Année scolaire 2022-2023

Étudiants : GISSINGER Esteban, VERMOT-DESROCHES Matthias

Rapport de projet final

Le PoLIDAR

I. Introduction

La cartographie est un processus permettant de créer une carte du relief d'une zone à partir de coordonnées récupérées après un scan de cette zone. Cette tâche peut être compliquée pour des humains, le relief peut contenir des falaises, des rochers, des collines ou des bosses par exemple ou bien la zone peut être une zone de danger à la suite d'une catastrophe naturelle. Pour faciliter ce processus, de nombreux robots ont été conçus pour aider en fonction de la zone à cartographier (sur terre, dans les airs, sous l'eau ou encore sur Mars avec le rover martien), mais la plupart demande encore une présence humaine afin d'assurer leur bon fonctionnement.

Notre objectif est donc de produire un robot cartographe ne nécessitant pas de présence humaine pour fonctionner et ce, grâce à un capteur laser Lidar. Ce robot devra être tout terrains sur terre. Il existe deux types de robot terrestre, celui avec des roues et celui avec des chenilles. Nous avons choisi d'utiliser six roues pour les déplacements en raison de leur efficacité.

Le robot devra donc pouvoir cartographier une zone donnée et sauvegarder la carte et le nuage de points 3D créés afin de les envoyer à l'utilisateur ou au moins permettre à l'utilisateur de les récupérer plus tard en se connectant directement au robot avec un ordinateur. Comme il devra être autonome dans ses déplacements, nous utiliserons donc les coordonnées prises par le Lidar pour lui permettre de se localiser et donc de se diriger et d'éviter les obstacles sans assistance humaine. Nous avons également choisi d'utiliser des capteurs ultrasons en cas d'urgence où le Lidar n'aurait pas détecté un obstacle comme une vitre (cas où le laser passerait à travers) par exemple.

Nous allons vous présenter ce que nous avons réalisé, en commençant par la réalisation du châssis du robot et du lidar. Nous présenterons ensuite la programmation Arduino du lidar et des capteurs, puis la programmation avec le système d'exploitation ROS. Enfin, nous terminerons par expliquer l'algorithme de fonctionnement du robot avant d'exposer les problèmes rencontrés lors de ce projet et de conclure sur nos réalisations et améliorations possibles.

II. Le Châssis

Le châssis est considéré comme la pièce la plus importante de notre robot, puisqu'il supportera les roues, la batterie, les capteurs et microprocesseurs. Aussi, nous avons tout d'abord réalisé une série de croquis afin de choisir une forme. Notre choix s'est porté sur une boîte ouverte, avec les microprocesseurs et la batterie posés au milieu et les capteurs sur l'étage du haut. Puis, nous avons choisi une liaison roue-moteur constituée d'un moteur CC 12V, d'un servomoteur 5V, d'une roue et de pièces de support.

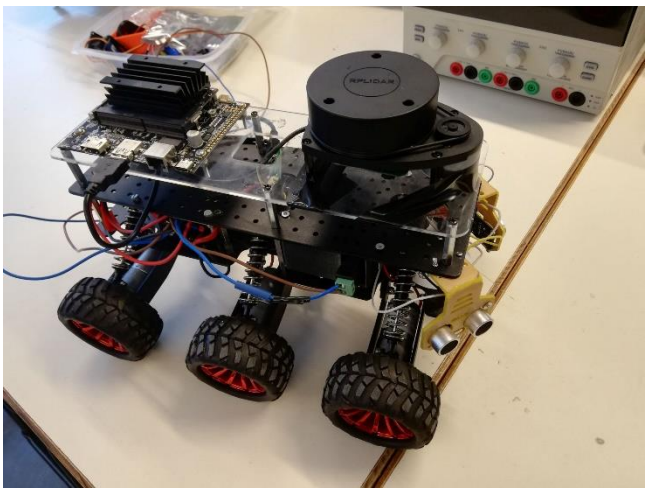
Nous avons ensuite réalisé un prototype du châssis et de la liaison avec des pièces en bois. Une fois réalisé, il nous suffisait de répéter la liaison roue-moteur avant d'assembler les pièces.

En raison d'un manque de temps, nous avons dû commander un châssis avec toutes les pièces (moteurs CC et roues inclus) directement. Cette configuration a amené à ne plus utiliser de servomoteurs.

Nous nous sommes ensuite occupés de la fixation des capteurs et microprocesseur sur la carte. Nous devions fixer deux capteurs ultrasons HC-SR02, un Lidar (Lidar 2D pour commencer avant de le remplacer par un 3D), une carte Nvidia jetson nano, le driver des moteurs et une carte Arduino Uno.

Manquant de place pour percer les trous de fixation pour la carte Nvidia et le Lidar, nous avons découpé un nouvel étage en PMMA. Faute de temps, nous avons laissé le Lidar 2D sur le robot.

Voici une photo du châssis complet :



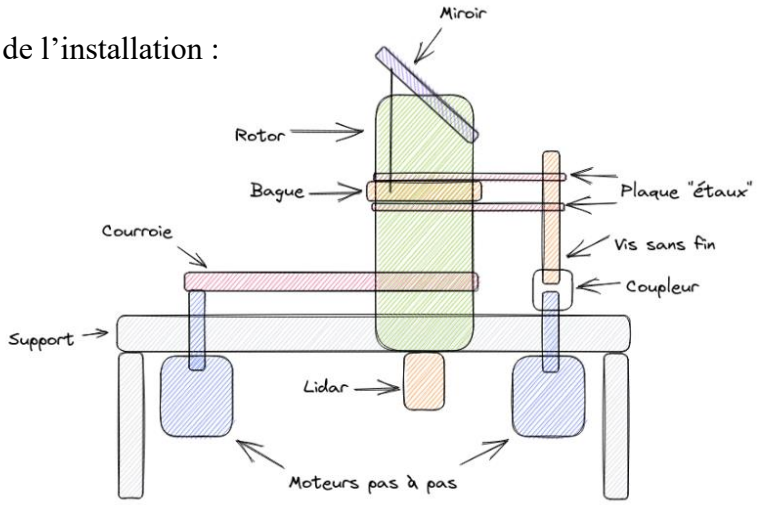
III. Programmation Arduino

En parallèle de la réalisation du châssis, nous avons également avancé la partie programmation du PoLIDAR : nous avons réalisé un algorithme permettant au robot de détecter les obstacles avec les capteurs ultrasons. Ce programme consiste à convertir la durée d'un aller-retour d'un ultrason en distance et à faire tourner le robot fonction de cette distance.

IV. Le fonctionnement du Lidar

Le lidar fonctionne grâce au garmin lidar-lite v3hp qui est fixé sur un support. Le laser est ensuite redirigé vers la direction souhaitée grâce à un miroir. Initialement, le système comprenait un moteur pas à pas pour faire tourner le miroir et deux servos-moteur pour son inclinaison. Cependant, les servo-moteurs ne permettaient pas de faire varier l'inclinaison correctement. La version finale utilise donc deux moteurs pas à pas. Le premier sert à faire tourner un rotor par l'intermédiaire d'une courroie. Le second sert à gérer l'inclinaison du miroir en modifiant la hauteur de la bague prise en étau par deux plaques fixées sur la vis sans fin.

Voici un schéma de l'installation :

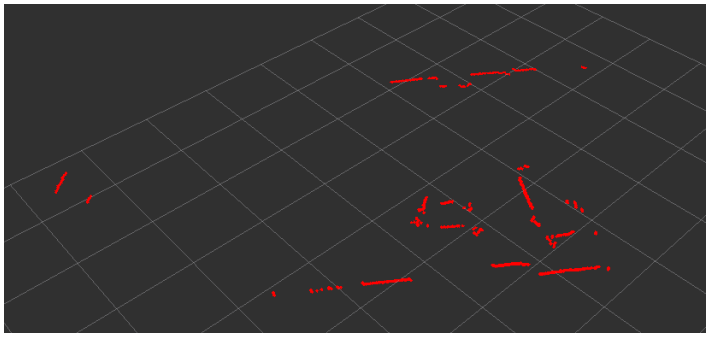


V. ROS

Pour la génération et la sauvegarde de la carte ainsi que la navigation autonome, nous avons choisi d'installer ROS sur la carte Nvidia. C'est un système d'exploitation qui permet de séparer un script ou programme en plusieurs nodes, afin de mieux gérer la mémoire, la fusion des données et le microprocesseur.

Après avoir fini l'installation et suivi les tutoriels pour débutants de ROS, nous avons utilisé la bibliothèque rplidar afin de générer une carte à partir des coordonnées relevées par le Lidar 2D.

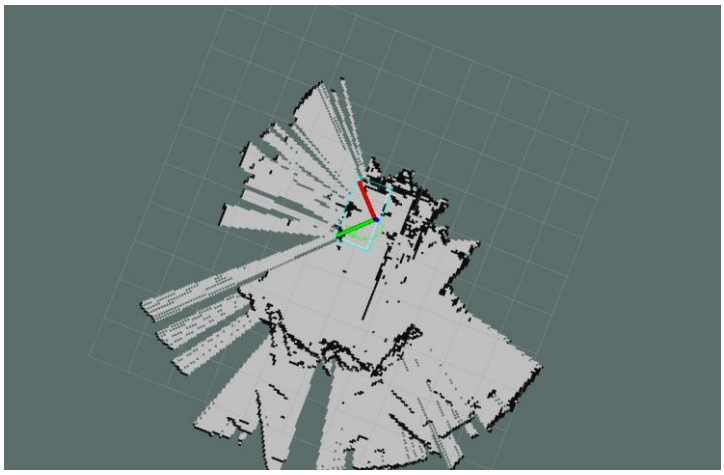
Voici un exemple de carte générée :



Vous pouvez remarquer que la carte générée n'est qu'un nuage de points en 2D et qu'elle n'indique rien de particulier sur le relief de la zone. Par ailleurs, si le Lidar est déplacé, la carte se met à jour et ne conserve pas le nuage de points précédent. Ce résultat n'étant pas utilisable pour la cartographie, nous avons décidé de l'améliorer. Nous avons choisi d'ajouter la bibliothèque hector_slam pour utiliser une partie d'algorithme SLAM (Simultaneous

Localization And Mapping), un algorithme permettant au robot de se localiser tout en cartographiant. Voici un exemple de carte générée avec SLAM :

Voici un exemple de carte générée avec SLAM :



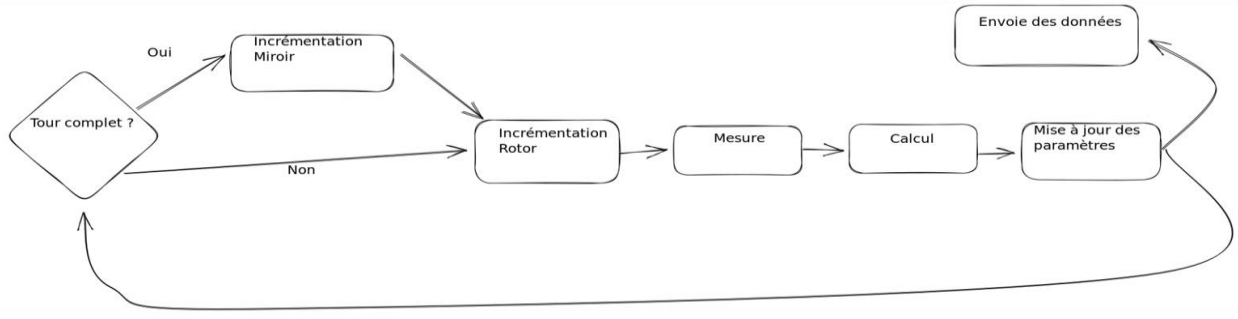
La carte est devenue un peu plus représentative du relief, le nuage de point a laissé place à une modélisation 2D. Chaque point noir correspond à un obstacle ou mur détecté et quand le Lidar est déplacé, la carte se met à jour en conservant les données précédentes. De plus, grâce au SLAM, si nous changeons notre Lidar 2D pour un 3D, la carte devrait, en théorie, afficher les reliefs de la carte.

Nous avons ensuite programmé un script puis un service permettant de sauvegarde la carte générée au format png toutes les dix minutes. Nous n'avons pas eu le temps ni de finaliser un script permettant la navigation autonome du robot ni d'essayer la génération de carte en fonctionnement avec le Lidar sur le robot.

VI. Algorithme de fonctionnement

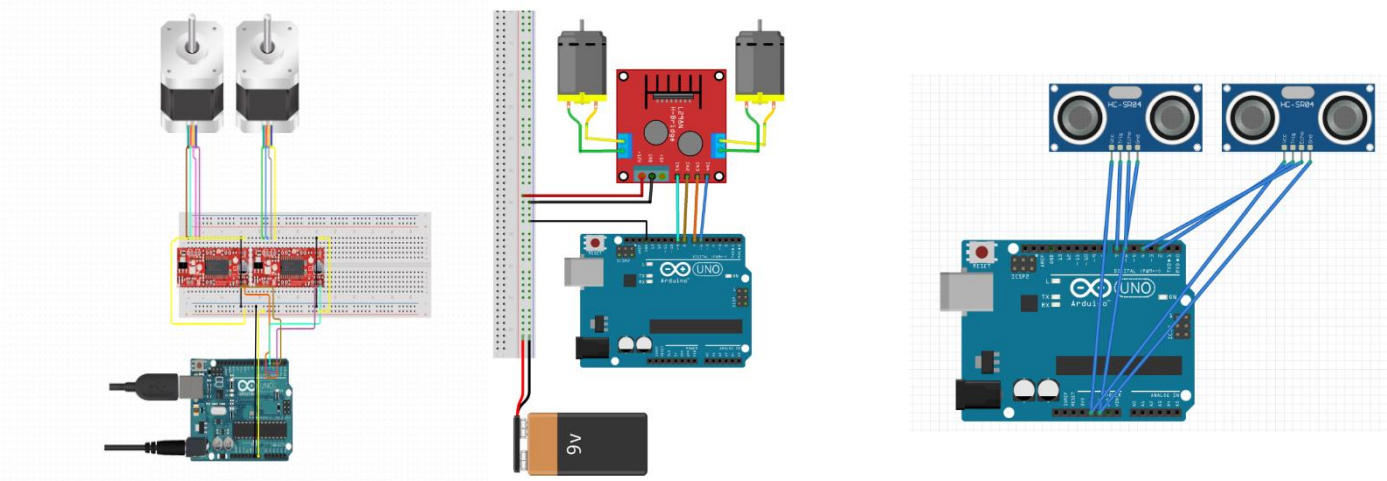
Nous allons maintenant expliquer ce que fera le robot en fonctionnement.
Une fois le robot alimenté et mis en route, il va commencer à rouler et le Lidar 2D à tourner. Pendant qu'il roule, les capteurs ultrasons émettront des ultrasons pour détecter de potentiels obstacles. Si un obstacle est détecté à gauche, la vitesse des moteurs de droite diminuera et celle des moteurs de gauche augmentera afin de permettre au robot de tourner à droite et donc d'éviter l'obstacle. Cela se passe de la même manière, mais en inversé, en cas d'un obstacle à droite. Si le robot fait face à un obstacle trop proche dans ces deux directions, donc un obstacle trop large, le robot fera marche arrière et tournera pour ne plus faire face à cet obstacle.

En parallèle à l'algorithme de détection d'obstacle, la carte Nvidia va générer une carte à partir des points relevés par le Lidar, cette carte se mettra à jour automatiquement lors des déplacements du robot. Ainsi, toutes les dix minutes, une carte au format png sera sauvegardée.
Voici l'algorithme de fonctionnement du Lidar :



VI. Schémas électriques

Schéma du lidar : schéma du driver des moteurs : schéma des capteurs ultrasons :



VII. Problèmes rencontrés

Lors de la réalisation de ce projet, nous avons rencontré certaines difficultés dont voici les deux principales :

- Problème de châssis : lorsque nous avons reçus le châssis, il est arrivé avec les mauvaises pièces, ce qui nous a forcés à changer ces pièces avec ce que nous avions et à percer de nouveaux trous. En effet, nous n'avons pu utiliser ceux existant de base car ils n'étaient pas assez espacés ou bien placés. Par ailleurs, comme ce châssis nous forçait à faire varier la vitesse des moteurs pour faire tourner le robot, la rotation du robot a été compliquée à paramétrer en raison du manque de masse sur le châssis, ce qui faisait patiner les roues lors du déplacement du robot.
- ROS : de l'installation à la programmation, chaque étape a donné lieu à de nombreuses erreurs dans la console et ce, même en suivant des tutoriels. Plusieurs recherches internet ont permis de résoudre ces problèmes mais cela nous a beaucoup ralentis.

VIII. Coûts du projet

Les coûts du projet se décomposent comme suit :

Objet	nombre	prix unitaire (€)	prix (€)
Step motor nema	2	12	24
garmin lidar-lite v3hp	1	140	140
Ensemble plastique-courroie-roulement	1	50	50
Châssis	1	120	120
Ensemble visseries et entretoises	1	10	10
rpLidar 2D	1	120	120
capteurs ultrasons HC-SR02	2	5	10
Driver des moteurs	1	6	6
Arduino UNO	1	25	25
Total			505

Membres de l'équipe	Heures de travail	Salaires (€/heures)	Salaires (€)	Total (€)
Matthias	200	23,75	4750	6175
Esteban	60	23,75	1425	
Coût total :	6680 €			

Le coût total de notre projet est de 6680€.

IX. Conclusion et perspectives d'améliorations

Nous avons réalisé un robot capable de générer une carte en 2D au format png et de se déplacer en évitant les obstacles grâce à des capteurs ultrasons. Nous avons aussi réalisé un support de Lidar permettant, une fois qu'il sera mis sur le robot, de générer une carte en 3D. Nous estimons que nous avons répondu en quasi-totalité au cahier des charges que nous nous sommes fixés en introduction. La seule réelle amélioration à faire par rapport au cahier des charges est l'autonomie du robot. L'année prochaine, nous pensons utiliser ROS pour prochainement développer la navigation autonome de notre robot. Nous pensons le faire en utilisant les coordonnées des points collectés par le Lidar pour indiquer la direction à suivre au robot. Nous pensons aussi ajouter un troisième capteur ultrasons et modifier l'algorithme liés à ces capteurs pour améliorer la détection d'obstacle. Nous souhaitons aussi créer notre propre châssis.

X. Bibliographie

https://journals.openedition.org/vertigo/15743
https://www.persee.fr/doc/quate_0004-5500_1981_num_18_1_1397
https://www.irobot.fr/fr_FR/roomba.html
https://escadrone.com/gammes/drones-aeriens/cartographie/
https://veillecarto2-0.fr/2022/01/14/des-robots-electriques-cartographient-le-fond-marin-la-derniere-frontiere-de-la-terre/
https://fr.wikipedia.org/wiki/Mars_Exploration_Rover
https://spectrum.ieee.org/wheels-are-better-than-feet-for-legged-robots
https://pastel.archives-ouvertes.fr/pastel-00935600/file/2012ENMP0103.pdf
https://www.instructables.com/YAAR-Yet-Another-Autonomous-Robot/
https://community.robotshop.com/robots/show/parovoz-6wd-all-terrain-robot-platform
https://charleslabs.fr/en/project-3D+Lidar+Scanner+MK2
https://learn.e.ros4.pro/fr/
https://automaticaddison.com/how-to-build-an-indoor-map-using-ros-and-lidar-based-slam/
https://www.theconstructsim.com/start-self-driving-cars-using-ros/
http://wiki.ros.org/fr/ROS/Tutorials