without a real timetable for the rest of the project, since we had to do a lot of quick prototyping to solve problems as we found them. In the end we decided to ditch the digital twin to have more time to polish the final product.

8.3 Risk analysis

Unplanned risks:

- Our initial design did not support adding features that are required but we didn't pay attention to in the initial design
- Problems with the code and the applicability for different kinds of motors?
- We ran out of time with the digital twin and the final prototype did not work with the chosen software

Realized risks:

- We do not have good notes for the final report - Working on it
- Our three original prototypes fail to fill the requirements - We found another idea from youtube
- We break too many parts and do not have time to print or buy new ones - Many motors of paja were broken so we used only 1 of them in the end. So we bought a 125 $ motor online.
- Delivery time for the motor for the final prototype, ordered it possibly late so we didn't have time to implement the encoder in the control code.

The risk analysis we did on the original project was rather complete, since there were only a couple of unplanned risks. The one with the largest impact was the fact that the digital twin ended up being really difficult to implement, especially with our final design. The software we had chosen for the twin simply doesn't support some of the features required for our prototype, and we didn't have time to start from scratch since our prototype still required a lot of work and the course was in its final weeks.
Another risk we did not planned for was that we forgot to include some features in our final prototype designs. For example, the motor enclosure and parts connected to it are hard to assemble, thanks to needing to take certain steps at the same time.

Most important risk that we planned for and ended happening was that our initial research prototypes would not fit our requirements. Further prototyping and research solved that one rather fast. Breaking too many parts and having to buy replacements was a problem we had planned for, but had to implement for different reason: A lot of the servos and stepper motors at the paja are either broken or modified in a way that doesn't fit our use case. In the end, we decided to order a high-quality stepper motor for our final high-power prototype that had some nice extra features, but we ordered it a bit too late to implement those features in the final code, another risk we had planned for.

8.4 Other remarks

Having a detailed plan in the beginning hindered our progress. After the plan had failed, we were able to actually make faster, and better, progress. This could be because we were not sure about the prototype we were going to make. Therefore, now we think we should have chosen the project plan model, which allowed a more unclear vision of the final prototype in the planning phase. Unclarity also resulted from the companies vague instructions and failure to deliver essential items, like the robot hand we were going to implement the feedback on.

We started the project fast, by just making any prototypes we wanted to try out. This could have been avoided by more efficient planning. We had a plan, which was very detailed, but did not cover the right information. We should have planned more about the prototypes, trying to foresee problems before they occured.

Solving problems was crucial to have a good learning experience. We are sure we all have learned skills during the course that will benefit our ability to accomplish new projects. Having the knowledge we have now, would have been a great benefit at the beginning of the course.

9. Discussion and conclusions

We worked hard on the project, and finally succeeded to make a viable prototype. During the course all of the team members worked on the project the required amount of hours to deserve 10 credits. Especially the last 5 weeks were intense, but through hard work and learning we made it work.

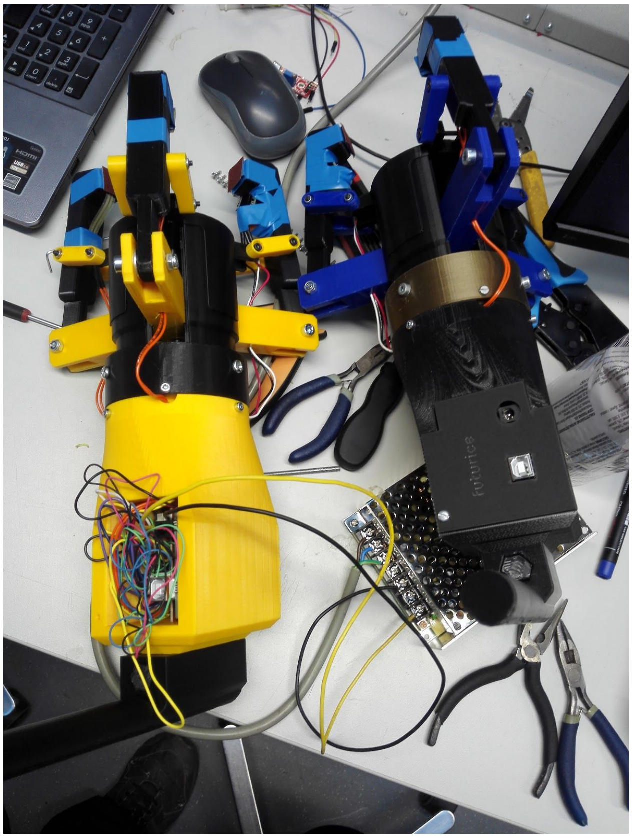


Figure 32. Picture of complete robot hands.

Greatest experience we all had, was learning how to manufacture a mechatronics contraption from scratch. During the course we learned to use every tool the workshop offers. Especially learning machining and turning was a experience we could not have had without this course. We also learned to design and manufacture PCBs and to make good mechanical decisions. During troubleshooting we had to resolve problems using drills, oscilloscopes and signal generators. We used the grinder to grind metal, soldering irons to solder components to boards and threading tools to make threads. We also acquired a great understanding of the limitations and possibilities of 3D-printing and laser cutter.

Although the plan we had in the beginning did not work out, we were able to recover and thrive with a new plan. One of the problems the first plan had was that it was too detailed and restricted. After we stopped micromanaging the details of the project, everything started to work. This was a great lesson in project management overall.

Figure 33. Having fun breaking and decorating eggs.

In the end, what made the whole project work were the great team members we had, and their hard, unrelentless work that was put in, even though the results were not promising. Most of this was thanks to good work ethics, but also the fact that we worked daily on the project, so that it would not slip our mind even for a day. This did not mean we worked day and night though, we also remembered to have fun, and to enjoy the breaks we had, making the quality of our work superb.

Thank you for a great course!

References

[1] https://openbionicslabs.com/obtutorials/ada-v1-assembly

[2] http://www.instructables.com/id/Simple-Robotic-Gripper/

[2]Semig P., and Wells C. A current sensing tutorial. https://www.eetimes.com/document.asp?doc_id=1279404

[3]High-side current sense measurement: Circuits and Principles. https://www.maximintegrated.com/en/app-notes/index.mvp/id/746

[4]Almassri, A.M., Wan Hasan, W.Z., Ahmad, S.A., Ishak, A.J., Ghazali, A.M., Talib, D.N. and Wada, C., 2015. Pressure sensor: state of the art, design, and application for robotic hand. *Journal of Sensors*, *2015*.

[5] https://www.tme.eu/en/details/czn-cp1/pressure-sensors/iee/cp1-149ns/

[6]https://cdn-learn.adafruit.com/downloads/pdf/force-sensitive-resistor-fsr.pdf

[7] https://www.tme.eu/en/details/ten-tf3_120-w/pressure-sensors/tenmex/tf3120-k/

[8]https://pressureprofile.com/blog/2014/7/31/a-handy-guide-to-tactile-sensor-selection

[31]https://knowyourmeme.com/memes/gravity-gun