

# **INSTRUMENTATION D'UNE PINCER POUR UN ROBOT INDUSTRIEL**

-

## **RAPPORT FINAL**

## Sommaire

|   |    |
|---|----|
| Présentation rapide de la SAE .....           | 2  |
| Contexte de la SAE.....                       | 2  |
| But du travail .....                          | 2  |
| Planning .....                                | 2  |
| Planning & diagramme de Gantt théorique.....  | 2  |
| Planning & diagramme de Gantt réel .....      | 4  |
| Répartition des tâches .....                  | 5  |
| Analyse critique de nos travaux .....         | 5  |
| Synthèse sur l'étude des capteurs .....       | 5  |
| Mesures sur table.....                        | 7  |
| La matière de l'objet .....                   | 7  |
| La couleur de l'objet.....                    | 8  |
| L'orientation de l'objet.....                 | 9  |
| La taille de l'objet .....                    | 10 |
| Réalisation des mors .....                    | 11 |
| Réalisation du PCB.....                       | 12 |
| Réalisation du boîtier .....                  | 12 |
| Construction.....                             | 13 |
| Problèmes rencontrés.....                     | 13 |
| Réalisation Programme .....                   | 14 |
| Intégration du système sur la pince.....      | 15 |
| Synthèse .....                                | 19 |
| Perspective pour les suites éventuelles ..... | 20 |
| Annexes .....                                 | 21 |

# Présentation rapide de la SAE

## Contexte de la SAE

Sur un robot industriel, la pince est un des dispositifs qui permet au robot d'interagir avec son environnement. Grâce à une pince, le robot peut par exemple ramasser, tenir ou déplacer des objets. Sur les robots industriels présents à l'IUT, la pince est commandée par le robot, mais il n'y a pas de retour d'information quant à la saisie et aux collisions.

## But du travail

En tant qu'étudiant, notre but était donc d'instrumenter ces pinces pour avoir d'une part un retour sur la saisie d'un objet et d'autre part pour détecter une collision de la pince avec un objet environnant. Le but étant de récupérer le retour en temps réel pour améliorer la robustesse du système.

# Planning

## Planning & diagramme de Gantt théorique

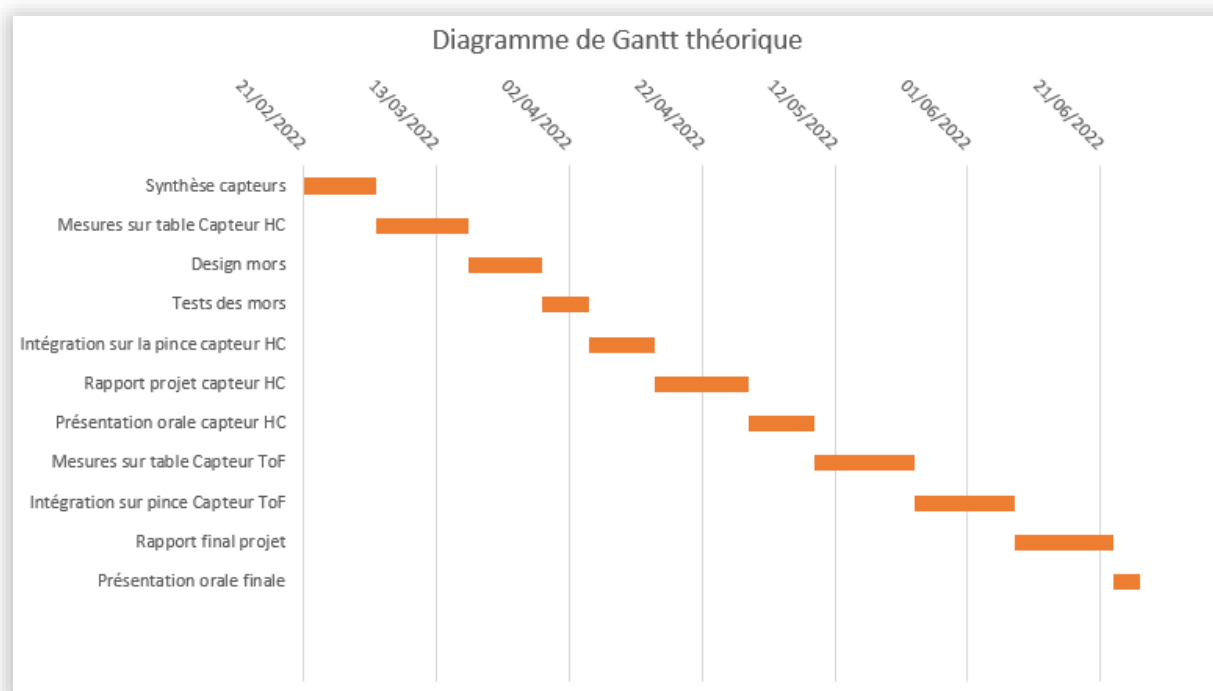
Nous avons, au début de notre projet, mis en place un planning théorique que nous avons essayé de respecter.

Nous avons retenu 11 étapes :

1. Synthèse capteurs
2. Mesures sur tables pour le capteur HC
3. Design mors
4. Tests des mors
5. Intégration sur la pince pour le capteur HC
6. Rapport de projet pour le capteur HC
7. Présentation orale pour le capteur HC
8. Mesures sur table pour le capteur ToF
9. Intégration sur la pince pour le capteur ToF
10. Rapport final du projet
11. Présentation orale finale

Ce planning s'étend du 21/02/2022 (la date de commencement de notre SAE) au 27/06/2022 (la date de notre présentation orale finale).

| Etapes                              | ▼ Date début ▼ | durée | ▼ Date fin ▼ |
|-------------------------------------|----------------|-------|--------------|
| Synthèse capteurs                   | 21/02/2022     | 11    | 04/03/2022   |
| Mesures sur table Capteur HC        | 04/03/2022     | 14    | 18/03/2022   |
| Design mors                         | 18/03/2022     | 11    | 29/03/2022   |
| Tests des mors                      | 29/03/2022     | 7     | 05/04/2022   |
| Intégration sur la pince capteur HC | 05/04/2022     | 10    | 15/04/2022   |
| Rapport projet capteur HC           | 15/04/2022     | 14    | 29/04/2022   |
| Présentation orale capteur HC       | 29/04/2022     | 10    | 09/05/2022   |
| Mesures sur table Capteur ToF       | 09/05/2022     | 15    | 24/05/2022   |
| Intégration sur pince Capteur ToF   | 24/05/2022     | 15    | 08/06/2022   |
| Rapport final projet                | 08/06/2022     | 15    | 23/06/2022   |
| Présentation orale finale           | 23/06/2022     | 4     | 27/06/2022   |



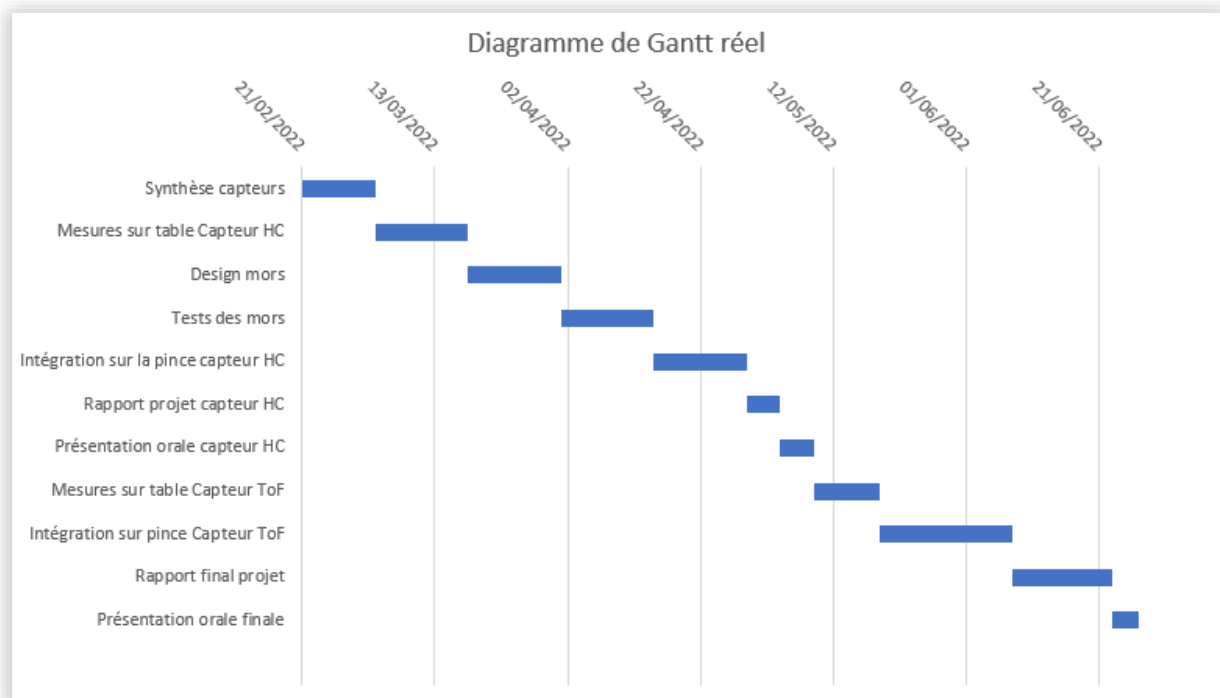
Dans le tableau, nous avons indiqué la date de commencement de chaque étape, la durée que l'étape aurait dû prendre, ainsi que les dates de fin des étapes.

En dessous du tableau, nous avons réalisé le diagramme de Gantt correspondant.

## Planning & diagramme de Gantt réel

Au cours de la réalisation de notre projet, nous n'avons pas pu respecter complètement le planning théorique annoncé.

| Etapes                              | Date début | durée | Date fin   |
|-------------------------------------|------------|-------|------------|
| Synthèse capteurs                   | 21/02/2022 | 11    | 04/03/2022 |
| Mesures sur table Capteur HC        | 04/03/2022 | 14    | 18/03/2022 |
| Design mors                         | 18/03/2022 | 14    | 01/04/2022 |
| Tests des mors                      | 01/04/2022 | 14    | 15/04/2022 |
| Intégration sur la pince capteur HC | 15/04/2022 | 14    | 29/04/2022 |
| Rapport projet capteur HC           | 29/04/2022 | 5     | 04/05/2022 |
| Présentation orale capteur HC       | 04/05/2022 | 5     | 09/05/2022 |
| Mesures sur table Capteur ToF       | 09/05/2022 | 10    | 19/05/2022 |
| Intégration sur pince Capteur ToF   | 19/05/2022 | 20    | 08/06/2022 |
| Rapport final projet                | 08/06/2022 | 15    | 23/06/2022 |
| Présentation orale finale           | 23/06/2022 | 4     | 27/06/2022 |



Nous constatons que certaines durées sont plus ou moins différentes selon les étapes dans le tableau ci-dessus, si on les compare aux durées du tableau du planning théorique.

On remarque, entre autres, que nous avons passé deux fois moins de temps que prévu à travailler sur le rapport de projet du capteur à ultrason et sur sa présentation orale.

## Répartition des tâches

Pour réaliser ce projet, nous nous sommes réparti les tâches à réaliser dans le but de travailler plus efficacement.

Alexandre a travaillé principalement sur la partie programmation de notre projet, il a réalisé une grande partie des programmes que nous utilisons.

Elliot et Romain se sont plus concentrés sur la conception des mors, du PCB et de l'intégration du système sur la pince. Ils ont également réalisé les mesures sur table et le test des mors sur la pince avec l'aide d'Alexandre.

Pour la réalisation de la synthèse sur les capteurs, des rapports et des présentations orales, nous avons également travaillé à 3.

## Analyse critique de nos travaux

### Synthèse sur l'étude des capteurs

Pour avoir un retour d'information concernant la saisie ou les collisions, nous avons à disposition plusieurs capteurs :

- Le capteur Capacitif CR30S
- Le capteur Ultrason HC-SR04
- Le capteur Photoélectrique VTF 180-2N4111
- Le capteur Laser ToF VL53L0X

Nous avons donc étudié ces 4 capteurs pour en savoir plus sur leur mode de fonctionnement et pour apprendre à les différencier. Pour chacun d'entre eux, nous avons dû expliquer dans un rapport de synthèse :

- Principe de fonctionnement
- Gamme de mesure, précision, fréquence
- Identifier le type de sortie
- Comment était-il alimenté et comment communiquer avec lui

- Enfin, quels choix étaient les plus optimaux pour l'intégrer sur la pince

Nous allons donc vous parler de ce que nous avons retenu de cette étude.

Concernant le capteur laser ToF, celui-ci peut être alimenté sous une tension comprise entre de 2.6V et 3.5V ce qui est compatible avec les GPIO de la Raspberry Pi de 3.3V. Il peut ainsi être commandé et branché directement depuis la Raspberry Pi, ce qui facilite grandement son installation sur la pince. De plus, sa gamme de mesure est cohérente avec les distances que nous voulons mesurer. Ce capteur peut donc être utilisé dans le cadre de l'instrumentalisation d'une pince industrielle.

Le capteur capacitif CR30S peut être utilisé pour s'arrêter avec une grande fiabilité devant un objet, ou encore d'envoyer un signal de confirmation de fermeture de la pince. Malheureusement, sa tension d'alimentation étant supérieur à 10V, son utilisation sur Raspberry pi n'est pas possible sans l'aide d'une alimentation externe supplémentaire, car le Raspberry ne peut fournir que 5V au maximum.

Le capteur à ultrasons HC-SR04 est un capteur très polyvalent, sa petite taille, sa gamme de mesure large qui reste précise de 3cm à 400cm, sa facilité d'utilisation et sa tension d'utilisation de 5V qui est compatible avec la Raspberry Pi le rend très adapté dans le cadre de notre SAE.

Le capteur Photoélectrique VTF 180-2N41112, pourrait être utilisé avec nos mors afin de détecter la présence des pièces près de la pince pour faciliter la manipulation de la pince ou encore éviter les collisions. Cependant, la tension d'alimentation de ce capteur, qui est de 10V DC à 30V DC est trop grande pour pouvoir être exploitée avec un Raspberry Pi sans l'aide d'une alimentation externe, car ses pins ne délivrent qu'une tension de 3,3V ou 5V maximum.

### **Conclusion :**

Parmi tous les capteurs que nous avons étudiés, le capteur laser ToF VL53L0X ainsi que le capteur à ultrasons HC-SR04 sont, selon nous, les plus faciles et pratiques à utiliser avec un Raspberry Pi. Il n'est cependant pas impossible d'utiliser les autres capteurs, à condition d'utiliser une alimentation externe pour compenser la limite de 5V délivrés par la Raspberry Pi.

## Mesures sur table

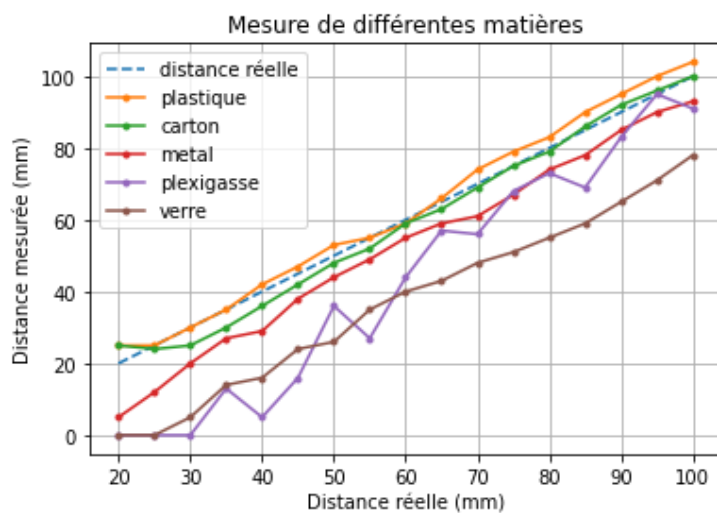
Nous avons tout d'abord effectué plusieurs mesures de distances sur table, avec différents types d'objets, pour tester les limites réelles du capteur. Ces tests ont pour but d'évaluer la précision de nos mesures en fonction de différentes conditions. Pour mieux présenter nos résultats, nous avons classé nos mesures en fonction des critères suivants :

- La matière de l'objet
- La couleur de l'objet
- L'orientation de l'objet
- La taille de l'objet

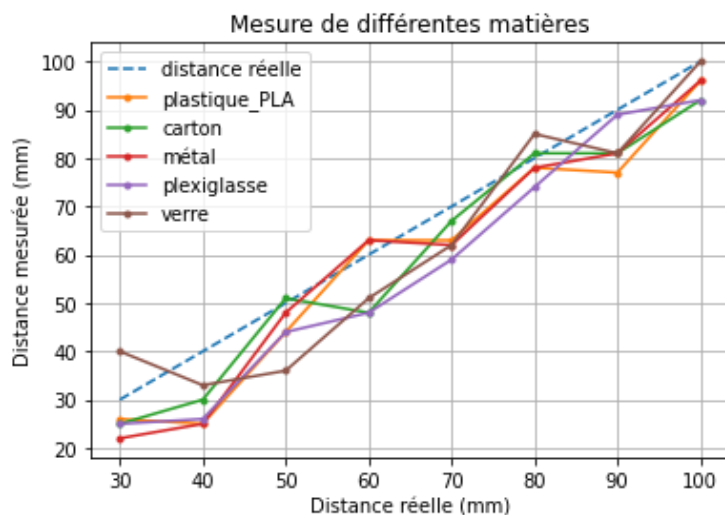
### La matière de l'objet

Pour déterminer si la matière de l'objet a un impact sur la précision de nos mesures, nous avons décidé de prendre plusieurs objets avec des matières différentes, à savoir : du plastique, du carton, du métal, du plexiglas et du verre. Nous avons ensuite fait une série de mesures sur plusieurs distances.

Voici les résultats que nous avons obtenus :



Capteur laser



Capteur à  
ultrasons



Si l'on observe le graph du capteur laser, on s'aperçoit que les matières qui diffractent la lumière ou laissent passer la lumière (le verre et le plexiglas) ont tendance à s'éloigner de la distance réelle. Tandis que les matières qui diffusent la lumière, telles que le plastique ou le carton, sont plutôt proche de la réalité.

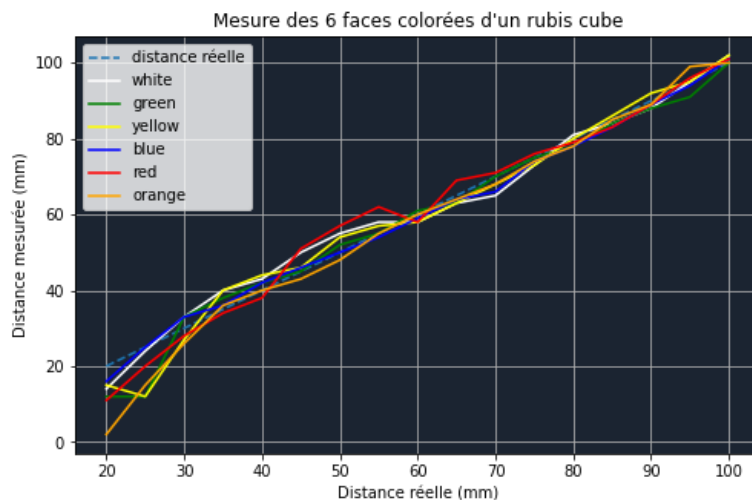
On peut donc en conclure que le capteur laser est inutilisable sur des objets qui diffracte la lumière tels que des objets en verre ou en plexiglas. Cela ne nous pose pas de problème, car notre capteur ToF va être utilisé face à une surface en plastique PLA.

Concernant le capteur à ultrasons, on ne remarque pas de grandes différences entre les courbes, on en déduit que la matière importe peu pour la qualité des mesures de ce capteur.

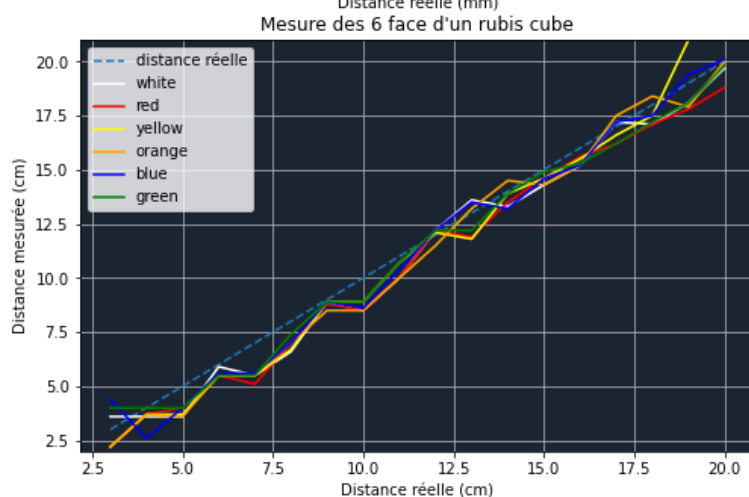
### La couleur de l'objet

Pour déterminer si la couleur de l'objet avait un impact sur la précision de nos mesures, nous avons décidé de prendre un Rubik's Cube pour faire des mesures avec chacune de ses faces. Sachant que le Rubik's Cube est en plastique (l'une des matières les plus fidèles comme montré précédemment) seules les différentes couleurs rentrent en jeu.

Voici les résultats que nous avons obtenus :



Capteur laser



Capteur à ultrasons

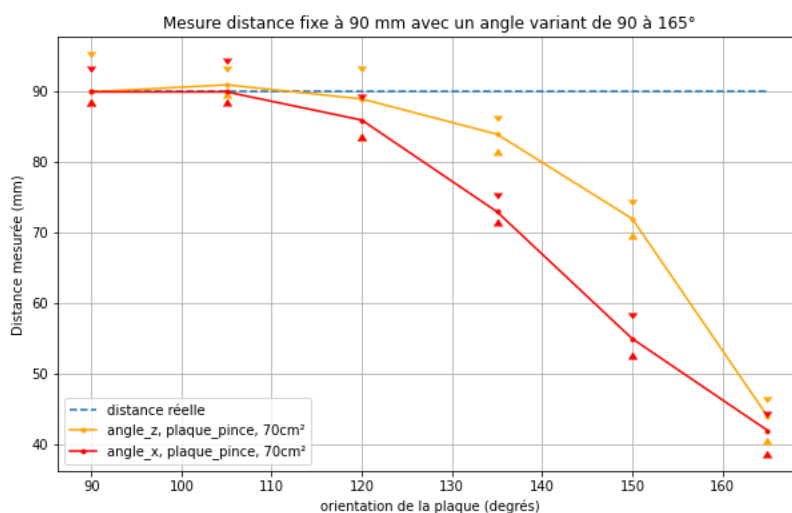
D'après les graphes que nous avons obtenu pour le capteur laser et le capteur à ultrasons, on remarque qu'il n'y a aucune différence et qu'elles convergent toutes vers la distance réelle.

Ainsi, on peut dire que la couleur de l'objet n'influe pas sur la précision de nos mesures que ce soit avec le capteur laser ou le capteur à ultrasons.

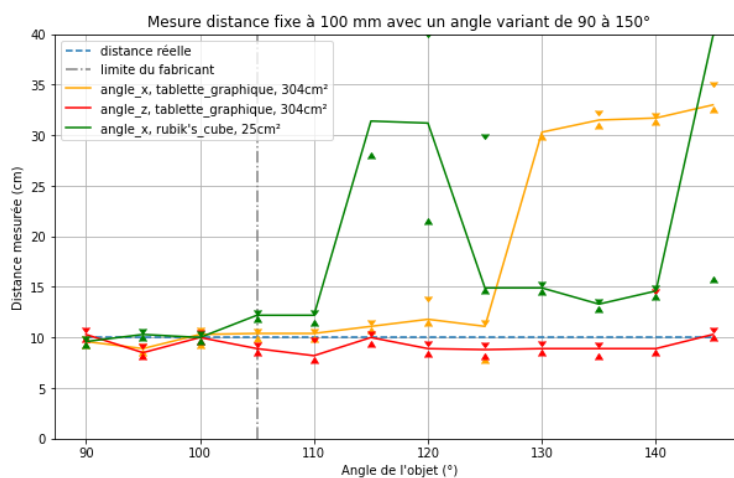
### L'orientation de l'objet

Pour déterminer si l'orientation de l'objet a un impact sur la précision de nos mesures, nous avons décidé de prendre une plaque en plastique pour faire des mesures en changeant l'orientation du capteur en fonction de la plaque.

Voici les résultats que nous avons obtenus :



Capteur laser



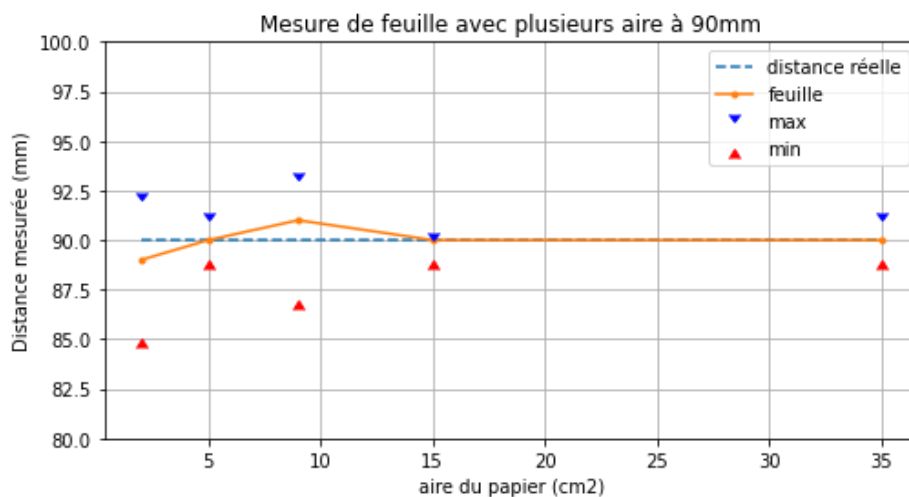
Capteur à  
ultrasons

D'après le graphe que nous avons obtenu, on remarque que lorsqu'on modifie l'angle (que ce soit de façon horizontale ou verticale), plus l'orientation de la plaque s'éloigne de 90 degrés, plus la distance mesurée est faussée. À partir de  $110^\circ$  les mesures sont inexploitable. Si l'on souhaite obtenir une mesure précise, il faut que l'objet soit positionné de façon perpendiculaire par rapport au capteur laser.

### La taille de l'objet

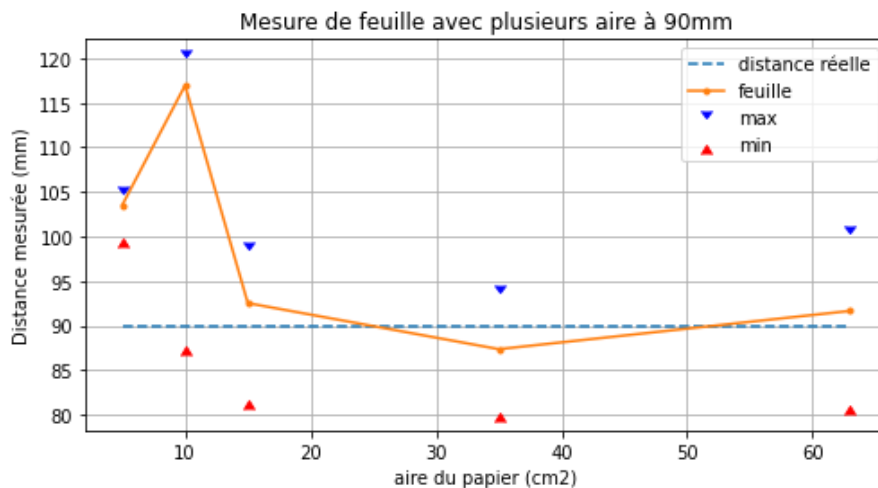
Enfin, concernant la taille de l'objet, nous avons décidé de prendre une feuille, que nous avons pliée en 2 entre chaque mesure. Ainsi, nous avons divisé par 2 la surface de la feuille (et donc la taille de l'objet) pour chacune des mesures prises par le capteur

Voici les résultats que nous avons obtenus :



Capteur laser

Le graphe que nous avons obtenu, représentant la mesure de la distance en fonction de la surface de la feuille, est plus ou moins une droite. Ceci signifie que la taille d'un objet n'a pas d'incidence sur les mesures que prend le capteur ToF.



Capteur à  
ultrason

Nous avons également fait les mêmes mesures avec le capteur HC, on constate toute de suite qu'avec ce capteur, une surface trop petite a des conséquences néfastes sur la qualité de la mesure. A partir de 15cm<sup>2</sup> les mesures sont exploitables, mais pas en dessous.

## Réalisation des mors

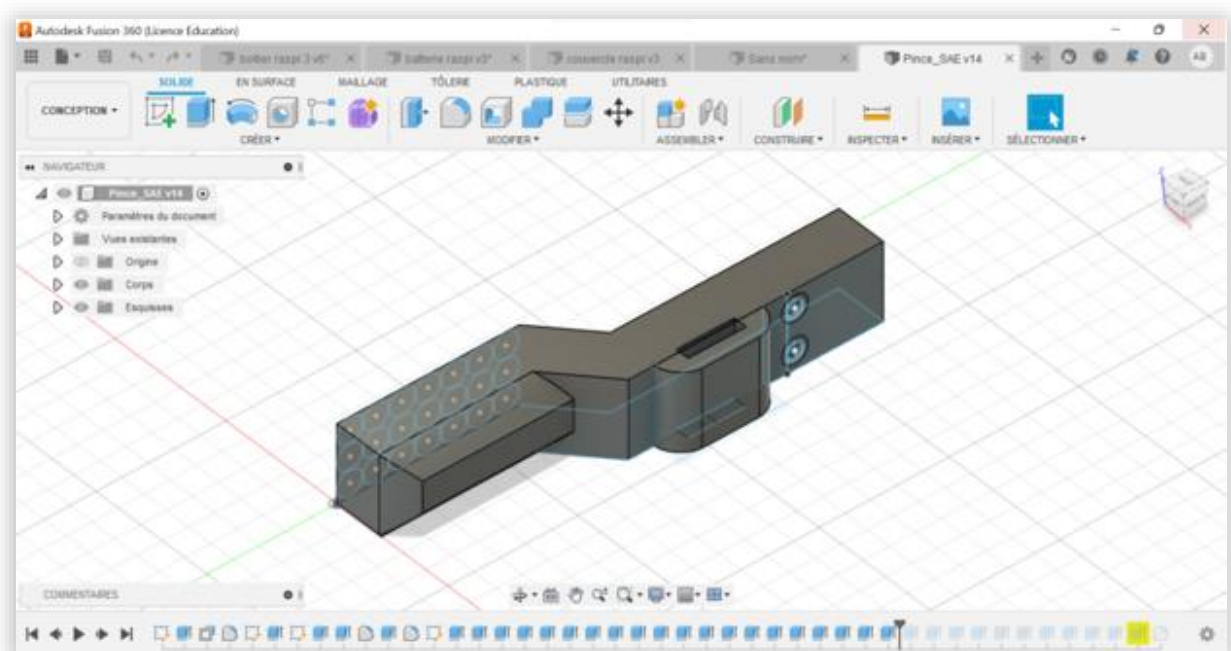
Afin d'intégrer le capteur sur les mors il nous faut designer des mors compatibles avec les 2 capteurs, ce qui est réalisable, car les 2 capteurs ne demandent que 4 pins de connexion. L'emplacement des pins va être utilisé comme support pour maintenir le capteur en place.

Pour concevoir les mors que nous devons intégrer sur la pince, nous avons utilisé le logiciel de modélisation Autodesk Fusion 360.

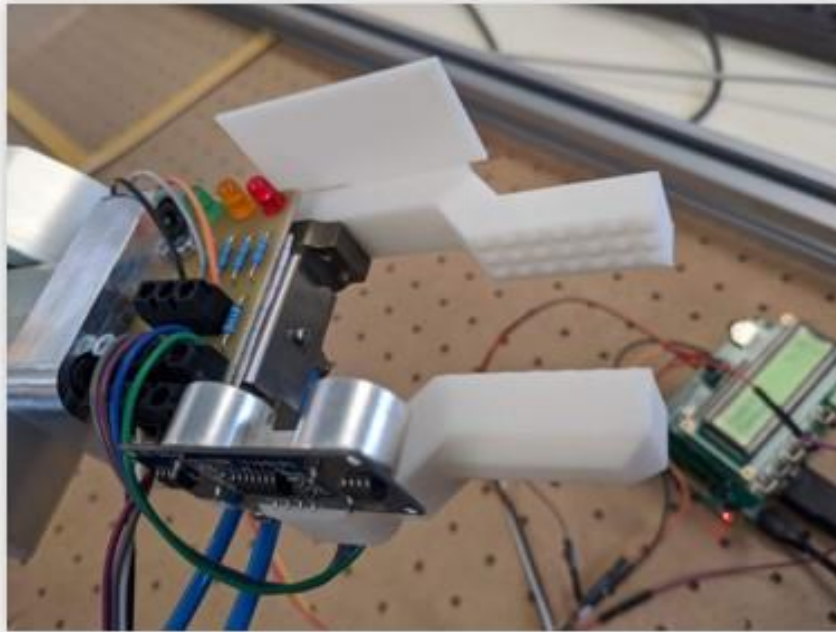
Nous nous sommes basés sur ces dimensions imposées :

- Taille minimum entre les mors pince fermé de 2,5cm.
- Surface de contact avec les objets pour les mors (1,5\*2cm).
- Taille minimum des câbles entre le Raspberry et la pince 30 cm.

Lors de la conception de nos mors, nous avons fait attention à bien réfléchir à l'intégration du capteur sur ceux-ci. Nous avons décidé de fixer le capteur sur le haut du mors droit, et d'insérer une plaque sur le mors gauche, le tout créé par une imprimante 3D. Puisque nous avons décidé d'implémenter le PCB



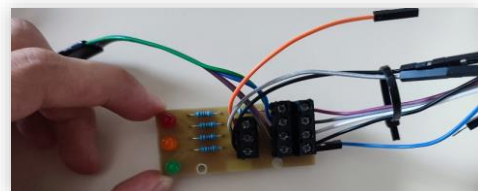
sur la pince, nous n'avons pas besoin de modifier nos mors pour y intégrer le PCB. Le principe de fonctionnement est simple, le capteur à ultrasons ou laser va envoyer des ondes sonores ou électromagnétiques en direction de la plaque en plastique. Les ondes vont rebondir sur cette dernière pour ensuite se propager dans la direction inverse, vers le capteur. Le temps de vol de l'onde entre son émission et sa réception est utilisé pour calculer la distance entre le capteur et la plaque.



### Réalisation du PCB.

Nous avons également réalisé un PCB sur Kiccad pour notre système. Ce PCB permet de réaliser la liaison électrique entre le capteur et le Raspberry Pi. Nous avons placé des borniers pour les liaisons électriques, des résistances et il y a également 3 LEDs rouge, orange et verte respectivement qui permettent d'indiquer l'état de la pince.

Notre PCB est assez petit, car nous l'intégrons sur la pince directement et la place disponible est limitée.



### Réalisation du boîtier

Le boîtier permet de protéger le circuit de la Raspberry Pi et ses modules de l'extérieur. Le but étant de pouvoir fixer le boîtier sur le bras du robot et ainsi limiter au maximum les longs câbles qui parcourent le bras robotique pour arriver sur la table, car cela peut limiter les mouvements du robot.

## Construction

La base du boîtier doit pouvoir maintenir fermement la Raspberry Pi. Pour cela, nous utilisons les 4 trous disponibles sur cette dernière. On trouve facilement sur internet les dimensions précises de la Raspberry Pi, ainsi la modélisation correspond parfaitement à la réalité. Pour se fixer sur ces trous, nous utilisons 4 trous taraudés pouvant accueillir une vis M2.5.

Un trou est placé sur le bas-côté pour faire entrer le câble micro USB d'alimentation.

Le couvercle du boîtier est imprimé à part. Il comporte 2 rectangles vides : l'un pour faire ressortir l'écran LCD, l'autre pour sortir les 12 câbles de breadboard connecté à la Raspberry Pi.

Pour maintenir ces câbles, nous utilisons de la colle chaude.

Sur 2 faces du boîtier, 9 trous ont été ajoutés afin de créer une ventilation, indispensable pour le micro-ordinateur qui dégage une chaleur non négligeable.

Pour alimenter la Raspberry Pi tout en restant indépendant du réseau électrique sur secteur, nous avons opté pour une batterie externe de 20 000 mAh ainsi qu'un câble micro USB coudé de 25cm pour une meilleure ergonomie.

## Problèmes rencontrés

Le bouton/joystick est très ergonomique à utiliser, nous avons essayé de l'utiliser avec le boîtier, mais malheureusement il est très difficile à utiliser une fois dans le boîtier, alors que les boutons ont juste besoin d'une rallonge. Nous avons donc abandonné le bouton/joystick pour se concentrer sur les 5 boutons de l'avant.

Les encoches du couvercle nous ont posé beaucoup de problèmes. En effet, ils doivent être parfaitement identiques aux encoches femelles du boîtier, mais avec une échelle inférieure de l'ordre du 10<sup>e</sup> de millimètre pour pouvoir s'emboîter.

L'emplacement des trous sur le boîtier pour communiquer avec les boutons nous a valu plusieurs ratés de positionnement. Nous ne pouvions pas mesurer avec précision à quelles hauteurs ils étaient placés, car la structure des 2 modules n'est pas fixe. De plus, toute la modélisation devait avoir comme référence les 4 trous de fixation de la Raspberry Pi.

#### **Boîtier dans sa version finale contenant la Raspberry Pi et ses modules**



#### Réalisation Programme

L'entièreté du programme a été développé en Python 3.9. Le programme principale (voir [annexe](#)) repose sur des fonctions asynchrones. Elles permettent de s'exécuter en même temps, par concurrence, c'est-à-dire lorsque qu'une fonction ne fait rien (pause *await*) une autre s'exécute. Toutes ces fonctions en concurrence vont appeler des fonctions qui se trouvent dans un autre fichier : `appli.py`. Ce fichier contient une classe Python qui gère l'entièreté de l'interface et du système de mesure. Le principe de classe est indispensable ici car il permet de se passer des variables globales en utilisant *self* (un objet contenant toutes les variables et fonctions créés dans la classe) dans chaque paramètre de fonction (appelé méthode dans une classe).

Sur la partie `appli.py` on peut voir le fonctionnement de la méthode d'initialisation, elle va rechercher des fichiers texte qui contiennent des valeurs sauvegardées de la dernière exécution du programme. La

détection de l'état de la pince se fait en fonction de la distance mesurer par le capteur. Cette mesure est appelée via une fonction qui se trouve dans un autre fichier. Soit le programme va accéder au fichier HCS404.py si le capteur utilisé est l'ultrason, soit dans le fichier VL53L0X.py si le capteur utilisé est le laser.

Nous avons rencontré beaucoup de problème au cours du développement, notamment sur les fonctions asynchrones et les classes qui étaient des nouvelles notions à assimilé pour nous. Alexandre nous à beaucoup aidé grâce à ses connaissances avancées dans le langage Python.

### Intégration du système sur la pince

Après avoir conçu notre système, nous allons maintenant parler de l'intégration de celui-ci sur la pince.

Tout d'abord, pour intégrer les 2 mors imprimés en 3D sur la pince, nous avons utilisé les 4 trous taraudés de chaque côté de la pince pour les fixer sur celle-ci.

Nous avons également réalisé le circuit électrique comprenant la Raspberry Pi, son alimentation, un optocoupleur et le capteur.

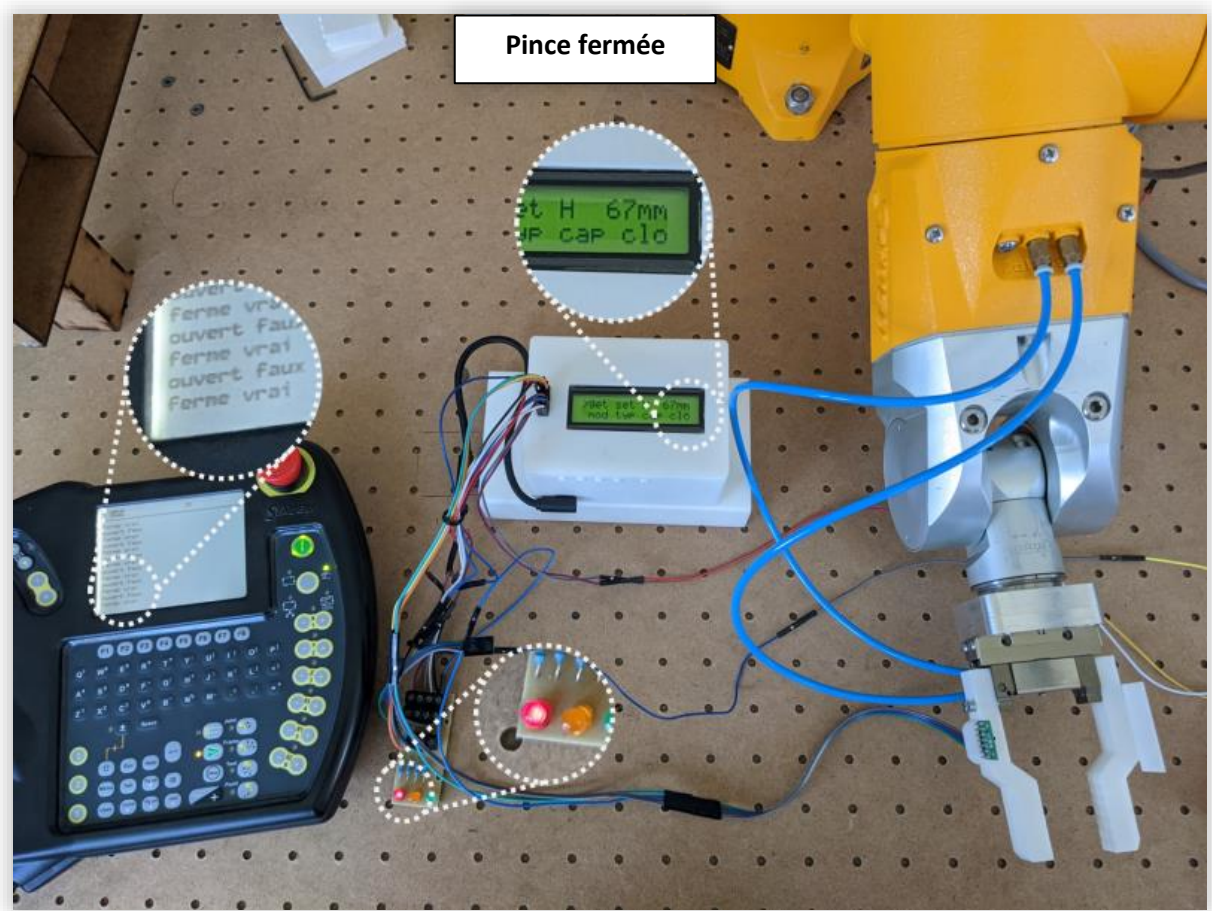
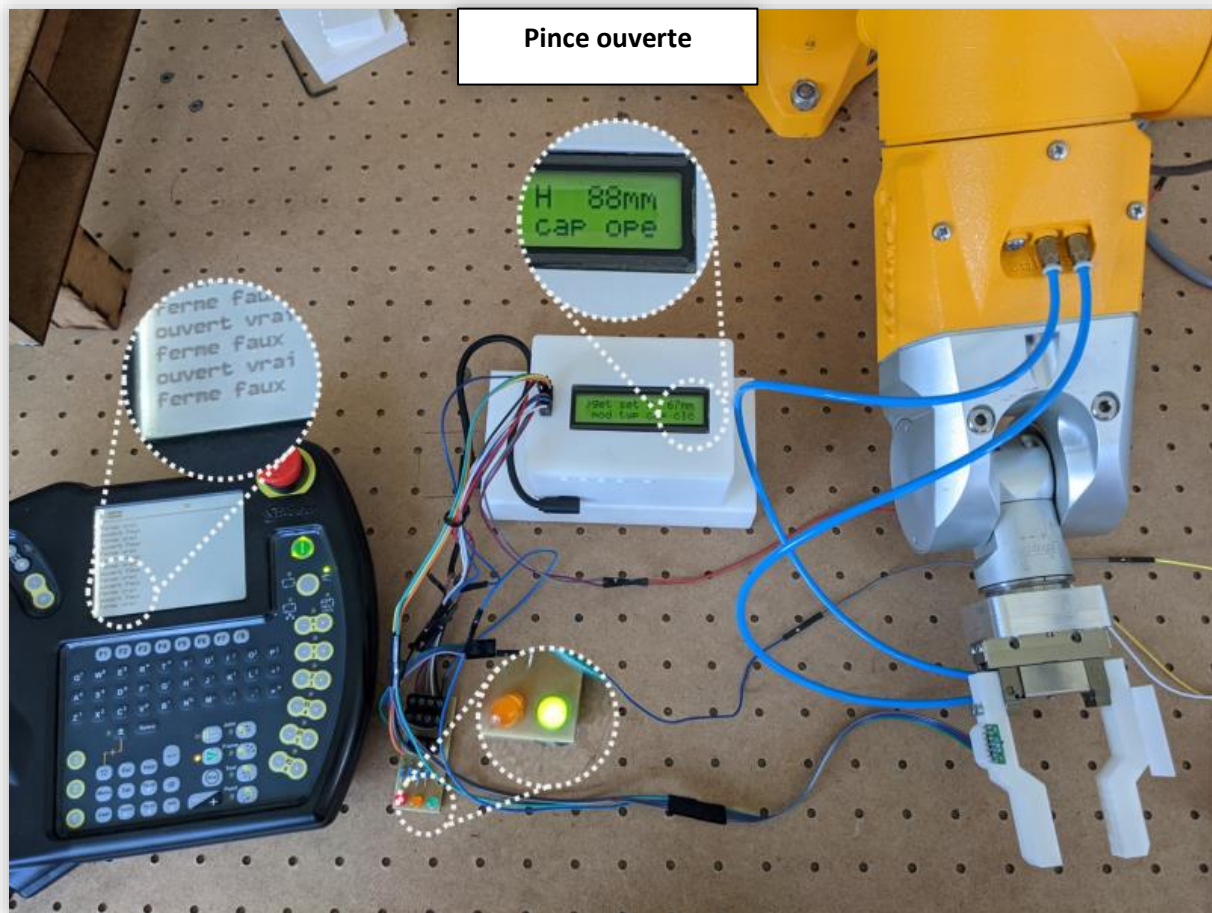
Grâce à l'aide d'un optocoupleur, nous pouvons envoyer un signal de 24V aux entrées TOR, à l'aide du signal sortant de la Raspberry Pi de 3.3V.

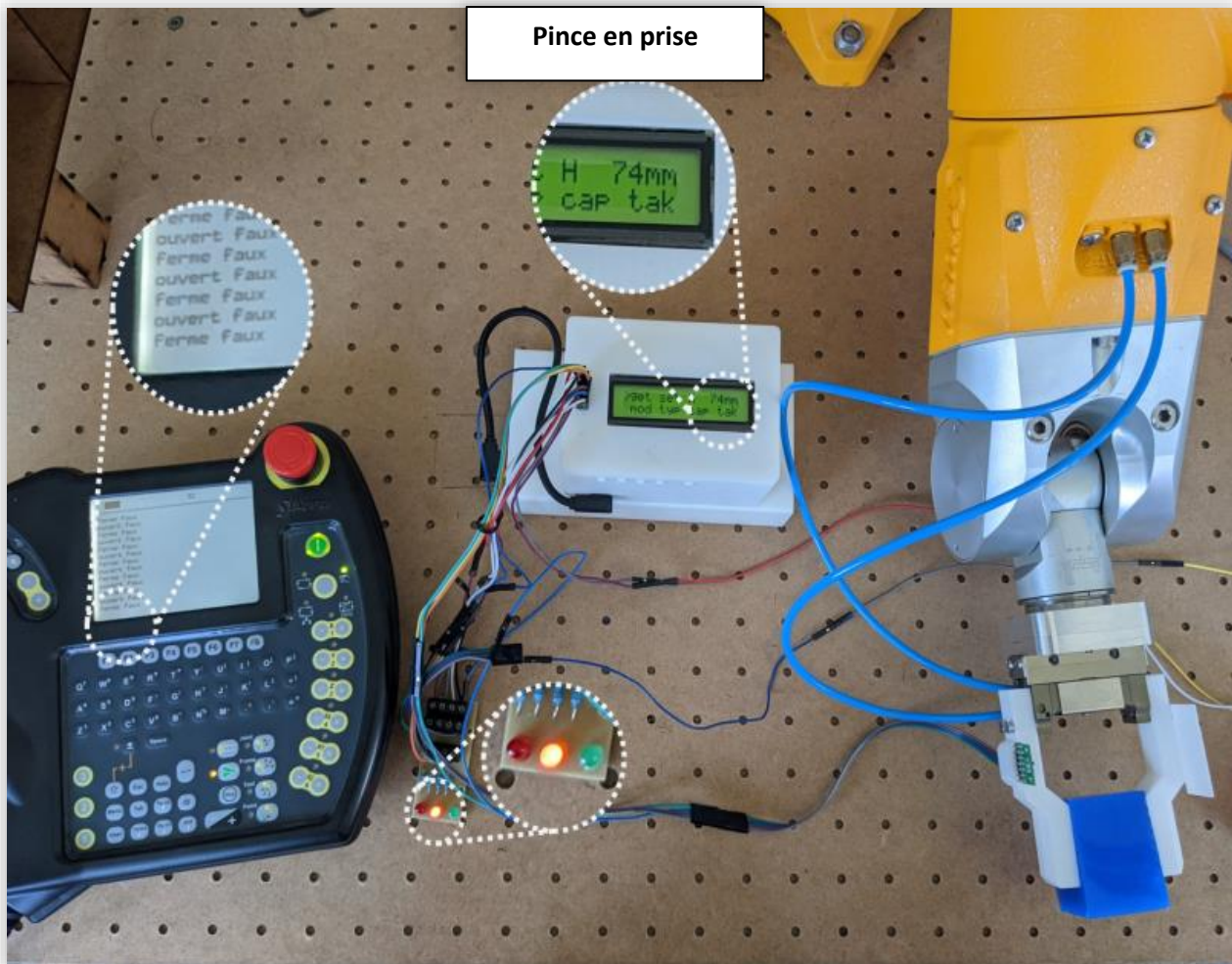
L'optocoupleur peut rendre des circuits électriques indépendants. En effet, l'optocoupleur est un composant électronique permettant de transmettre un signal d'un circuit électrique à un autre sans être relié électriquement avec lui permettant de garantir l'isolation galvanique. Il est formé d'une LED infrarouge et d'un phototransistor.

Lorsqu'un courant électrique passe dans la LED, elle va émettre un signal infrarouge dans un boîtier hermétique à la lumière. Le phototransistor va ensuite capter la lumière infrarouge et devenir passant pour transmettre le courant électrique.

Ci-dessous le résultat final de notre système fonctionnel. Une batterie externe de 20000mAh est placée sous le boîtier







Lors de nos tests en condition réels sur le bras robotique, nous pouvions choisir la précision du capteur ToF, il en possède 4 : speed, good, better et best. Le mode speed est le moins précis des 4, mais à l'inverse du mode best, il est le plus rapide avec un taux de rafraîchissement de 50 mesures / seconde contre 30/s en good, 15/s en better et 5/s en best.

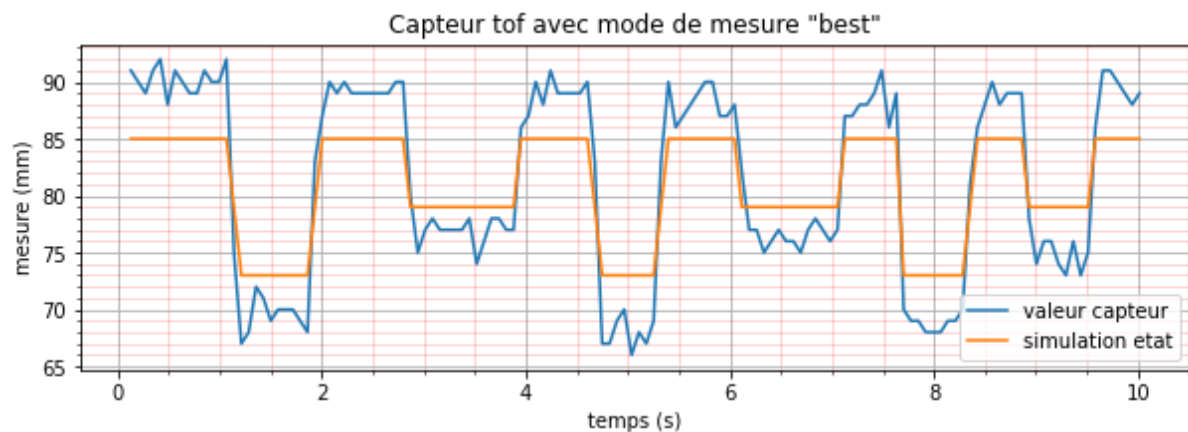
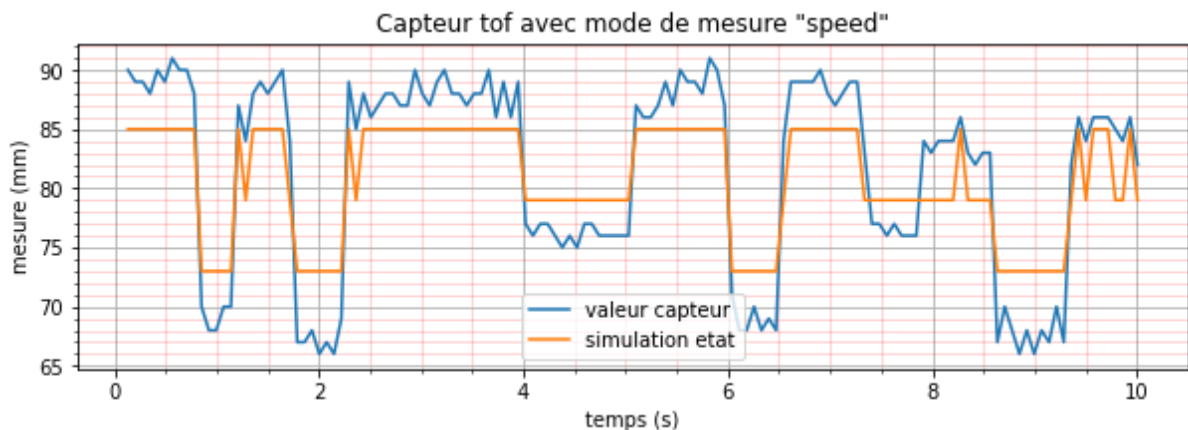
Tous ces modes sont implémentés dans le module open source du capteur ToF que nous avons récupéré sur [GitHub](https://github.com). Pour avoir une similarité avec le capteur ultrason, nous avons créé une classe Python qui possède la même gestion de ces 4 modes.

Pour revenir à nos tests réels, il est clair qu'en choisissant le mode best, on obtient de meilleurs résultats dans le maintien d'un état détecter de la pince au cours du temps. Si on passe à des modes moins précis, il en résulte des fluctuations de l'état (par exemple la pince est ouverte et l'état

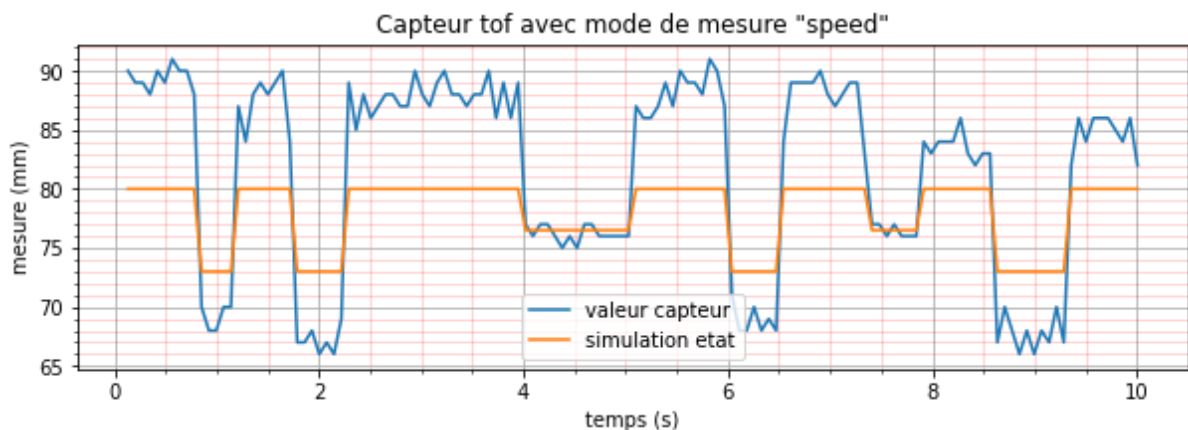


retranscrit varie entre ouvert et en prise). Pour essayer de comprendre pourquoi ce phénomène se produit, nous avons réalisé des mesures semblables à un oscillogramme.

La courbe bleue représente les mesures du capteur au cours du temps, en orange la courbe est la retranscription de l'état de la pince au cours du temps. Les 3 états sont facilement lisibles : ouvert = 85mm, en prise=79mm, fermé=74mm.



On remarque tout de suite que pour le même réglage des valeurs limite, le mode best est beaucoup plus précis et comporte moins d'erreur. Le même phénomène se produit avec le capteur ultrason. Comme nous l'avons vu le mode best comporte moins d'erreur, mais il permet que de fournir 5



mesures par secondes. La première solution qui vient à l'esprit serait d'utiliser un compromis entre speed et best comme good ou better, mais il existe une 2eme solution qui consiste à modifier les valeurs limites d'état.

En changeant la valeur max de 85mm à 80mm, on obtient un signal aussi propre que celui du mode best. Ainsi, nous pouvons allier la haute fréquence de mesure de ce mode sans obtenir d'erreur. Néanmoins, une nouvelle contrainte est créée : l'état en prise, devient très serré entre le max et le min. Les objets pris par la pince doivent alors obligatoirement être d'une taille proche de 77mm.

On peut donc définir le théorème suivant : plus les valeurs limites d'états sont éloignées, plus la taille de l'objet entre les mors peut varier. La réciproque est aussi vraie.

## Synthèse

Au cours de cette année, nous avons appris à instrumenter une pince pour améliorer sa robustesse. L'amélioration majeure apportée est un retour d'information sur l'état de la pince via un capteur de distance. Le retour d'information se faisant au travers de 3 LEDs, de l'écran LCD mais aussi sur la télécommande du robot. Pour intégrer le capteur, nous avons créé une nouvelle version des mors. Nous avons découvert, pour certain, la modélisation 3D ainsi que l'impression 3D. Le système est embarqué via un Raspberry Pi et le module PifaceCad comportant un écran LCD et des boutons. Nous avons appris à nous servir de Linux et enrichi nos connaissances en Python.

Nous avons réalisé le projet sur une période de 126 jours et nous avons pu respecter les délais pour chaque tâche demandée.

### Tableau des coûts du projet :

| <u>Objet</u>   | <u>Coût</u>  |
|--|--------------|
| Raspberry Pi + PifaceCad                                     | 56,88€ (TTC) |
| Composants électronique (LED, résistances, borniers, câbles) | 3,00€ (TTC)  |
| Capteur ToF  | 16,00€ (TTC) |
| Capteur HC   | 5,00€ (TTC)  |
| Plastique imprimante 3D                                      | 10,00€ (TTC) |

|  |                      |
|--|----------------------|
| Outils (pince coupante, tournevis, etc...) | 20,00€ (TTC)         |
| <b>TOTAL :</b>                             | <b>110,88€ (TTC)</b> |

## Perspective pour les suites éventuelles

Après la réalisation de ce projet, notre système permet l'instrumentalisation d'une pince, ce qui nous ouvre la voie à de nombreuses perspectives.

Tout d'abord, on pourrait améliorer la stabilité du capteur sur les mors, ainsi que celle de la plaque fixée sur un des mors pour ainsi améliorer l'exactitude des mesures et du retour d'information concernant la prise d'objet. On envisagerait également d'améliorer le circuit électrique reliant le système, en implémentant, entre autres, l'optocoupleur dans le circuit et en travaillant sur la gestion des câbles pour obtenir un système plus compact et mieux rangé.

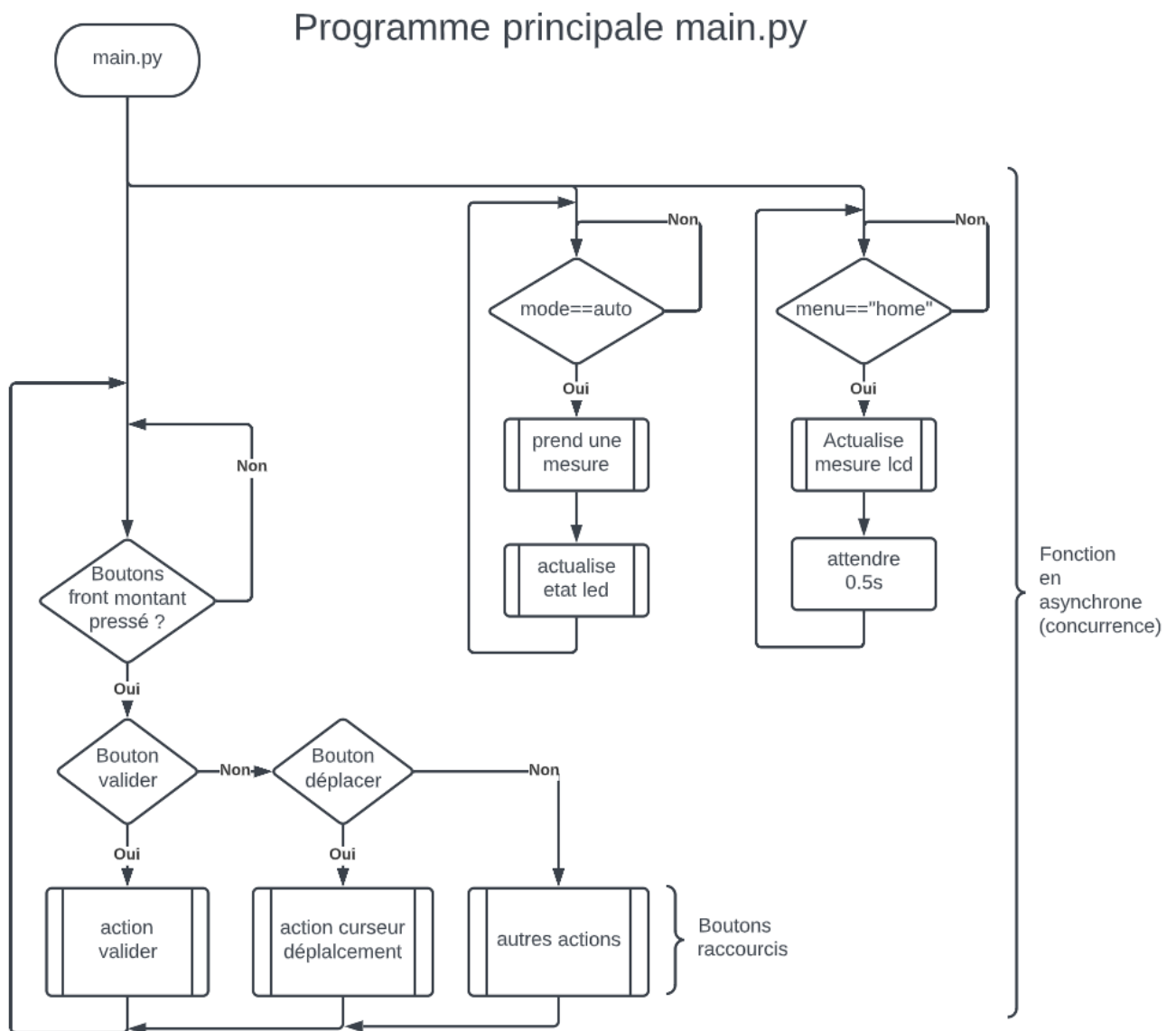
En effet, ce système pourra améliorer la prise d'objet de la pince, tout en implémentant une détection de collisions.

Du fait de cette instrumentalisation de la pince, on peut imaginer que l'on pourra plus facilement résoudre certains problèmes, dont celui des Tours d'Hanoï, qui requiert une précision accrue de la prise d'objet faite par la pince.

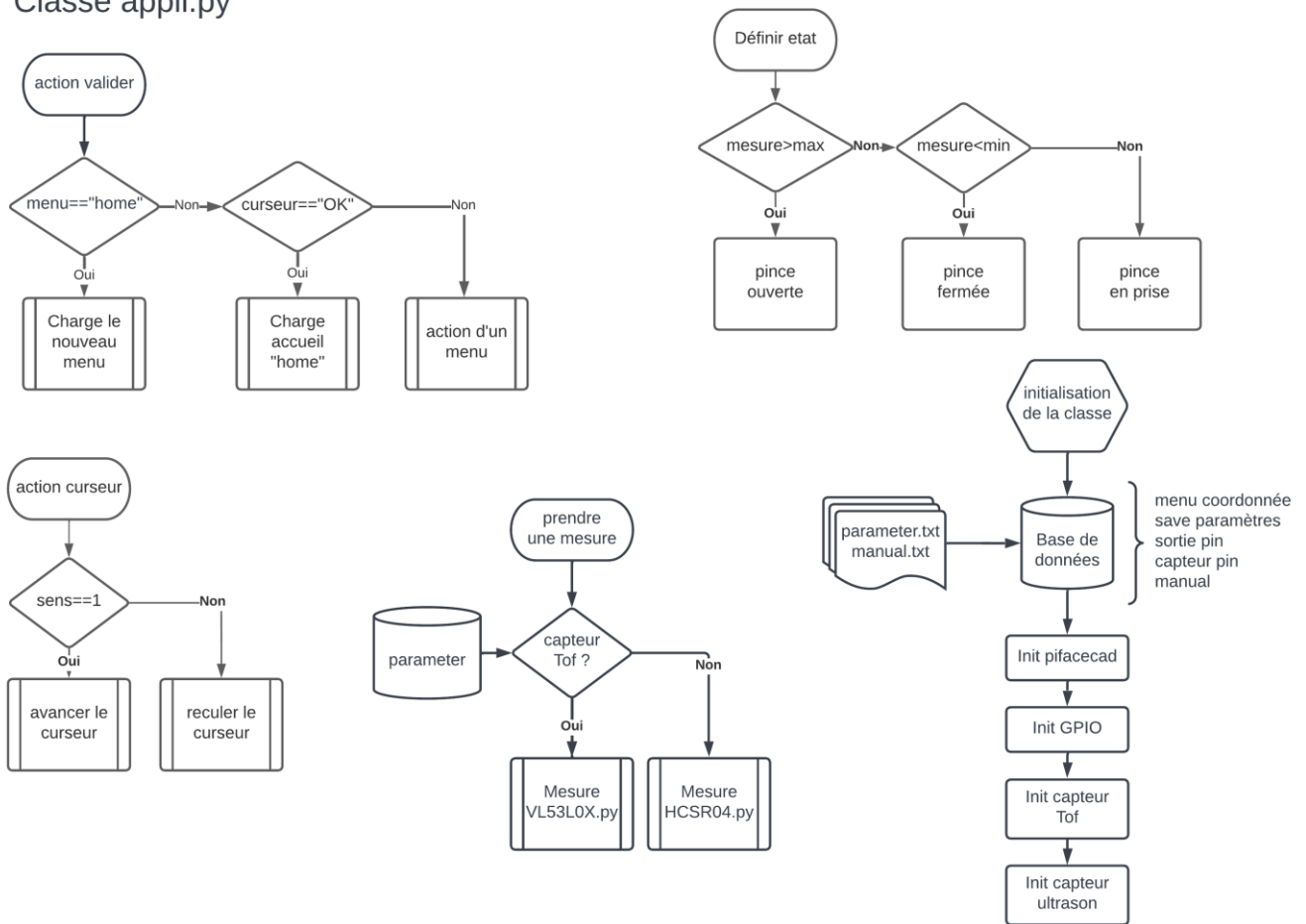
On pourra également espérer créer l'automatisation de certaines tâches grâce au système à l'aide de la création de programmes supplémentaires.

## Annexes

Organigramme de l'application en Python :



## Classe appli.py



Toutes les ressources (programme, model 3D) disponible à l'adresse suivante :

<https://github.com/arduilex/SAE-pince-industrielle>

Pièces 3D :



Boîtier RaspberryPi



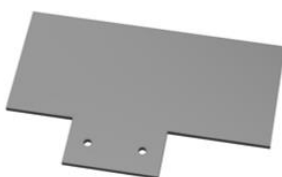
Couvercle



Boîtier batterie



Clips



La plaque



Les mors



Les boutons