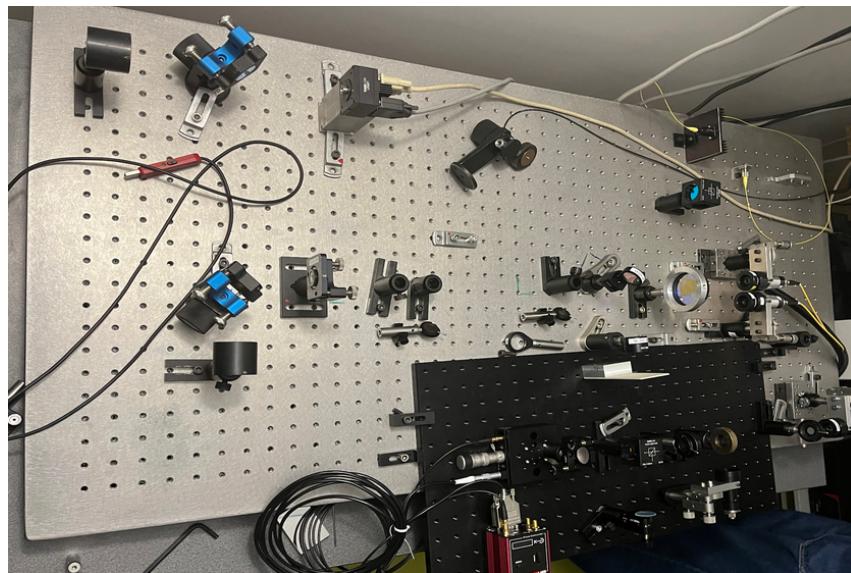




INSTITUT D'OPTIQUE GRADUATE SCHOOL

Rapport technique PROTIS

Della Vittoria Louis, Oulyadi Helwan, Chan-Lap Étienne, Farokhpéy Darius



Nous attestons que ce travail est original, que nous citons en référence toutes les sources utilisées et qu'il ne comporte pas de plagiat

11 avril 2025

Table des matières

1	Introduction	2
2	Contexte	2
3	Rappel des objectifs	2
4	Cahier des charges	2
5	Schéma fonctionnel	2
5.1	L'interfaçage sous Python	3
5.1.1	Objectifs	3
5.1.2	Solutions	3
5.1.3	Tests de validation réalisés et résultats obtenus	4
5.2	Le calcul de la fonction de corrélation numérique	4
5.2.1	Objectifs	5
5.2.2	Solution	5
5.2.3	Tests de validation réalisés et résultats obtenus	5
5.3	Le contrôle de la cale piézo-électrique	6
5.3.1	Objectifs	6
5.3.2	Solution	6
5.3.3	Tests de validation réalisés et résultats obtenus	6
6	Notice d'utilisation	7
7	Bilan du projet	7
7.1	Partie technique	7
7.2	Travail en équipe	8
8	Conclusion	9

1 Introduction

Dans ce rapport technique, nous allons présenter le travail réalisé au cours des différentes séances de PROTIS sur le projet DEPhI d'interférences à un photon. Notre projet s'est concentré sur la modernisation de l'interface déjà présente sur la manipulation.

2 Contexte

L'objectif du projet DEPhI interférences à 1 photon est d'améliorer le TP QLMN qui étudie l'effet Hong-Ou-Mandel en ajoutant un module permettant de réaliser des interférences de photons uniques. Ce projet est séparé en 3 parties : une partie optique réalisé lors des semaines DEPhI, une partie électronique qui se concentre sur le circuit de détection et qui est réalisé par un autre groupe PROTIS et enfin une partie interfaçage réalisée par notre groupe.

3 Rappel des objectifs

L'objectif principal de notre projet est de réaliser une interface codée en Python qui, à partir des données renvoyées par une carte Nucléo, codée en C, permet de tracer le nombre de coïncidences en fonction du temps, de calculer et d'afficher la fonction de corrélation, $g^{(2)}$, pour un montage à 2 ou 3 détecteurs.

4 Cahier des charges

Cet objectif sera accompli au travers de diverses contraintes au sein du cahier des charges suivant :

- Mettre en place la liaison-série entre Python et la carte Nucléo : Pouvoir récupérer des données toutes les secondes
- Concevoir une interface complète et modulable à l'aide de *PyQT5*.
- Réussir à récupérer et à afficher un signal périodique sur l'interface Python
- Calculer et afficher le $g^{(2)}$
- Contrôler la cale piézo-électrique de la manipulation

Ces différents critères représentent la majeure partie de notre travail.

5 Schéma fonctionnel

Le schéma fonctionnel du projet est présenté ci-après :

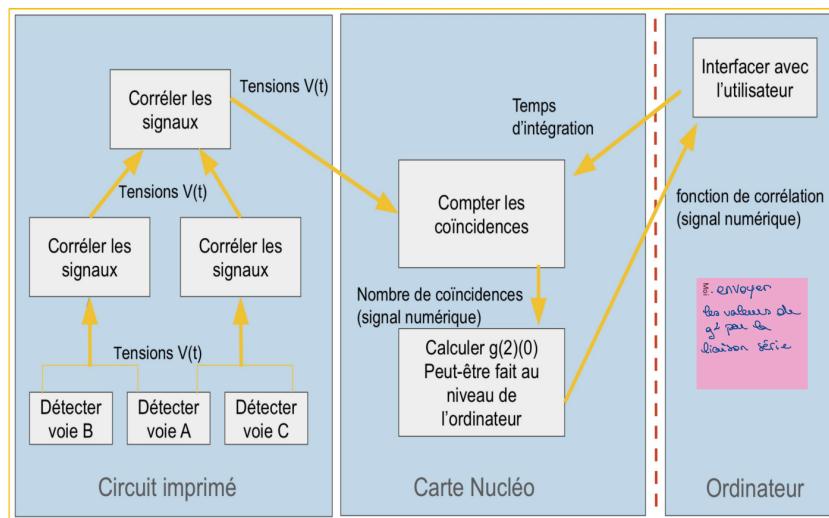


FIGURE 1 – Schéma fonctionnel du projet

Notre travail s'est concentré sur la partie ordinateur du schéma ainsi que sur le lien entre l'ordinateur et la carte Nucléo. Le diagramme de classe présente les différentes classes *Python* utilisées dans notre projet pour atteindre nos objectifs.

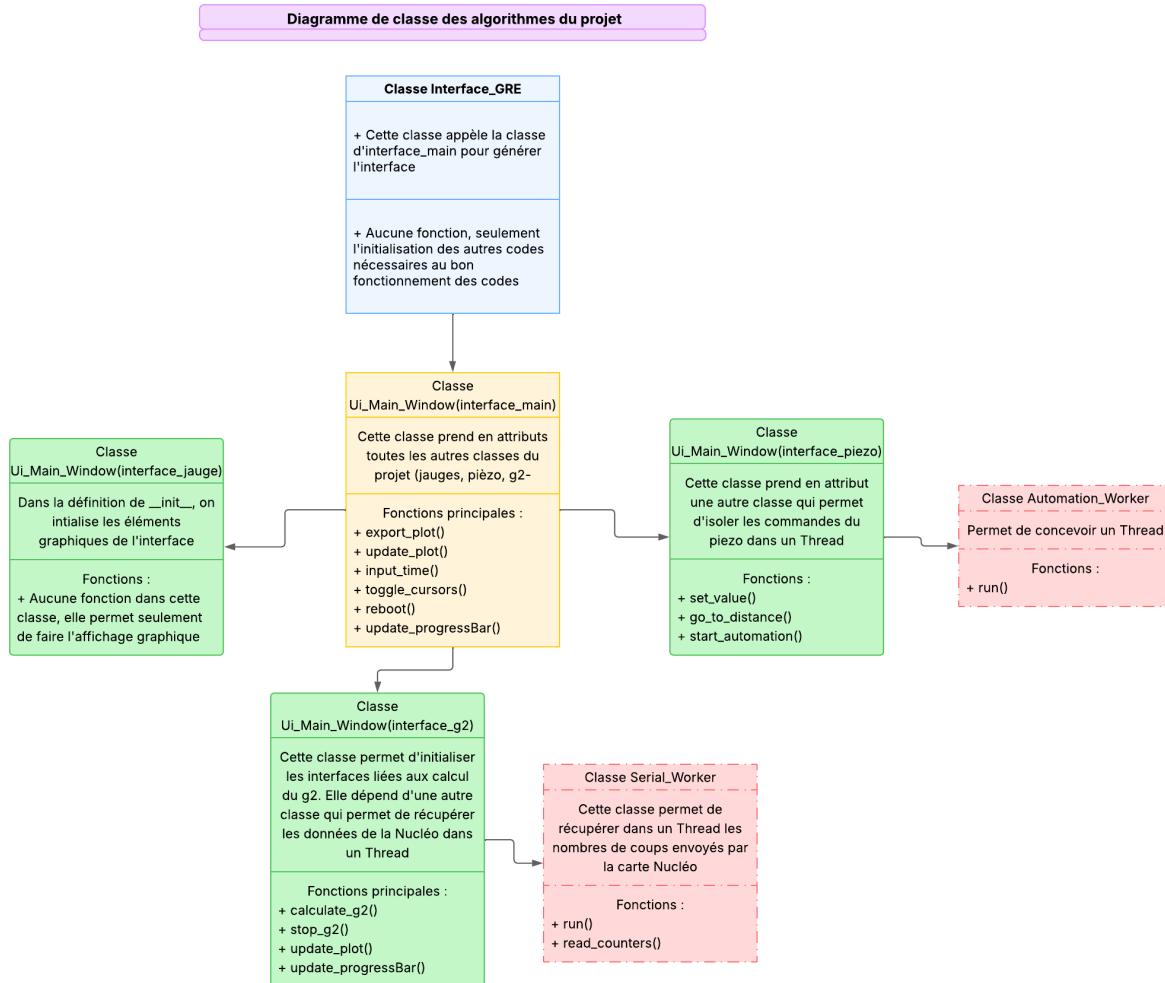


FIGURE 2 – Diagramme de classes des codes

5.1 L'interfaçage sous Python

Au cours de ce projet, nous avons du réaliser une interface sous Python. Pour ce faire, nous avons utilisé le module *PyQT5* ainsi que des commandes *CSS*.

5.1.1 Objectifs

Les objectifs de l'interface sont les suivants :

- Être facile d'utilisation
- Être compact et regrouper tous les éléments nécessaires à l'expérience
- Pouvoir gérer plusieurs fenêtres en même temps
- Gérer une acquisition en temps réel

5.1.2 Solutions

Les solutions mises en œuvre pour atteindre les objectifs sont les suivantes :

- Être facile d'utilisation : utilisation de la bibliothèque python intégré pour profiter des widgets déjà codés (boutons, slider, progressbar, graphiques, toolbar, checkbox,...)

- Être compact et regrouper tous les éléments nécessaires à l'expérience : utilisation d'un logiciel de conception d'interface *PyQT* (*Qt Designer*) pour faciliter la mise en place des widgets. Développement de fonction adaptée à chaque fonctionnalités souhaitées.
- Pouvoir gérer plusieurs fenêtres en même temps : Utilisation des commandes intégrées au module *PyQT*.
- Gérer une acquisition en temps réel : Utilisation de la méthode des Threads permettant d'isoler certaines classes ou fonctions afin d'éviter que certaines commandes bloquent l'actualisation de l'interface.

5.1.3 Tests de validation réalisés et résultats obtenus

Les images ci-après présentent nos résultats obtenus durant la présentation finale. Malheureusement, les images ne peuvent retranscrire l'acquisition en temps réel, toutefois cela a bien été observé en séance.

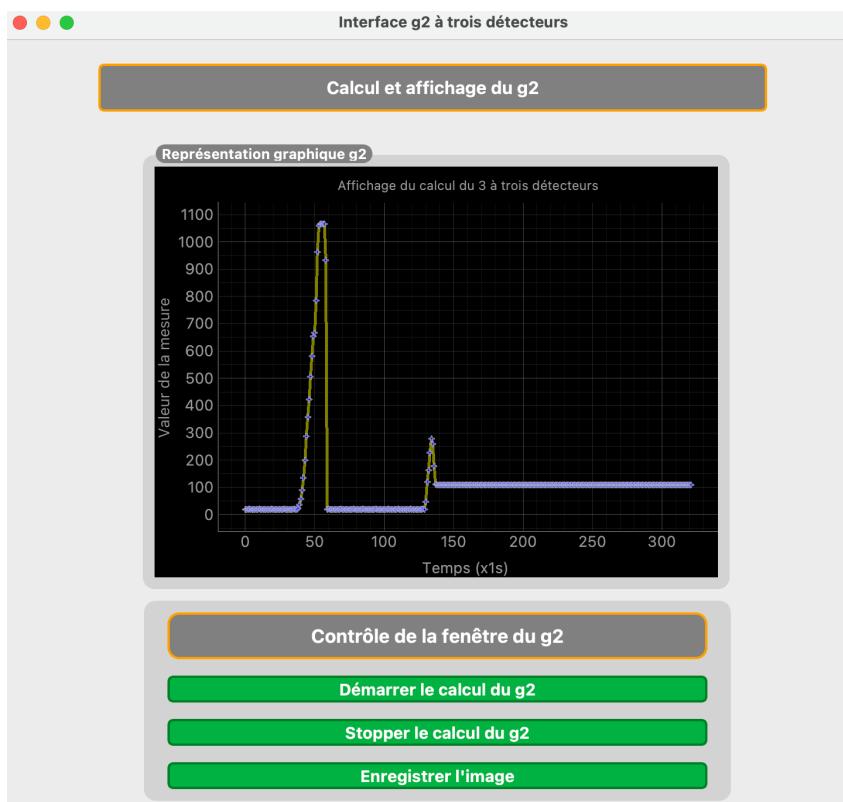


FIGURE 3 – Acquisition du $g^{(2)}$ en temps réel

5.2 Le calcul de la fonction de corrélation numérique

Le calcul de la fonction de corrélation numérique $g^{(2)}(\tau)$ est au cœur du projet DEPhI GRE, elle permet de caractériser la nature statistique de la lumière. En effet, cette fonction permet de déterminer le degré de corrélation des photons reçus dans chaque voie (mode), observée à deux instants séparés d'une durée τ . En pratique, cela consiste à comparer le nombre de photons détectés dans chaque voie.

Ainsi, Si la distribution temporelle des photons dans chaque mode est totalement aléatoire et indépendante, alors il n'y a pas de corrélation, ce qui correspond à un degré de corrélation égal à 1. À l'inverse, si les distributions temporelles des photons sont corrélées, c'est-à-dire qu'il y a un photon simultanément dans chaque mode, alors la fonction $g^{(2)}(0)$ prend une valeur supérieure à 1, ce qui reflète directement le degré de corrélation. Enfin, si la distribution temporelle est telle qu'il n'existe aucune corrélation entre les deux modes, alors $g^{(2)}(0) = 0$, indiquant qu'il n'y a qu'un seul photon parmi les deux modes.

5.2.1 Objectifs

L'objectif est de calculer la fonction de corrélation numérique dans deux cas. Le premier cas est le cas où le montage fonctionne avec deux détecteurs et le deuxième cas correspond au système fonctionnant avec trois détecteurs. Au final, il faut que le programme puisse calculer une fonction de corrélation numérique à deux et trois détecteurs.

5.2.2 Solution

Notation

T, correspond au photon transmis par un cube séparateur.

R, correspond au photon réfléchi.

G, correspond au photon qui annonce l'arrivée de son jumeau.

Cas à deux détecteurs

La fonction de corrélation numérique à deux détecteurs est $g^{(2)}(t) = \frac{N_{TR}(t)}{N_T(t)N_R(t)}$. Avec $N_R(t)$ et $N_T(t)$ le nombre de coups reçus sur chaque voie, directement reçus par la Nucléo et $N_{TR}(t)$ le nombre de coïncidences entre les signaux réfléchis et transmis par le cube séparateur de faisceau. Il faut faire attention à ce que le temps aveugle de la diode à avalanche soit supérieur à la durée de la fenêtre de coïncidence, sinon on peut se trouver dans le cas où on compte 2 coïncidences alors qu'il n'y a qu'un seul photon sur la voie T (ou R).

Cas à trois détecteurs

La fonction de corrélation numérique à trois détecteurs est $g^{(2)}(t) = \frac{N_{GTR}(t)N_G(t)}{N_{GT}(t)N_{GR}(t)}$ en conservant les mêmes notations qu'au dessus. En particulier, N_{GTR} correspond au nombre de coïncidences entre les trois voies.

Il nous reste à implémenter ces calculs en Python.

5.2.3 Tests de validation réalisés et résultats obtenus

La mise en situation réelle n'est pas possible car une partie de la carte d'acquisition n'est pas encore prête. Donc pour valider notre programme et nos calculs de $g^{(2)}$, on a simulé les signaux que la carte d'acquisition enverraient à la Nucléo à l'aide de pulse envoyés par des GBF dont nous comptions les fronts montants. Le résultat obtenu n'est pas absurde avec le résultat attendu. En effet les signaux reçus sont totalement indépendant et donc le $g^{(2)}$ attendu doit être constant (voir FIGURE 3)

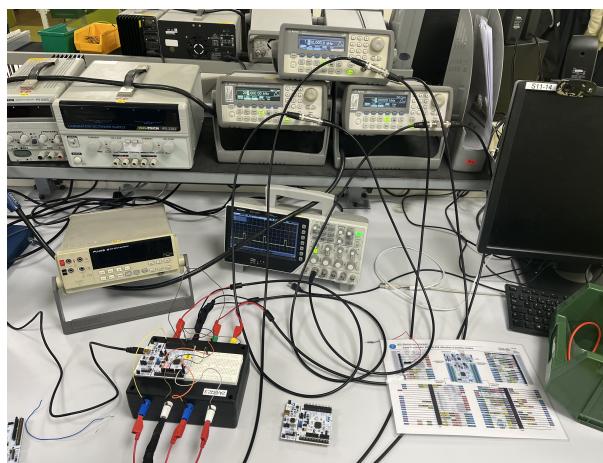


FIGURE 4 – Montage de test

Toutefois, le projet DEPhI associé n'est pas finalisé. Par conséquent, nous n'avons pas pu effectuer de tests avec des signaux issues de conditions propres à l'expérience.

5.3 Le contrôle de la cale piézo-électrique

5.3.1 Objectifs

L'un des objectifs du projet Déphi est d'obtenir des interférences à l'aide d'un interféromètre de Michelson. L'un des miroirs de cet interféromètre est translaté par une cale piézo-électrique que nous souhaitons pouvoir manipuler directement depuis l'interface pour pouvoir faire défiler les franges d'interférences.

5.3.2 Solution

Pour diriger la cale piezo-électrique thorlabs fournit un logiciel nommé Kynesis. thorlabs fournit aussi un Git comprenant différents codes dans différents langages pour contrôler ces cales. Nous nous sommes donc inspiré **de ce code** que nous avons implémenté dans notre interface. Nous avons créé 3 commandes principales : la première permet de déplacer la le miroir avec une précision de $0,01\mu m$, la seconde permet de revenir à la position initiale et la troisième permet de balayer toute la plage de déplacement de la cale (entre 0 et $20 \mu m$) avec un pas que nous avons pour l'instant fixé à $1\mu m$.

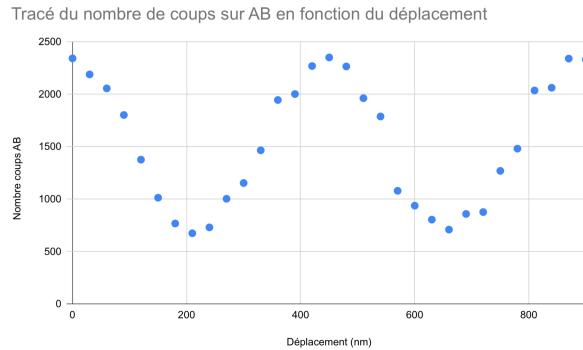
Voici à quoi ressemble la fenêtre de commande qui s'ouvre quand on clique sur "piezo control" :



FIGURE 5 – Fenêtre piezo control

5.3.3 Tests de validation réalisés et résultats obtenus

Nous avons pu tester l'interface directement sur le système optique du projet DEPhI. les trois commandent fonctionnent et le miroir se déplace bien selon les consignes données. Lors des matinées DEPhI nous avons même pu tester cette partie de l'interface sur le montage actif et voici le résultat obtenu :



Comme vous pouvez le voir sur cette courbe on obtient une sinusoïde caractéristique d'interférences, sa particularité est qu'elle est obtenue à partir d'interférences à 1 photon!

6 Notice d'utilisation

Dans cette partie, nous détaillons une brève notice d'utilisation du code. Pour faire fonctionner l'interface, il est nécessaire de placer les différents codes présents dans le fichier .zip dans le même dossier. Ensuite, il suffit d'exécuter le code *interface_GRA* qui regroupe toutes les commandes nécessaires à l'initialisation de tous les autres codes. Il faut également faire attention à bien renseigner la liaison série utilisée. Ensuite, une interface s'ouvre et dès lors il est possible d'avoir accès à toutes les fonctionnalités de l'interface. Pour ouvrir l'interface permettant d'afficher le $g^{(2)}$, il suffit de cliquer sur l'icône dans la barre de tâches à gauche de l'interface. L'interface a été codée de manière à ce que l'utilisation soit facile et logique. Il est possible d'ouvrir plusieurs fenêtres en même temps. Ci-après, une image de l'interface en cours d'utilisation.



FIGURE 6 – Interface en cours d'utilisation

7 Bilan du projet

7.1 Partie technique

Le diagramme ci-après présente comment nous avons réparti les différentes tâches sur l'ensemble des séances.

PROTIS IHM

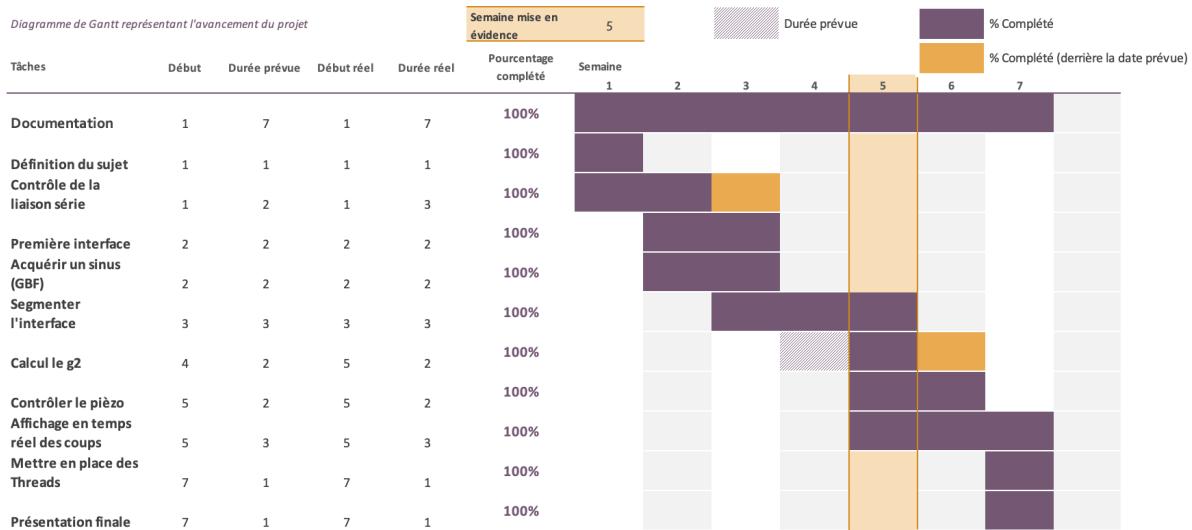


FIGURE 7 – Diagramme de Gantt du projet IHM

Nous faisons un bilan des performances :

- L'interface fonctionne très bien pour des temps d'acquisition de l'ordre de la miliseconde. Toutefois ici ce n'est pas limitant car la carte Nucléo nous envoie des données toutes les secondes.
- L'interface est robuste et peut gérer plusieurs acquisitions en même temps.
- La carte Nucléo peut traiter au maximum 1M coups/s sur les 3 voies : élément limitant.

Nous proposons quelques pistes d'amélioration :

- Une nouvelle carte Nucléo pouvant traiter jusqu'à 600 000 coups/s par voie.
- Segmenter encore davantage le code pour le rendre plus modulable.

7.2 Travail en équipe

Pour travailler efficacement en équipe, nous avons partagé nos ressources à l'aide de Google Drive au sein d'un dossier lui-même présent dans le Drive du projet DEPhi du groupe GRA. Cela nous a permis d'accéder à leurs ressources théoriques et ainsi d'avoir une meilleure compréhension du projet, avec notamment la fonction d'autocorrélation.

Ensuite, nous avions des difficultés à segmenter et distribuer les tâches de manière équitable. En effet, on ne pouvait pas tous travailler sur un même code. Il y avait donc une hétérogénéité du travail. Néanmoins, il y avait une très bonne efficacité une fois les informations bien comprises par tous et les tâches annexes mieux réparties, comme la compréhension du $g^{(2)}$ ou le contrôle des piézo-électrique qui a nécessité une réflexion supplémentaire.



FIGURE 8 – Photo d'une partie du groupe au travail

8 Conclusion

Pour conclure, ce projet fut riche en apprentissage, aussi sur la programmation que sur l'électronique. Nous avons également pu mettre en application nos connaissances théoriques de physique quantique pour faciliter la compréhension du problème. L'interface développée en Python, utilisant le module PyQt5, a permis de créer une solution modulable et facile d'utilisation, répondant aux exigences du cahier des charges.

Enfin, ce projet a non seulement renforcé nos compétences techniques, mais a également développé notre capacité à travailler en équipe et à gérer des projets complexes. Les pistes d'amélioration identifiées, telles que l'utilisation d'une nouvelle carte Nucléo et une segmentation plus fine du code, ouvrent la voie à des développements futurs pour optimiser encore davantage l'interface et ses fonctionnalités.