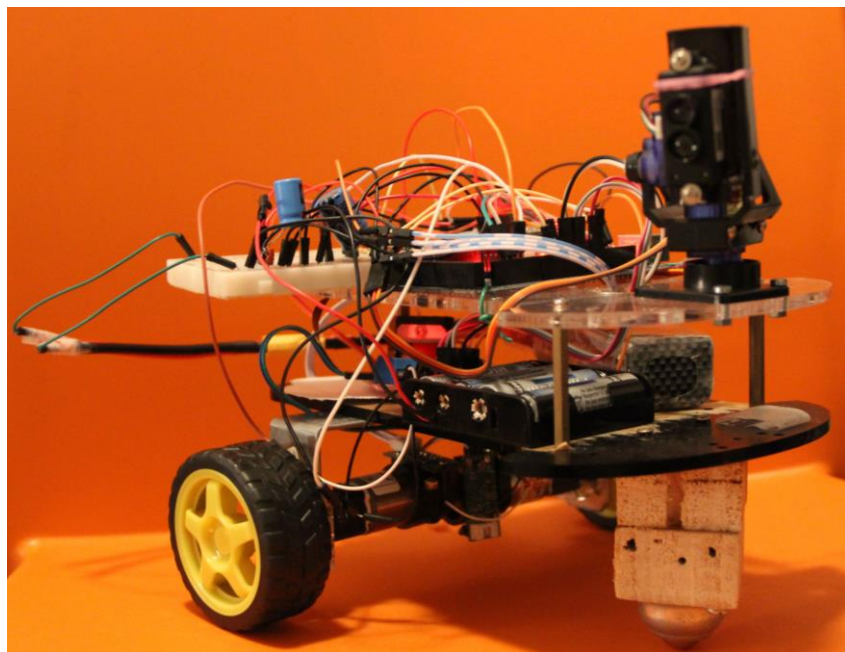




**POLYTECH<sup>®</sup>**  
NICE-SOPHIA



# Slam Arduino Bot

RAPPORT DE PROJET

De La Fuente Léo, Fonteneau Félix | Electronique avec Arduino | 01/06/2018

## Sommaire

---

- 1- Présentation du projet et de ses problématiques.
- 2- Robot
- 3- Lidar
- 4- Télécommunication
- 5- Informatique et traitement des données
- 6- Regroupement des différents modules
- 7- Conclusion

# 1-Présentation du projet et de ses problématiques

Slam Arduino Bot est un petit véhicule permettant de cartographier son environnement sans données GPS.

Le robot est contrôlé par son utilisateur, il se déplace, mesure des distances et transmet les données. Du côté de l'ordinateur, une application traite les données reçues, trace la carte en 2D en fonction des données et envoie les ordres au robot.

Tout d'abord, le robot s'initialise puis se connecte à l'application de l'utilisateur par wifi. Puis le robot commence à réaliser des mesures afin de scruter son environnement. Il transmet les données à l'application qui les traite, et affiche la carte réalisée. L'application attend ensuite les ordres de l'utilisateur concernant ses déplacements qui vont être envoyés au robot. Après avoir effectué les déplacements nécessaires, le robot va réaliser de nouvelles mesures et ainsi de suite.

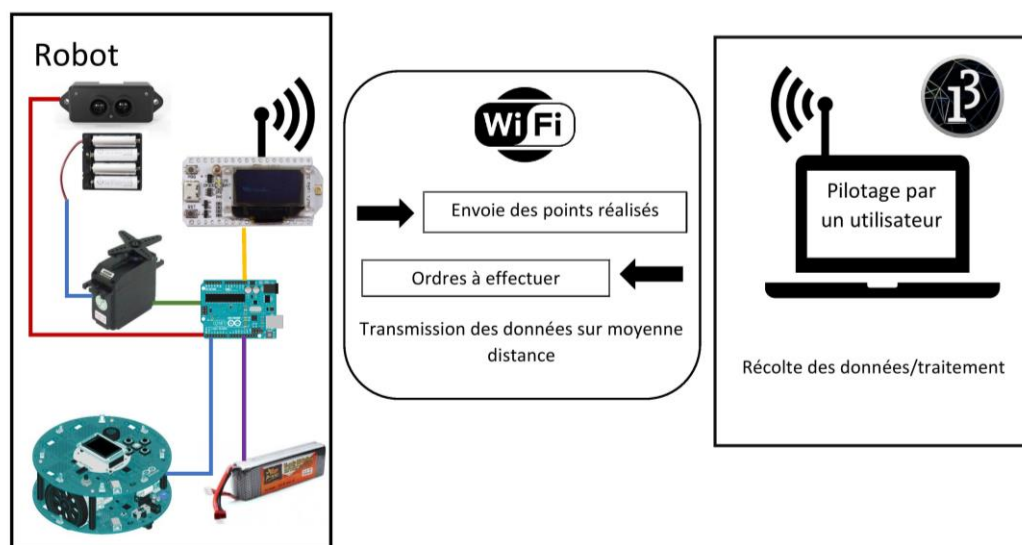
Nous distinguerons par la suite quatre grandes parties constituant ce projet, trois sont principalement matérielles (hardware):

- Le robot
- Le lidar
- Le module de communication

Et une partie purement informatique (software) se trouvant sur l'ordinateur de l'utilisateur.

Ces différents modules sont détaillés dans les pages suivantes.

Le problème théorique le plus important face à ce projet est l'odométrie. En effet, il est très difficile de situer exactement le robot dans son environnement. Nous y reviendrons en détail dans la partie traitement de l'information.



*Schéma du projet final*

## 2-Robot

---

### A) PRESENTATION DU MODULE

Le robot est une partie du projet qui met en relation trois domaines : la mécanique, l'électronique, et l'informatique. Sa principale utilité est de pouvoir exécuter une action de déplacement de manière précise. Cette action est récupérée depuis un port série sur une Arduino.

Les déplacements que peut faire le robot sont :

- Avancer/reculer en ligne droite d'une distance donnée
- Tourner dans le sens horaire/anti-horaire d'un angle donné

Ces fonctions élémentaires permettent de pouvoir faire se déplacer le robot en n'importe quel point de l'espace, de manière précise.

#### 1 Matériel :

- Arduino :

Le robot embarque une carte Arduino Méga permettant de recevoir les ordres et de commander les moteurs. L'avantage de ce modèle d'Arduino est de pouvoir connecter plusieurs ports séries. Ceux-ci nous seront utiles pour pouvoir commander le lidar et transmettre les informations.

- Moteurs :

Deux moteurs à roues codeuses permettent de réaliser les déplacements. Ces moteurs sont composés de lasers optiques qui envoient des impulsions lorsque le moteur réalise un certain nombre de tour. L'avantage de ces moteurs est de pouvoir récupérer des informations sur la rotation des moteurs par l'Arduino.

- Pont en H L298N :

Ce circuit permet de faire le lien entre l'Arduino, les moteurs et l'alimentation des moteurs. Le L298N permet ainsi de récupérer grâce à 3 fils les ordres de l'Arduino. Les deux premiers servent à commander le sens de rotation du moteur, en y appliquant des états haut ou bas (0 ou 5V), et le dernier permet de donner la vitesse demandé grâce à un signal PWM. Ce circuit est branché sur du 12V pour pouvoir ensuite alimenter les moteurs. Il renvoi finalement sur les deux fils du moteur le bon signal en 12V pour pouvoir faire tourner dans le sens horaire/anti-horaire le moteur, à la vitesse demandée.

- Roues :

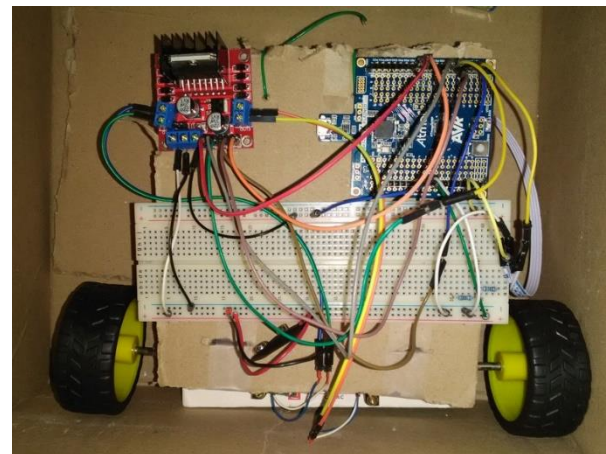
Le robot ne possède que trois roues. Les deux premières sont positionnées sur les moteurs, et la dernière est une roue folle qui est à l'avant du robot, et qui permet de faire rouler le robot dans toutes les directions. La rotation du robot s'effectue juste avec les deux moteurs : On actionne le premier moteur dans un sens, et le second dans le sens inverse.

## B) TRAVAIL EFFECTUE

### 2 Premiers pas

Notre premier but était de réaliser un robot sur un bout de carton pour bien comprendre comment actionner les moteurs avec l'Arduino.

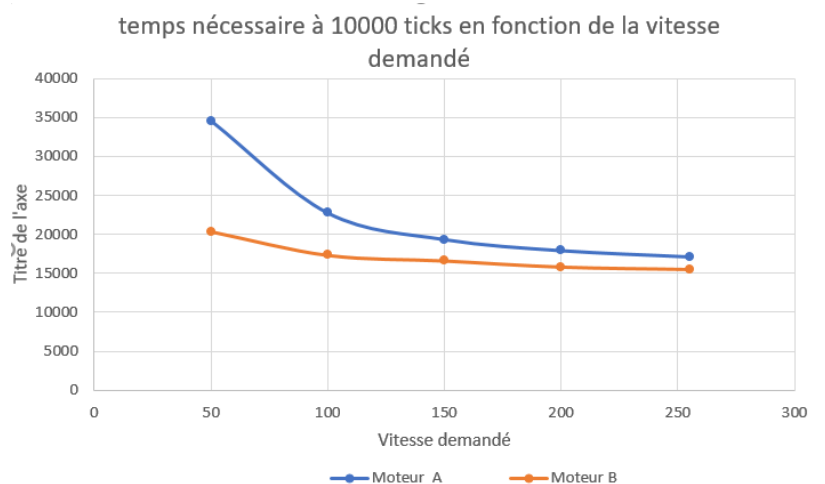
Nous avons tout d'abord mis du temps à bien câbler et comprendre le fonctionnement du pont en H, car plusieurs câbles défectueux sont venus nous rendre la tâche plus difficile. Une fois les câbles changés, nous avons pu faire les premiers tests du robot.



Nous avons ensuite changé de support en *Premier support du robot en carton* récupérant un châssis en plexiglass. Nous avons dû pour cela réaliser des pièces à la découpeuse laser pour adapter la fixation des moteurs. En réalisant les premiers tests sur le nouveau support nous nous sommes vite rendu compte que les moteurs ne tournent pas à la même vitesse, bien que le même ordre leur soit envoyé. Cela a pour conséquence que le robot ne se déplace pas en ligne droite.

### 3 Coefficient de frottement

Nous nous sommes donc demandé si l'on pouvait trouver une relation linéaire entre la vitesse demandée, et la vitesse mesurée sur chaque moteur. Cela nous aurait permis d'appliquer un facteur sur la vitesse demandée sur un des moteurs, et ainsi avoir la même vitesse finale sur les deux moteurs. Nous avons donc réalisé un code (`calcul_frottement_moteurs`) permettant de calculer le temps nécessaire pour réaliser un certain nombre de tour du moteur. En exécutant ce code avec différentes vitesses demandées sur les deux moteurs, nous avons eu comme résultat les deux courbes ci-présentes.



Les courbes nous ont montré que le frottement n'était pas du tout linéaire. Une solution simple et efficace n'était donc pas envisageable en appliquant un coefficient de proportionnalité.

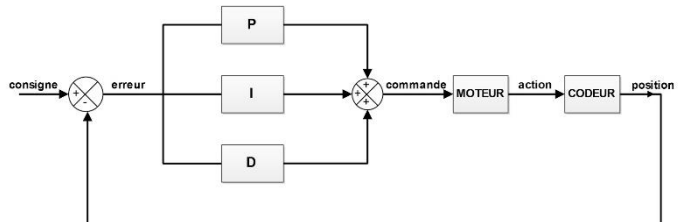
#### 4 Asservissement

Un asservissement en vitesse s'est donc imposé comme la solution à notre problème.

« Mais qu'est-ce qu'un asservissement en vitesse ??? »

L'asservissement en vitesse d'un moteur est une méthode permettant de réguler précisément la vitesse de rotation du moteur en fonctionnant en boucle. Cette boucle commence par envoyer un ordre au moteur, puis calcul la vitesse de rotation

du moteur grâce à la roue codeuse, et réajuste l'ordre donné pour que la vitesse du moteur soit au plus proche de la valeur attendue. Cet algorithme est exécuté indéfiniment pour réajuster en temps réel la valeur de la vitesse.



Source : [naytech.blogspot.com](http://naytech.blogspot.com)

La réalisation de cet asservissement qui s'exécute toutes les 10 milliseconde permet ainsi que les deux moteurs tournent à la même vitesse.

Notre robot peut grâce à cela se déplacer en ligne droite, et tourner sur lui-même.

#### 5 Calcul de distance

Une variable de comptage est incrémentée à chaque impulsion lu sur les roues codeuses. Celle-ci nous permet de réaliser l'asservissement en vitesse, mais nous permet aussi de connaître la distance effectuée par le robot. Le rapport de réduction étant de 1:80 et sachant que l'on récupère deux impulsion par tour, on sait que lorsque l'on reçoit une impulsion, la roue a parcouru 1/160ième de son périmètre. On peut donc en connaissant le périmètre de nos roues, déduire le nombre d'impulsion nécessaire pour que le robot parcoure une distance donnée. Comme les moteurs ne peuvent pas être arrêter et comportent une roue libre, nous avons fait attention à faire ralentir le robot lorsqu'il commence à se rapprocher de son point d'arrivée.

Pour ce qui est de la rotation, le calcul est réalisé en connaissant l'entraxe de nos roues. Les roues possédant un léger décalage d'alignement, nous avons été contraints d'appliquer une correction sur la valeur mesurée sur l'entraxe des roues. Celle-ci ne s'appliquant que pour la rotation dans le sens horaire, le robot ne peut finalement exécuter que des tours dans ce sens pour avoir une mesure précise de la rotation effectuée.

### C) PROBLEMES RENCONTRES

Nous avons grillé nos premières roues codeuses en appliquant une tension trop forte dessus car la data sheet des roues codeuses était dans un premier temps introuvable, et nous les avons donc branchés selon la data sheet d'une roue codeuse semblable. Nous avons été obligés d'en récupérer deux nouvelles.

### D) BILAN

Finalement le robot se déplace bien tout droit d'une distance donnée et réalise bien l'angle demandé en tournant sur lui-même. Cette rotation ne se fait que dans le sens horaire mais cela n'est pas très pénalisant. Le robot est autonome grâce à la batterie lipo 12V qu'il embarque qui alimente les moteurs, l'Arduino et le pont en H.

## 3-Le Lidar

---

### A) PRESENTATION DU MODULE

Le module lidar est l'élément permettant de réaliser les mesures de distance via envoi d'ondes électromagnétiques.

Il s'alimente en 5V et est monté sur deux servomoteurs. Ces deux servomoteurs permettent d'effectuer des rotations sur un axe vertical et horizontal afin de pouvoir faire un nuage de points. Chaque point est associé à une norme et un angle selon les servomoteurs.

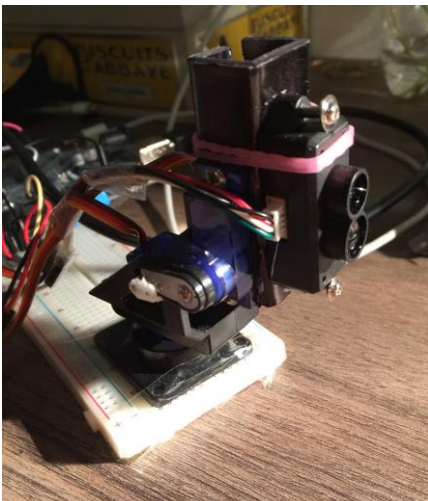
Le lidar va effectuer des mesures avec une fréquence maximale de 10 Hz dans notre cas. Même si les mesures peuvent être plus rapides (max 100Hz), la précision sera très réduite.

### B) TRAVAIL EFFECTUE

Pour la construction de ce module, nous avons tout d'abord utilisé un télémètre laser monté sur un seul servomoteur. Ce télémètre était très précis et pouvait effectuer des mesures allant jusqu'à 40m. Le problème est que la fréquence des mesures était très faible : environ une mesure toute les 0,3 secondes. De plus il ne fallait pas que le module bouge lors de l'acquisition.



Télémètre laser de marque HIREED  
Utilisé au début du projet



Nouveau lidar

Après un court-circuit de ce laser qui le rendu non opérationnel, nous nous sommes tournés vers un lidar. Celui-ci a la particularité d'effectuer des mesures en continu sans attendre d'ordre quelconque et à une fréquence élevée (voir présentation du module).

Nous avons aussi fait un code sur Processing afin de traiter les données recueillies.



### C) PROBLEMES RENCONTRES

Premièrement, nous avons eu un problème de court-circuit avec le laser comme précédemment cité.

Puis un problème récurrent et non résolu concernant le lidar définitif. En effet, le lidar TF-mini que nous utilisons, est de basse qualité. Quand il mesure en continu, il va, par moment s'interrompre et afficher des valeurs constantes aberrantes. L'arrêt de ces mesures ne peut être rétabli, car le module ne reçoit aucune information.

Nous avons pensé au début que son alimentation n'était pas suffisante, c'est-à-dire que l'élément demandait trop de courant par saccade. Nous avons donc changé son alimentation et mis en parallèle une grosse capacité permettant de pallier le manque d'intensité, mais cela n'a pas résolu le problème.

Une tentative de le débrancher et rebrancher automatiquement fut envisagée, mais inefficace. Nous avons donc un module lidar, affichant de temps en temps des mesures erronées, sans que nous sachions pourquoi.

### D) BILAN

Le lidar est un module qui remplit à moitié sa tâche, il fonctionne très bien en temps normal, mais il arrive indépendamment de l'état du robot, que le laser plante et affiche des données erronées.

Il a cependant permis de faire progresser grandement le projet, et au long terme, le remplacer par un lidar plus fiable est nécessaire.

## 4-Télécommunication

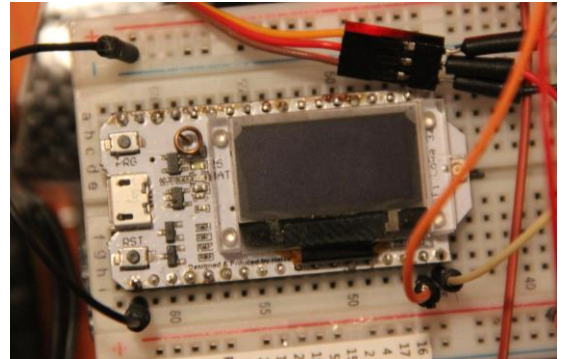
---

### A) PRESENTATION DU MODULE

Le module de télécommunication permet de transmettre les mesures effectuées par le robot à l'utilisateur, mais aussi les ordres de l'utilisateur au robot.

Pour cela nous utilisons la carte esp32 wifi-kit de Heltec pour sa souplesse entre Lora et wifi.

Afin de connecter l'application à l'ordinateur (processing) et l'esp, il faut que les deux éléments se trouvent sur le même réseau. Nous avons donc utilisé un réseau wifi fermé tel qu'une box ou un téléphone en partage de connexion.



*Esp32 wifi-kit*

L'application (processing) agit comme serveur, elle va attendre la connexion de l'esp32 avant d'établir la communication. L'esp joue donc le rôle de client. Dès que la première connexion est établie, les deux éléments communiquent par question/réponse afin de transmettre les données.

### B) TRAVAIL EFFECTUE

Tout d'abord, avant de programmer sur la carte, il faut mettre à jour l'IDE Arduino afin de pouvoir téléverser sur l'esp. Cette mise à jour ne marche pas tout le temps et est plutôt laborieuse.

La communication par wifi entre la carte et l'ordinateur demande des connaissances et notions de base avant de pouvoir l'utiliser. De plus toutes les méthodes et protocoles ne sont pas forcément évidents, ils demandent un certain investissement et temps de manipulation avant d'être compris.

Cependant Heltec nous offre de nombreux exemples de code réalisant différentes connexion ou fonctionnalités, cela permet d'y voir plus claire.

### C) BILAN

Cet élément a été plutôt chronophage et difficile à appréhender et nous a demandé beaucoup d'investissement. Mais après beaucoup de temps passé, le module marche très bien. La connexion entre l'esp et l'ordinateur est bonne et fiable.

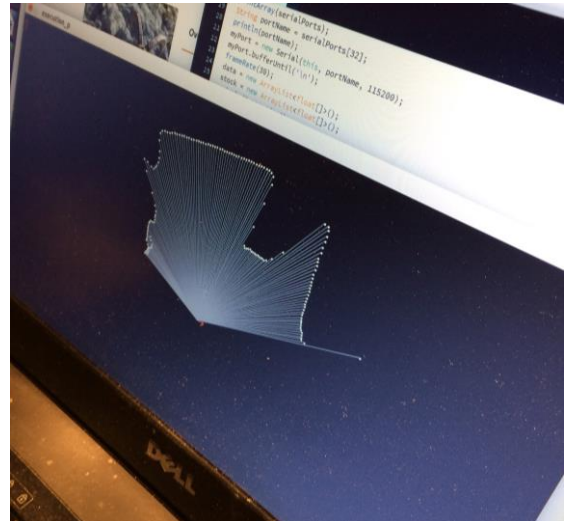
## 5-Informatique et Traitement des Données

### A) PRESENTATION DU MODULE

La partie d'implémentation d'une application sur l'ordinateur a été un point essentiel dans le développement de notre projet.

Elle se compose en trois sous-parties.

- La première est la programmation d'un serveur sous processing afin de pouvoir récupérer les données.
- La seconde est la réception, traitement et stockage des données dans une base de données.
- La dernière concerne l'interface graphique traçant la carte et les interactions avec l'utilisateur.

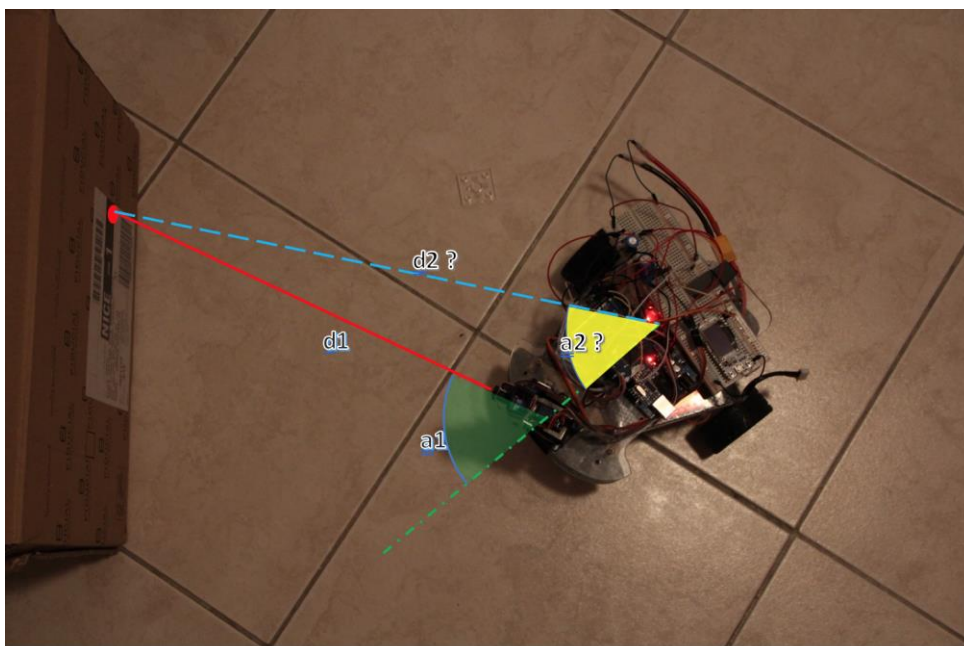


*Tracé de carte avec les données du lidar*

### B) TRAVAIL EFFECTUE

Tout d'abord, la première chose réalisée fut le traitement des données en parallèle de la construction du module laser.

Cette manipulation ne fut pas si facile. En effet à chaque vecteur (distance ; angle) il faut associer un trait et un point sur la carte en fonction de paramètres trigonométriques afin de se ramener au centre de rotation.



Après avoir traité les données, la réalisation de la carte se fait grâce à l'interface de base de processing. Des fonctionnalités supplémentaires telles que le zoom ou le fait de se déplacer dans la carte ont été ajoutés.

Puis en parallèle de la partie Télécommunication, nous avons développé le serveur wifi sous processing afin de récupérer les données de l'esp32. Le fonctionnement du serveur se base sous la forme suivante : acceptation de la communication => réception des données => réponse du serveur (ordre à envoyer).

### C) BILAN

Le bilan de ce module est plutôt positif, l'application sert de passerelle entre l'utilisateur et le robot, afin de mieux cartographier son environnement.

## 6-Regroupement des différents modules

---

Les différents modules étant réalisés et fonctionnant de manière indépendante, il a fallu les combiner pour réaliser le projet final.

### A) COMBINAISON LIDAR ET ROBOT

Dans un premier temps nous avons assemblé les codes du lidar et du robot pour que le robot puisse réaliser les mesures au bon moment. Cela a pu se faire en réalisant une nouvelle fonction dédiée spécialement à la prise de mesure du laser, et en faisant attention à réinitialiser les mesures d'asservissement à la fin de ces données. En effet, le programme d'asservissement qui s'exécute toutes les 10 millisecondes et faussé après un temps durant lequel le robot reste statique. On met donc à zéro tous les compteurs avant d'exécuter une nouvelle action.

### B) COMBINAISON ESP<sub>32</sub> ET ARDUINO

La connexion entre l'Arduino et l'esp se fait par un port série. L'esp ne recevant que des informations en 3.3V, nous avons réalisé un pont diviseur de tension entre la transmission de l'Arduino et la réception de l'esp.

C'est à ce moment-là que les problèmes sont apparus...

La connexion en port série entre l'Arduino et l'esp est devenu assez aléatoire. Parfois les valeurs que l'on voulait échanger se transmettaient bien, avec de temps en temps de mauvais caractère qui apparaissaient. D'autre fois nous ne recevions plus rien sur aucune des cartes alors qu'aucun branchement n'a été changé et que les codes restaient strictement identiques. Et enfin nous avions de temps en temps des caractères incompréhensibles qui étaient échangés entre les cartes alors que nous ne souhaitions pas envoyer de données.

Nous avons pu ainsi avoir quelques données échangées entre le robot et processing, mais sur la centaine de points envoyé par l'Arduino, une dizaine seulement étaient affichés sur l'écran de processing.

Nous n'avons malheureusement pas pu résoudre ce problème qui nous rend impossible le fait de terminer le projet entièrement.

### C) BILAN DU REGROUPEMENT DES MODULES

La tâche consistant à regrouper les modules nous paraissait plutôt aisé, et à finalement était l'un des plus gros problèmes. Celui-ci qui nous a pris beaucoup de temps et a finis par bloquer la réalisation final du projet.

## 7-Conclusion

---

Le projet consistait à réaliser un robot dirigé par un utilisateur pour scanner une pièce et en faire une carte 2D sur un ordinateur. Nous l'avions découpé en 3 modules différents que nous sommes arrivés à faire fonctionner indépendamment. Le regroupement entre le robot et l'esp permettant la connexion wifi n'as finalement pas pu être réalisé à cause de nombreux problèmes que nous avons rencontré et que nous ne sommes arrivé à résoudre. Le projet ne réalise donc finalement pas la tâche demandée, mais il n'en est pas très loin.

Cet expérience nous a permis de comprendre les difficultés que l'on peut avoir lors de la réalisation d'un projet, et qu'il est difficile de prévoir la durée des tâches à effectuer pour avoir un planning juste. Il nous a aussi permis de nous rapprocher encore plus de l'environnement Arduino, tout en nous demandant des connaissances dans des matières et des domaines que l'on ne connaissait pas (processing, mécanique). Cela nous a aussi permis d'apprendre à rechercher par nous-même des solutions aux problèmes que l'on peut rencontrer.

La taille conséquente du projet nous demandé de faire attention à bien avancer pas à pas et à ne pas négliger certaines parties. De plus des capacités d'abstraction et de prises de recules ont été nécessaire afin de bien organisé les différents interactions entre toutes les partie du projet.