

Projet CSE : Localisation GPS du robot Agribot



Départements : TIN et TIC
Unité d'enseignement CSE

Auteurs : **Bondallaz Corentin**
Derder Hakim
Penalva Carl

Professeurs : **Hochet Bertrand**
Mosqueron Romuald

Classe : **CSE**

Salle : **A07**

Date : **14 juin 2022**

Table des matières

Table des matières	2
1 Introduction	4
1 Cahier des charges	4
2 Description du projet	4
3 Documentation existante	5
4 Fonctionnement de l'Agribot.....	5
4.1 Hardware	5
4.1.1 Schéma de câblage du robot.....	7
4.2 Software	7
4.2.1 Localisation	7
4.2.2 Architecture ROS.....	8
5 Prise en main de l'Agribot.....	9
5.1 Lancement pilotage manuel	9
5.2 Lancement de l'acquisition de données GPS	9
6 Développement	10
6.1 Alimentation par batteries.....	10
6.2 Mise en place de la localisation	11
6.2.1 Configuration GPS du robot	11
6.2.2 Configuration de la Base Station.....	12
6.3 Possibilités d'amélioration.....	13
7 Conclusion.....	14
7.1 Revue des objectifs	14
7.2 Commentaires personnels	14
8 Annexes.....	15
8.1 Anciens rapports	15
8.2 Evaluation du rapport	15
9 Bibliographie	15

LISTE DES FIGURES

Figure 1 Agribot. Annexe [2]	4
Figure 2 Arrosage des champs. Annexe [1].....	4
Figure 3 Agribot étage 1. Annexe [2]	5
Figure 4 Agribot étage 2. Annexe [2]	6
Figure 5 Agribot étage 3, vue de côté.....	6
Figure 6 Agribot étage 3, vue de dessus	6
Figure 7 Schéma de câblage.....	7
Figure 8 Schéma localisation RTK. Source [2]	7
Figure 9 Architecture ROS du robot.....	8
Figure 10 Exemple des frames utilisées pour un package ROS	8
Figure 11 Première idée de convertisseur DC-DC. Source [3]	10
Figure 12 Carte convertisseur DC-DC actuellement installé	10
Figure 13 Configuration X-Sens.....	11
Figure 14 Visualisation de la localisation avec MT-Manager.....	11
Figure 15 Visualisation localisation avec MT-Manager	11
Figure 16 Localisation avec antenne fixe	12
Figure 17 Visualisation avec RVIZ.....	13

1 Introduction

Le principal but de notre projet de CSE est de reprendre le projet Agribot pour en améliorer les fonctionnalités et clarifier la documentation.

1 Cahier des charges

Pour ce projet, nous nous sommes donné plusieurs objectifs :

- Rechercher et comprendre le fonctionnement hardware et software de l'Agribot.
- Regrouper les procédures importantes pour prendre en main le projet
- Comprendre et identifier les problèmes rencontrés par Mme Martin durant son TB.
- Améliorer le système d'alimentation de l'Agribot qui comporte des défauts.
- Apprendre à utiliser l'Agribot pour recevoir des données GPS avec une précision de 1 cm.
- Améliorer la localisation GPS du robot.

2 Description du projet

Le projet Agribot est proposé à l'origine par La Ferme du Moulin BIO à Bavois. Elle désire avoir un système autonome pour pouvoir arroser suffisamment les plantons de son champ, qui nécessitent une quantité d'eau importante. Un tel robot parcourrait automatiquement les lignes du champ pour arroser chaque planton, permettant ainsi d'optimiser l'eau utilisée pour l'irrigation.

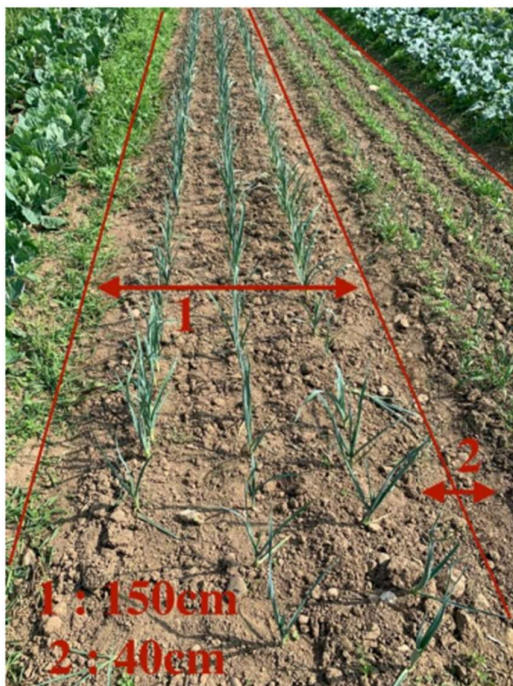


Figure 2 Arrosage des champs. Annexe [1]



Figure 1 Agribot. Annexe [2]

L'Agribot est un projet qui a été le sujet de plusieurs travaux au fil des semestres. En l'état actuel, certaines parties importantes du projet, comme la reconnaissance des plantons par analyse d'image ou la réalisation du bras d'arrosage ont pu être développés dans le cadre de projets de bachelor notamment. Mais ce sont des travaux indépendants au prototype montré ci-dessus.

Le déplacement autonome d'un prototype de base roulante en intérieur avec détection d'obstacle a aussi pu être réalisé et c'est ce prototype-là qui a été repris par Mme Martin pour s'occuper de la géolocalisation en extérieur. Il y a cependant quelques points à améliorer concernant la localisation GPS ainsi que l'alimentation du robot. Ces deux points seront décrits plus en détails au point 7.3.

3 Documentation existante

Pour ce projet, le rapport de TB de Mme Eloïse Martin a pu être récupéré (Annexe [1]) ainsi que la description des spécifications de l'Agribot de Mr. Arzur Gabriel Catel Torres (Annexe [2]). Un tutoriel expliquant comment mettre en place une localisation GPS en se basant sur les composants utilisés a aussi été utilisé (Source [1]).

4 Fonctionnement de l'Agribot

4.1 Hardware

Le robot Agribot est composé de plusieurs éléments. Les différents étages ainsi que les roues proviennent d'un kit livré en pièces détachées. Le reste du premier étage, qui permet d'assurer le déplacement en intérieur et en extérieur, est composé des éléments suivants :

- **4x Servomoteurs**. Ils fonctionnent en recevant du PWM mais peuvent tourner en continu. Ils fonctionnent en 12V et peuvent consommer jusqu'à 20A en pic.
- **4x Drivers** de moteurs *Cytron MD10C*. Ils peuvent alimenter des moteurs de 5 à 30V et peuvent fournir jusqu'à 13A en continu et 30A en pic.
- **1x Batterie** LiPo 12V, 8500 mAh et spécialement conçue pour alimenter des moteurs.
- **1x Carte i2cPWM** board – *ServoShield*. Cette carte communiquant en I2C permet d'envoyer des signaux PWM aux drivers pour contrôler la vitesse des moteurs.
- **1x Interrupteur** général permettant de contrôler toute l'alimentation.

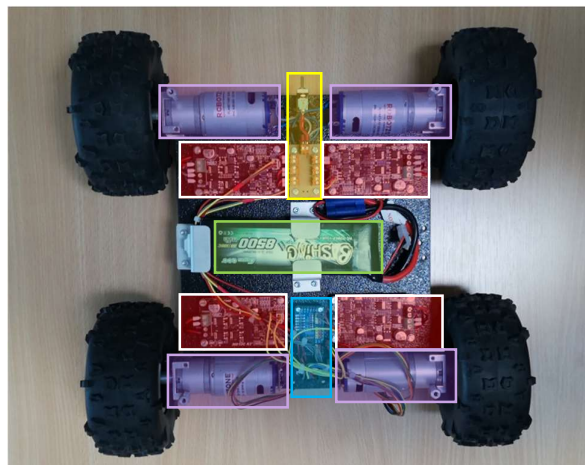


Figure 3 Agribot étage 1. Annexe [2]

Le second étage du robot est consacré à la détection d'obstacle. Il est composé des éléments suivants :

- **1x Rplidar A1**. Il s'agit d'un Scanner rotatif à 360° avec une portée de 12m. Sa fréquence de détection peut aller de 2 à 10Hz et est modifiable avec un signal PWM. La fréquence qui a été choisie est actuellement de 5.5Hz. Cet étage ainsi que les support du 3eme étage en force de C sont fait de sorte à entraver le moins possible la vision du scanner.
- **1x AgriBot Sensor Board**. Cette carte développée par le REDS permet de récupérer les données des encodeurs des moteurs avec une grande fréquence d'acquisition et communique la position en I2C au microcontrôleur central du robot.

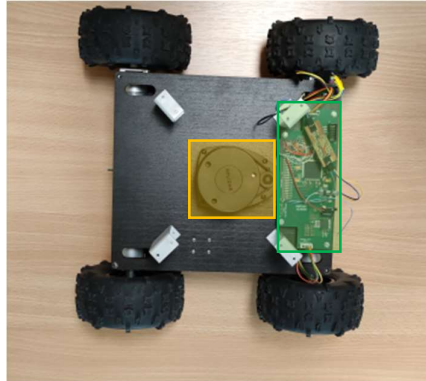


Figure 4 AgriBot étage 2. Annexe [2]

Enfin, le 3^{ème} étage comporte la partie électronique de contrôle ainsi que la localisation.

- **1x Xsens Mti 680G**. Ce module permet de recevoir des données RTK et GPS pour se localiser en extérieur. Elle est également équipée d'accéléromètres. Sa grande précision permettra d'obtenir une localisation de l'ordre du centimètre.
- **1x Antenne GPS**
- **1x module radio X-Bee** monté sur un adaptateur X-Bee – RS232 pour recevoir les données RTK de la balise. Cette dernière possède le même module.
- **1x Ecran tactile** qui ne sera pas utilisé pour ce projet.
- **1x Caméra T265**. Cette caméra ne sera pas utilisée mais a déjà servi pour de l'odométrie par analyse d'image.
- **1x Carte d'alimentation 5V**. Cette carte permet de convertir les 12V de la batterie en 5V pour alimenter toute la partie électronique de contrôle.
- **1x Raspberry Pi 4**. Il s'agit du microcontrôleur principal.

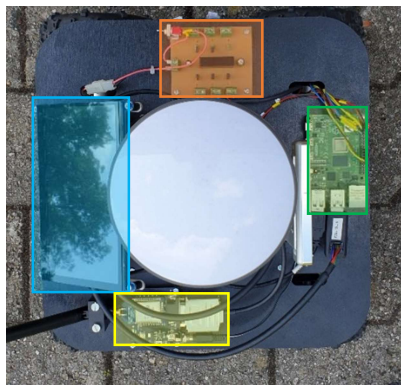


Figure 6 AgriBot étage 3, vue de dessus

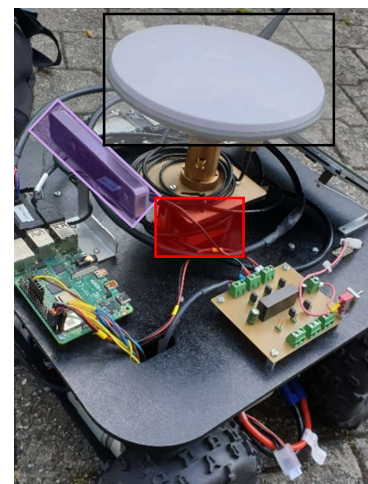


Figure 5 AgriBot étage 3, vue de côté

Le robot va fonctionner ainsi : le X-Sens va récupérer des données GPS récupérées avec son antenne ainsi que des données RTK transmises par le module radio, qui les reçoit d'une balise fixe (voir point 4.2.1). Ainsi, le X-sens est capable d'envoyer au Raspberry Pi 4 une localisation précise au centimètre. Il va en parallèle récupérer l'odométrie provenant des 4 roues motorisées et fusionner cette odométrie avec les données de positionnement RTK pour avoir une localisation robuste qui se corrige automatiquement.

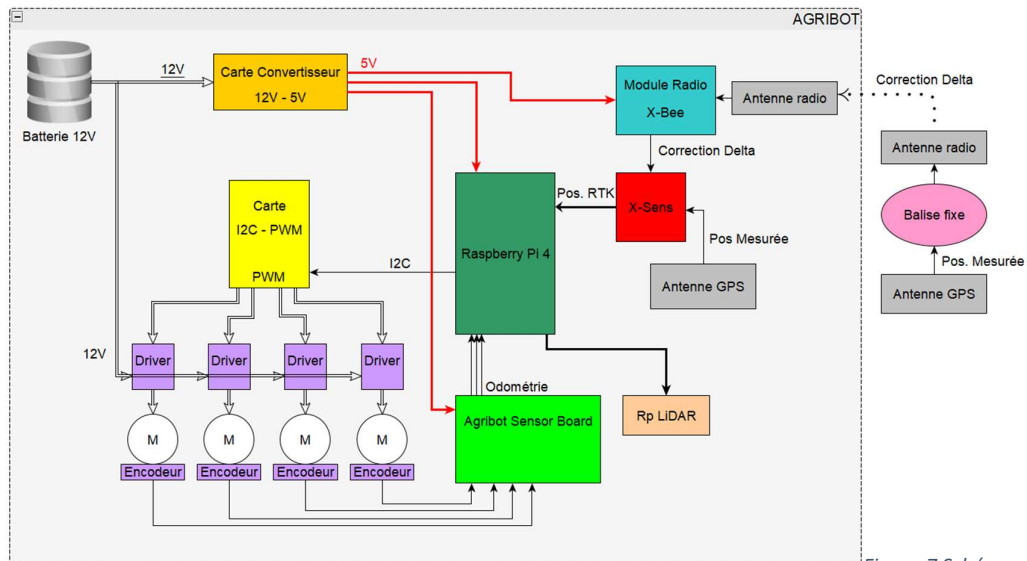


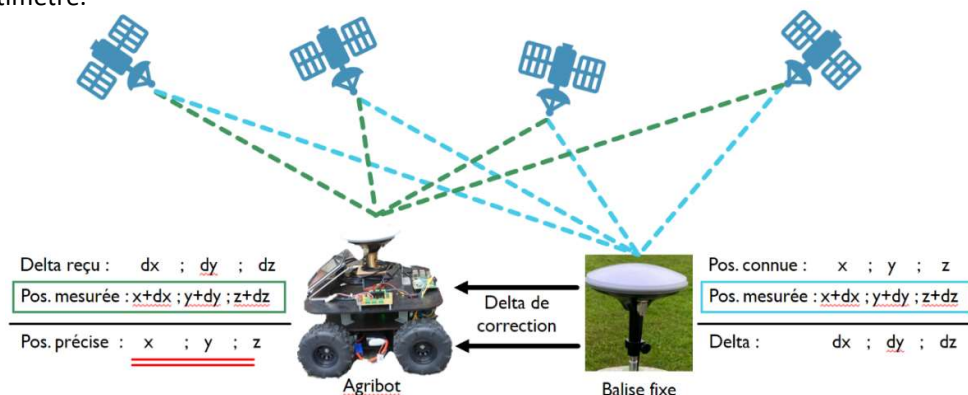
Figure 7 Schéma de câblage

4.2 Software

4.2.1 Localisation

Le fonctionnement de la localisation en extérieur se passe de la façon suivante (voir ci-dessous, source [2]) : L'Agribot va utiliser l'antenne GPS connectée au Xsens pour récupérer des données de localisation. Mais cette méthode n'est pas parfaite et possédera des imprécisions, du au déplacement constant des satellites au-dessus. Pour compenser ces imprécisions, une balise fixe possédant la même antenne et dont la localisation exacte est connue est placée à côté du champ. Cette localisation peut facilement prendre plusieurs heures pour se peaufiner, mais une fois la position acquise, elle servira de référence à l'Agribot.

Ainsi, les 2 antennes GPS reçoivent logiquement les mêmes perturbations dues aux satellites, mais la balise peut alors mesurer le delta entre ce qu'elle mesure et sa position de référence et va donc envoyer à l'Agribot un correctif de position en temps réel. La précision de l'Agribot peut alors atteindre le centimètre.



4.2.2 Architecture ROS

ROS (Robot Operating System) est un ensemble d'outils et de bibliothèques permettant de développer du software pour robot. Dans ce projet, c'est utilisé pour implémenter toutes les fonctionnalités de l'Agribot.

L'avantage de ROS est qu'il facilite la mise en place de la localisation avec les packages disponibles. Nous avons utilisé *robot_localization*, qui donne accès à des nœuds permettant d'utiliser les données de différents capteurs pour mettre en place la localisation d'un robot. Il n'est pas judicieux d'utiliser uniquement les données GPS pour effectuer la localisation. Le package nous permet de fusionner les données de plusieurs capteurs afin d'obtenir une position précise. Dans notre cas, nous utilisons les données de l'odométrie des roues, de l'IMU (Inertial Measurement Unit) et du GPS. Nous utilisons deux nœuds *ekf_localization_node* qui sont une implémentation d'un filtre étendu de Kalman. Ce filtre est un filtre à réponse impulsionnelle infinie qui estime les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées. Le nœud utilise un modèle de mouvement omnidirectionnel pour projeter l'état dans le temps, et corrige cette estimation en utilisant les données des capteurs. L'un des *ekf_localization_node* contient la fusion des données de l'IMU et de l'odométrie des roues et le second contient la fusion de l'IMU, de l'odométrie et du GPS. Nous avons également besoin d'un nœud *navsat_transform_node* afin de convertir les coordonnées GPS longitude et latitude en coordonnées x et y relative à la map du robot. Ci-dessous une image illustrant comment les différents nœuds se connectent pour fournir au final une frame d'odométrie et une frame map. Une frame est un système de coordonnées spécifique à un capteur. Les frames sont reliées entre elles grâce à des transformations afin que le robot puisse se retrouver dans une frame de référence. La frame représentant le monde est ici la map.

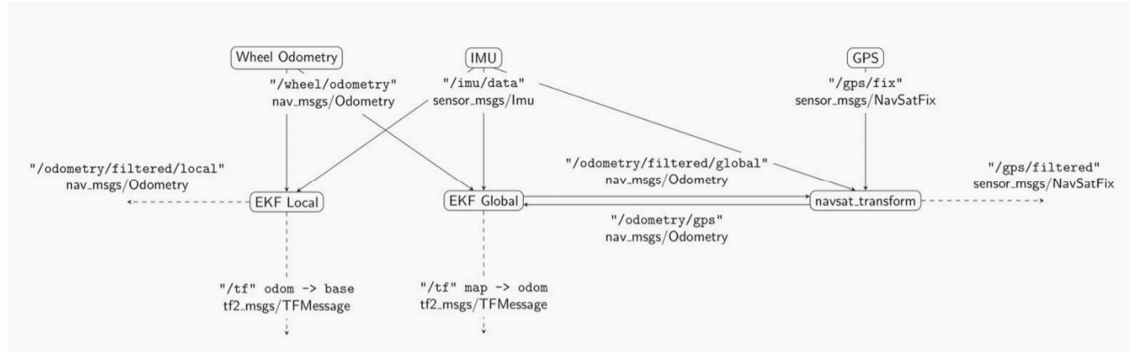
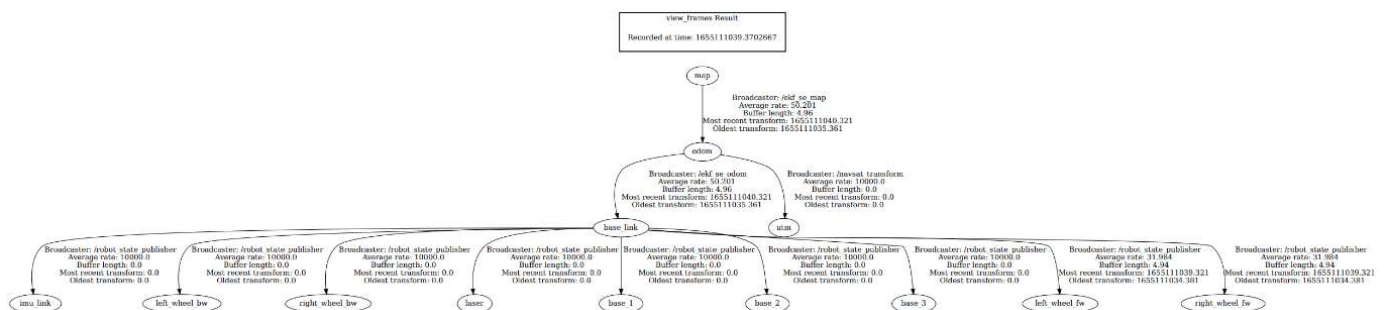


Figure 9 Architecture ROS du robot

On voit ici l'arbre des frames obtenu à la fin du projet. On voit la frame d'odométrie et la frame map au-dessus.



5 Prise en main de l'Agribot

5.1 Lancement pilotage manuel

Il existe un programme de démo pour pouvoir piloter l'Agribot manuellement avec une manette de PS4. Le lancement de cette démo fut un bon point de départ pour prendre en main le projet.

1. Pour commencer, allumer l'Agribot en levant les 2 switch du 1^{er} et du 3^{ème} étage. Veiller à ce que la batterie 12V soit bien chargée.
2. Lancer un terminal Git Bash sur un ordi connecté au réseau HEIG-Devices.
3. Se connecter en SSH à l'Agribot avec la commande « `ssh ubuntu@agribot` » (mdp : redsreds).
4. Se connecter de la même façon sur un second terminal.
5. Sur le 1^{er} terminal, lancer la synchronisation de la manette avec la commande « `ds4drv` ».
6. Appuyer sur share+PS sur la manette.
7. Sur le second terminal, taper « `roslaunch agribot agribot_manual_mapping.launch` ».
8. Appuyer sur L2+R2 sur la manette.
9. A présent la manette est connectée au Pi4 qui permet de piloter le robot manuellement.

5.2 Lancement de l'acquisition de données GPS

Cette procédure explique comment récupérer des données RTK avec la station de base pour les envoyer par radio à l'Agribot pour qu'il augmente la précision de sa localisation. A noter que cela ne fonctionnera que si le robot est placé en extérieur, dans une zone relativement dégagée et à ciel ouvert.

1. Pour commencer, allumer l'Agribot en levant le switch du 3^{ème} étage. Veiller à ce que la batterie 12V soit bien chargée. Ceci est nécessaire pour allumer le module radio du robot.
2. Débrancher le câble USB du Xsens MTi-680G du robot et le brancher sur un ordinateur.
3. Ouvrir MT Manager sur l'ordinateur et scanner les ports pour se connecter au robot.
4. Allumer la station de base en utilisant le switch et la batterie à l'intérieur ou en se connectant en USB à un ordinateur. L'antenne de la balise doit absolument rester stable et immobile.
5. Attendre que la station de base soit correctement calibrée.
6. Dans MT Manager, ouvrir l'onglet *Status Data* et vérifier que les signaux *GNSS fix* et *RTK status* soit à 1, c'est à dire tout en haut du graphique. Le signal *RTK status* peut mettre un moment avant de passer à 1. S'il se trouve à 0.5, cela veut dire que les messages RTCM sont reçu mais le RTK n'est pas fait.
7. Toujours dans MT Manager, ouvrir un onglet *Position Data* et déplacer le robot afin de constater le fonctionnement de la localisation GPS.
8. Une fois les tests effectués, on peut débrancher le X-Sens de l'ordinateur et le remettre sur la pi du robot. Un redémarrage du robot est ensuite recommandé.

6 Développement

6.1 Alimentation par batteries

Un des points qui restait à améliorer était l'alimentation par batterie. Avant que le projet n'ait été récupéré en début de semestre, le Pi4 redémarrait promptement dès que les roues commençaient à bouger. Ce problème est dû à une surconsommation de la batterie à cause d'un trop fort appel de courant des roues, ce qui a pour conséquence de sous-alimenter le Pi4 et force un redémarrage.

Pour contrer ce problème, une seconde batterie avait été ajoutée mais ne constitue pas une solution très pratique. Il fallait revenir à 1 seule batterie pour tout le système. C'est pourquoi un convertisseur DC-DC a été recherché pour transformer le 12V de la batterie en 5V pour toute la partie électronique de contrôle (voir figure 12). Il fallait veiller à ne pas prendre un simple convertisseur 5V, mais plutôt aller jusqu'à 5.1 ou 5.2V pour anticiper les éventuelles chutes de tension causées par les moteurs, dans le cas où la batterie ne serait pas entièrement chargée. La solution de gauche était une solution simple et avec de bonnes fixations, mais avec la journée Portes-Ouvertes de la HEIG-VD, le PCB de droite a été installé en urgence. Etant donné que le convertisseur choisi fonctionne très bien également, la solution a été gardée.



Figure 11 Première idée de convertisseur DC-DC. Source [3]

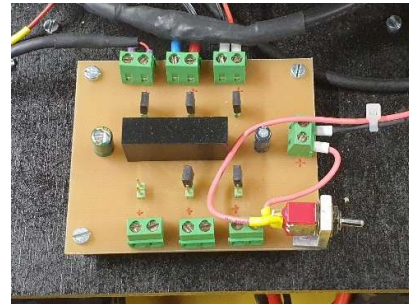


Figure 12 Carte convertisseur DC-DC actuellement installé

6.2 Mise en place de la localisation

6.2.1 Configuration GPS du robot

Pour commencer, nous avons utilisé le logiciel fourni par Xsens, MT Manager, afin de vérifier le bon fonctionnement du GPS et du RTK. Pour ceci, il faut brancher le module Xsens en USB à l'ordinateur.

Ensuite dans MT Manager, scanner les ports afin de le détecter, et appliquer la configuration suivante dans *Device settings* (un fichier de configuration avec celle-ci est disponible en annexe 4) :

The screenshot shows the 'Device settings' window in MT Manager. It has tabs for 'Xbus mode', 'String report mode', and 'CAN mode'. The 'String report mode' is selected. The 'Preset' dropdown is set to 'Default'. There are checkboxes for 'Link Formats' and 'Link Freqs'. The 'Timestamp' section has checkboxes for 'Packet Counter' (checked), 'Sample Time Fine' (checked), 'Sample Time Coarse' (unchecked), and 'UTC Time' (unchecked). The 'Orientation' section has a dropdown set to 'Quaternion'. The 'Inertial Data' section has checkboxes for 'Δq' (checked), 'Rate of Turn' (unchecked), 'Δv' (checked), 'Acceleration' (unchecked), and 'Free Acceleration' (unchecked). The 'Magnetic Field' section has a checkbox for 'Magnetic Field' (unchecked). The 'Temperature' section has a checkbox for 'Temperature' (unchecked). The 'Pressure' section has a checkbox for 'Barometric Pressure' (unchecked). The 'High-Rate Data' section has checkboxes for 'Acceleration HR' (unchecked) and 'Rate of Turn HR' (unchecked). The 'Status' section has checkboxes for 'Status Word' (checked) and 'Status Byte' (unchecked). The 'Position and Velocity' section has checkboxes for 'Latitude and Longitude' (checked), 'Ellipsoid Altitude' (dropdown set to 'Fixed Point 16.32'), 'Velocity' (checked), and 'GPS Age' (checked). The 'GNSS Data' section has checkboxes for 'Pvt Data' (checked), 'SatInfo' (checked), and 'Pvt Pulse Timestamp' (checked). The right side of the window shows various data format and frequency settings for each section.

Figure 13 Configuration X-Sens

Ceci permet d'activer le GPS. Pour vérifier les données, ouvrir l'onglet *Position* (à droite sur l'image ci-dessous). On voit ici les coordonnées de la HEIG-VD. Également ouvrir l'onglet *Status Data*, celui-ci permet de vérifier que le RTK est activé (à gauche sur l'image), uniquement possible après calibration avec la station de base.

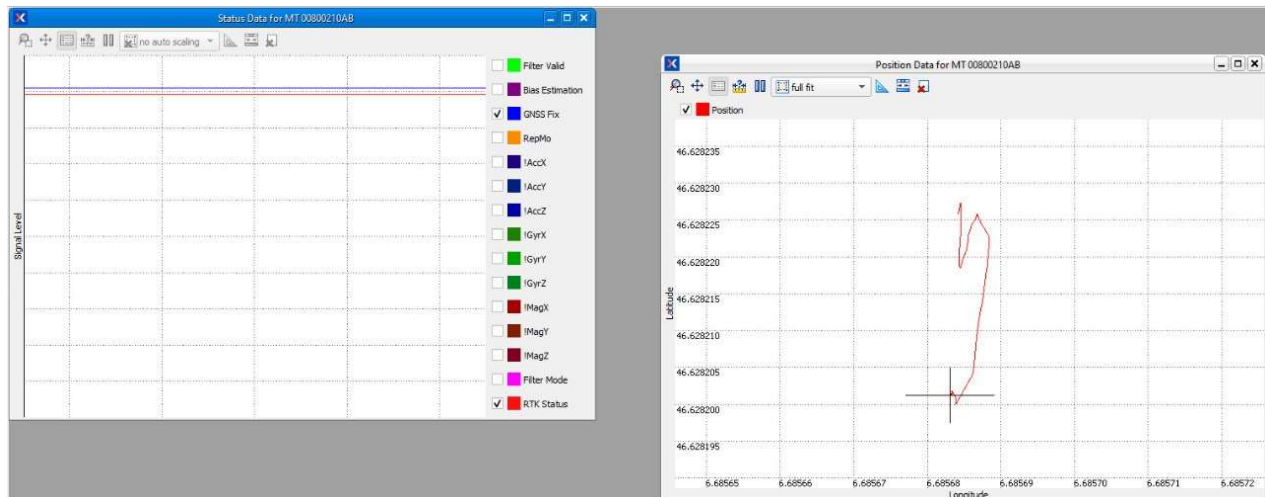


Figure 14 Visualisation de la localisation avec MT-Manager

6.2.2 Configuration de la Base Station

Pour mettre en place la station de base, nous avons utilisé le logiciel U-Center de u-blox qui permet de configurer le module GPS ZED-F9P. Nous avons suivi un tutoriel Github (en source [1]) et utilisé les fichiers de configuration créés lors du précédent TB (en annexes 5 et 6). Nous avons principalement utilisé le mode survey in, comme nous n'avions pas d'endroit fixe où placer la station de base. Dans ce mode, la station doit se calibrer et le processus peut prendre plusieurs heures. La précision voulue est paramétrable dans le menu UBX/CFG/TMODE3. Pour accélérer, nous avons eu recours à un NTRIP caster. Le NTRIP (Network Transport of RTCM via Internet Protocol) permet de recevoir des messages de corrections de positions d'une antenne GPS grâce à une connexion internet. L'antenne choisie se trouve à Zimmerwald dans le canton de Berne. Avec cette mise en place, nous arrivons à obtenir une précision de 3 cm en environ 2 heures. Une fois dans son emplacement final la station de base pourra être passée en mode position fixe et la calibration ne prendra plus autant de temps.

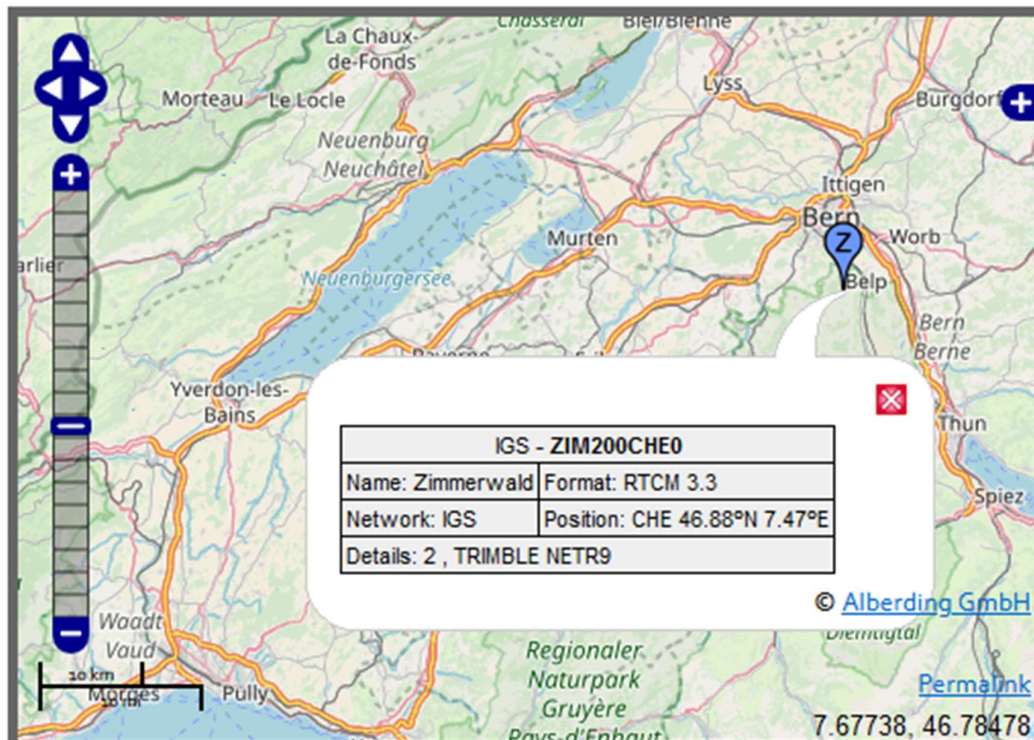


Figure 16 Localisation avec antenne fixe

6.3 Possibilités d'amélioration

Nous n'avons pas réussi à faire fonctionner la localisation. Le problème vient sans doute d'une mauvaise utilisation du package `robot_localization`. Les données générées individuellement par les capteurs sont corrects. La frame d'odométrie est fonctionnelle mais la fusion des données GPS avec les autres provoque des erreurs. On peut voir le robot tourner sur lui-même sur la map et lorsqu'on le déplace, il semble sauter d'une position à l'autre. Une piste fournie par M. Catel Torres serait que plusieurs topics sont publiés sur la même frame ce qui donne lieu aux sauts de position. Il est possible que des erreurs soit présentes dans les fichiers de configuration des `ekf_localization_node`.

Ci-dessous une illustration du robot dans la map sur RVIZ.

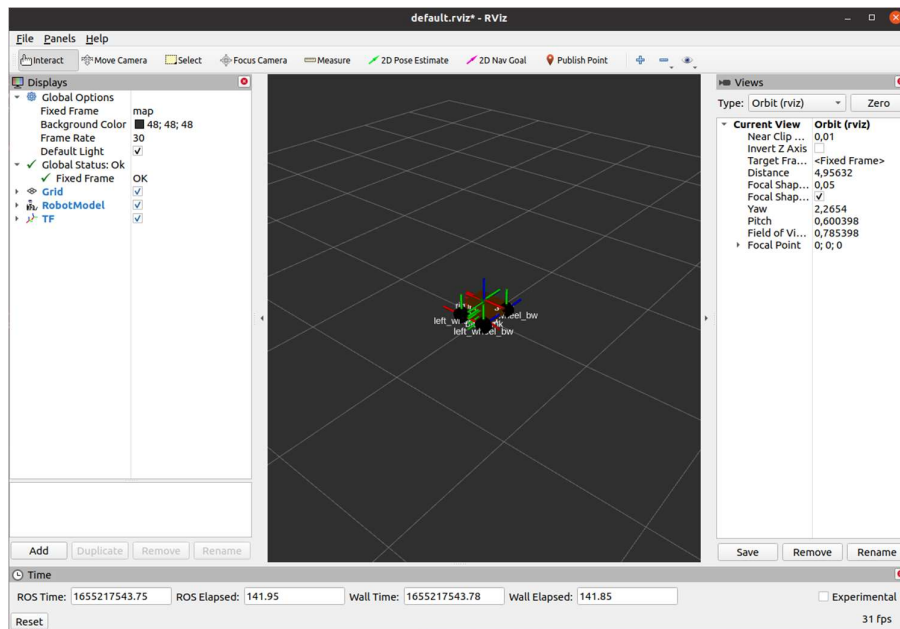


Figure 17 Visualisation avec RVIZ

7 Conclusion

7.1 Revue des objectifs

- Rechercher et comprendre le fonctionnement hardware et software de l'Agribot.

Objectif partiellement réussi. Le fonctionnement hardware du robot est bien compris et pourrait même nous aider à réaliser nos propres travaux de bachelor (notamment les composants utilisés pour la base roulante). Mais la structure software fut très compliquée à comprendre. Nous n'avons pas réussi à bien prendre en main le fonctionnement de ROS et il était très difficile de s'y retrouver dans les très nombreux fichiers et paquets ROS que le projet a accumulé, ce qui est surtout dû au manque de documentation pour cette partie.

- Regrouper les procédures importantes pour prendre en main le projet

Objectif réussi. Le pilotage manuel a pu être réalisé et la procédure générale pour lancer un script ROS est maintenant maîtrisée.

- Comprendre et identifier les problèmes rencontrés par Mme Martin durant son TB.

Objectif réussi. Il s'agissait principalement de la synchronisation des données odométriques des roues avec les données GPS du X-sens. En revanche, nous n'avons pas pu déterminer la cause du problème de la localisation.

- Améliorer le système d'alimentation de l'Agribot qui comporte des défauts.

Objectif réussi. Le système d'alimentation est maintenant efficace et n'a plus de problème de sous-tension. Le fonctionnement du robot avec 1 seule batterie est aussi bien plus pratique.

- Apprendre à utiliser l'Agribot pour recevoir des données GPS avec une précision de 1 cm.

Objectif réussi. On arrive à présent à recevoir les données RTK avec la balise, puis à les envoyer par radio au robot pour apporter correctif de positionnement.

- Améliorer la localisation GPS du robot.

Objectif partiellement réussi. La localisation ne fonctionne pas mais on reçoit des données GPS, ce qui n'était pas le cas au début.

7.2 Commentaires personnels

Ce travail de groupe nous a beaucoup appris sur les méthodes de localisation en extérieur ainsi que sur l'utilisation de ROS. Nous pensons toutefois qu'une restructuration du projet serait bénéfique pour améliorer l'organisation des fichiers dans le Pi4 du robot. Il y a en effet beaucoup de fichiers ROS dispersés et dupliqués, ce qui complique le travail de recherche. Une possibilité pour ceux qui reprendront le projet serait aussi de chercher un package ROS différent de celui utilisé, qui pourrait mieux fonctionner dans notre situation. Il a été compliqué de reprendre un projet aussi avancé sans connaître quoi que ce soit sur ROS, comprendre comment fonctionne le package robot_localization et la solution qui était déjà mise en place fut un challenge.

Date : mardi, 14 juin 2022

Bondallaz Corentin



Derder Hakim



Penalva Carl



8 Annexes

8.1 Anciens rapports

1. Eloïse Martin. (2021). *Travail de Bachelor : Déplacement autonome d'un robot pour l'agriculture durable*. Annexes / TB_Martin_2021_rapport.pdf
2. A.Gabriel Catel Torres. (2021). *Agribot – Specs*. Annexes / agribot_doc.pdf

8.2 Evaluation du rapport

3. HEIG-VD. (2022). *Evaluation Rapports*. Annexes / notation-rapport_template.xlsx

8.3 Fichiers de configuration

4. xsens_custom_config_20211014.xsa
5. simpleRTK2B_fixed_position_20211015.txt
6. simpleRTK2B_survey_in_10m_20211014.txt

9 Bibliographie

- 1) Source [1] : Michalis Logothetis et Lefteris Griparis. (2020). *xsens_imu_gps_rtk*. https://github.com/ikh-innovation/xsens_imu_gps_rtk
- 2) Source [2] : Wikipédia. National Oceanic and Atmospheric Administration. (2022). https://fr.wikipedia.org/wiki/GPS_différentiel
- 3) Source [3] : Distrelec. (2022). *TMDC 40-2411-DC/DC-Wandler 9...36V 5.1V 8A 40W, TracoPower*. https://www.distrelec.ch/de/dc-dc-wandler-36v-1v-8a-40w-traco-power-tmdc-40-2411/p/30008556?ext_cid=shg0oaqchde-P-Shopping-Fallback&pi=30008556&gclid=Cj0KCQiA95aRBhCsARIsAC2xvfyxTu-T_04TIR9Ab4ROpGjOU0wEo_lw2ITiejZJBXxpGm6oI_goDQoaAq3zEALw_wcB
- 4) Source [4] : Ros.org. (2022). http://docs.ros.org/en/api/robot_localization/html/integrating_gps.html
- 5) Source [5] : Youtube. (2022). *ROS Developers Live-Class #51: How to fuse Odometry & IMU using Robot Localization Package*. <https://youtu.be/QZ5q59H2qaI>
- 6) Source [6] : Youtube. (2022). *ROS Developers Live-Class #52: Localize a robot using GPS*. <https://youtu.be/J77kNrfYKoE>