

Rapport Phase 3



Projet Fil Rouge
Cartographie par vehicule autonome

MAI 2022

ÉQUIPE N°6

Damien SOUBIRANT

Elliott GAUDILLAT

Samuel MISSIER

Quentin PHAN

Loan BERNAT

Tobias BRISSAUD



SOMMAIRE

1 - Conception

1.1 - HARDWARE

1.2 - CHOIX DES MÉTHODES

1.3 - DESIGN DE L'APPLICATION

1.4 - COMMUNICATION ROBOT/APPLICATION

1.5 - MAPPING

2 - Manuel d'utilisation

2.1 - PRENDRE EN MAIN LE LOGICIEL

2.2 - PARAMÉTRER LE SCAN

2.2.1 - CONNEXION BLUETOOTH

2.2.2 - NOM DE LA PIECE

2.3 - SE CONNECTER AU ROBOT

2.4 - COMMANDER LE ROBOT

2.4.1 - MODE MANUEL

2.4.2 - MODE AUTOMATIQUE

2.4.2 - MAPPING

2.5 - HARDWARE

3 - Expériences Personnelles

0

INTRODUCTION

Cartographie par véhicule autonome

La phase 3 du Projet Fil Rouge est une partie totalement indépendante puisqu'elle ne traite pas du moteur de recherche élaboré lors des deux premières phases. Les objectifs de cette phase sont multiples. Le but principal étant de développer un robot mobile capable de cartographier son environnement en toute autonomie.

Le groupe a dû faire preuve de créativité car il n'existe pas de solution unique, nous nous sommes adaptés au contenu de la caisse qui nous a été confiée en proposant des solutions abordables et les plus simples à mettre en place possibles. Cette part de liberté nous a permis de créer et développer des compétences en autodidacte grâce à internet. Cette autonomie s'est aussi caractérisée par la recherche des documentations et leur décryptage sans aide extérieure.

Enfin le travail de groupe a été le maître mot de cette troisième phase de projet puisque très peu encadrée, nous n'avions que peu de contact avec notre client.

1

CONCEPTION

1.1 - Hardware

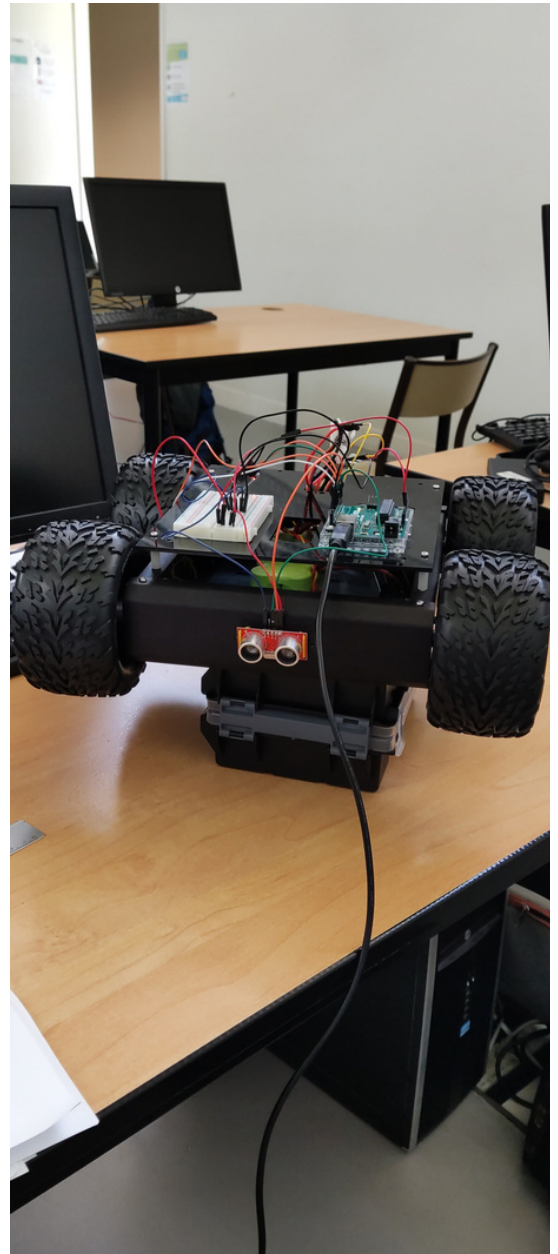
Gaudillat Eliott & Phan Quentin

Le hardware était grandement conditionné par le contenu de la boîte, Cependant nous avons quelques choix à faire.

Souhaitant utiliser le Bluetooth, ceci nous a forcé à utiliser une Arduino Uno au lieu de la Due car la librairie SoftwareSerial nécessaire au Bluetooth n'était pas utilisable sur une Due.

Afin de repérer le robot dans l'espace nous avons décidé d'utiliser deux capteurs ultrasons, un à l'avant du robot et le second sur le côté droit, suite à la demande du groupe du développement software.

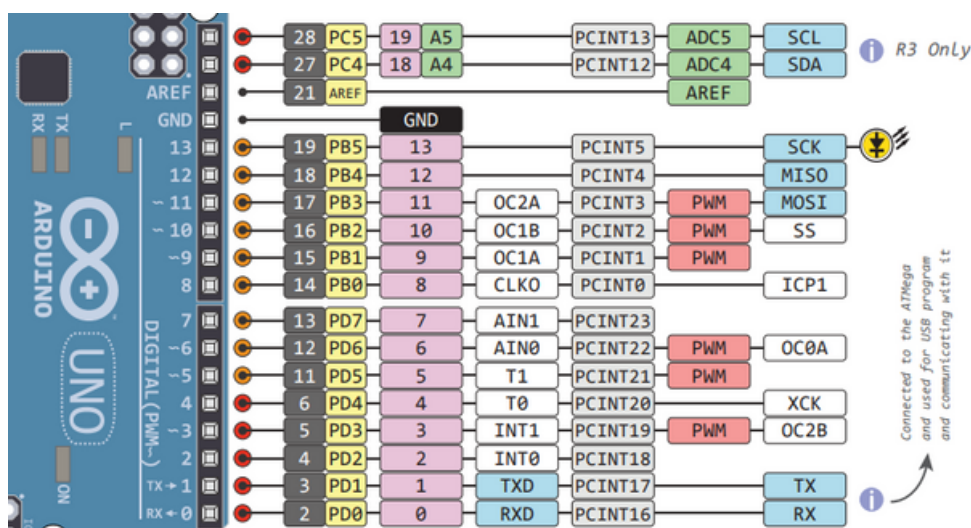
La carte Arduino est alimentée par son port Vin à partir de la sortie 5V du contrôleur moteur.



1 CONCEPTION

Afin de pouvoir mettre en œuvre l'assemblage des différentes parties qui composent le robot, nous avons dû nous documenter sur les ports de la carte Uno.

Ci-dessous les fonctionnalités de chacun des pins :



Le choix des broches était primordial pour la suite et pour la commande des moteurs.

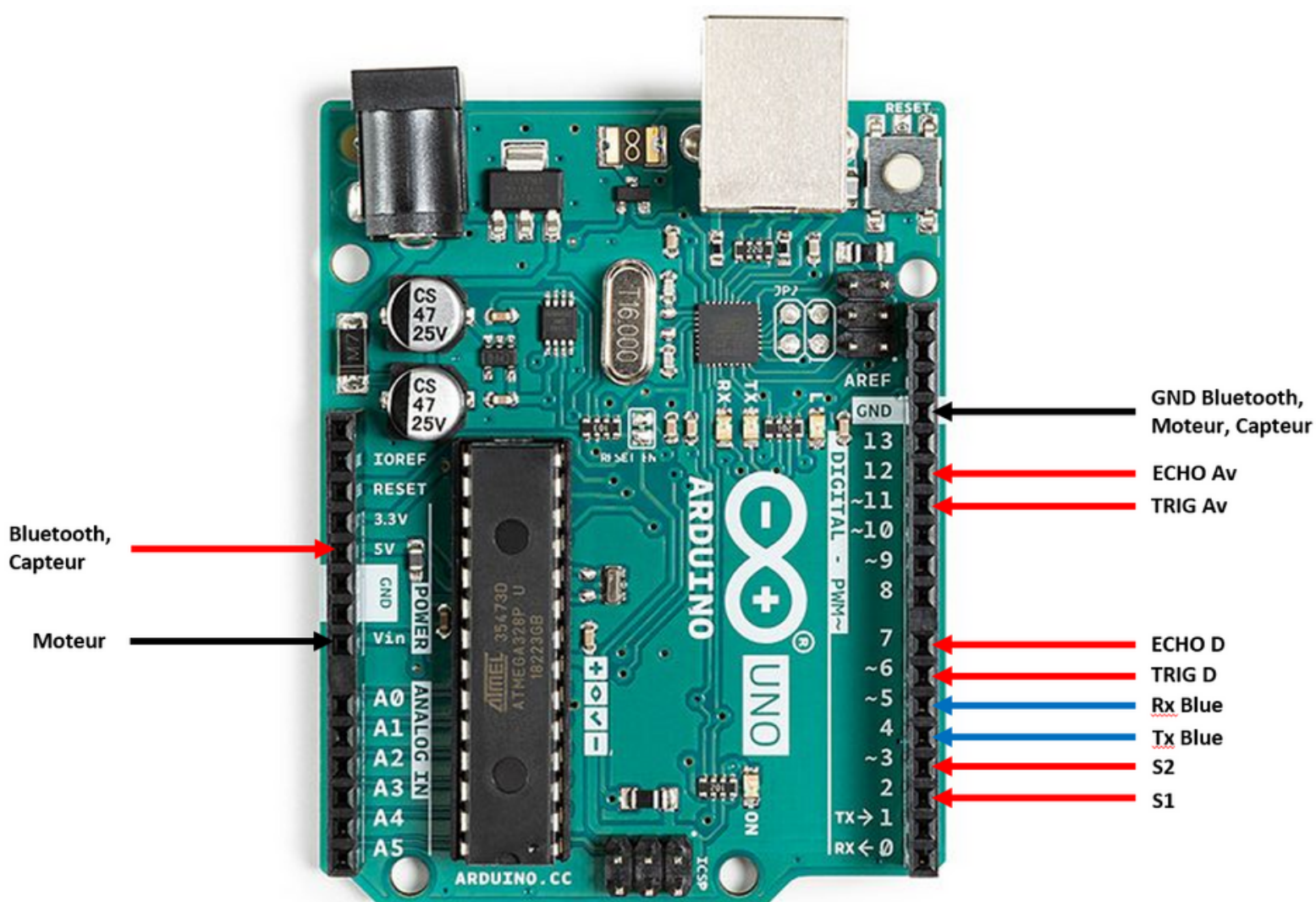
Nous avons alors choisi les broches ayant fonction PWM (Pulse Width Modulation) en sortie numérique, qui nous permettent de faire varier la vitesse de rotation du moteur.

Les signaux numériques ont une valeur de sortie LOW = 0V ou HIGH = 5V qui correspond à la tension d'alimentation de l'Arduino, VDD. L'Arduino Uno est ici équipée d'un microcontrôleur de la famille AVR 8 bits : l'ATMega328, sa simplicité d'utilisation justifie le choix de la carte, bien que compensée par sa faible puissance de calcul.

1 CONCEPTION

A la suite de nos différents choix par rapport aux composants nous avons alors réalisé la disposition de notre matériel en fonction de la carte et de sa documentation technique.

Comme nous l'avons évoqué précédemment le choix de notre branchement afin de contrôler les moteurs était l'une des choses à bien respecter et nous l'avons ainsi mise en œuvre. Pour les connexions des alimentations et des masses nous avons respecté la disposition indiquée dans la documentation.



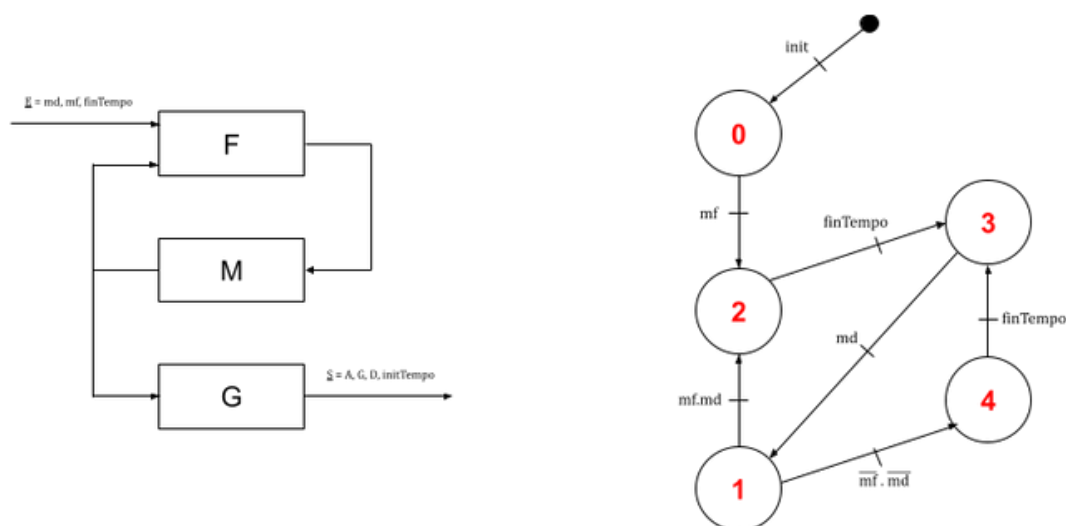
- ⑤ Pin Tx Bluetooth → Réception info Arduino
- ⑥ Pin Rx Bluetooth → Transmission ordre par Arduino

1.2 - Choix des Méthodes

Brissaud Tobias & Missier Samuel

Le comportement du robot est conditionné par sa configuration technique, explicitée plus haut. Avec les deux capteurs ultrasons mis en place, un visant l'avant du robot et l'autre visant son côté droit, nous avons décidé de programmer le robot afin qu'il s'approche d'un mur perpendiculairement, puis effectue une rotation à 90° vers la gauche. Ainsi, le robot s'arrange pour longer le mur à sa droite et effectue des virages à angle droit à gauche comme à droite pour toujours longer un mur du côté droit.

Loan et Eliott ont ainsi pu élaborer une Machine à Etats Finis, donnée ci-dessous, où md et mf représente respectivement les seuils de detection des murs à droite et en face (les variables passent à 1 quand un mur passe le seuil de détection).



Ainsi, en utilisant les fonctions de la librairie Servo.h, il nous a été possible de commander les moteurs et donc d'établir des fonctions simples pour contrôler les mouvements du robot et sa vitesse.

```

void robot_backward() {
    portS1.writeMicroseconds(1350);
    portS2.writeMicroseconds(1350);
}

void robot_left_turn() {
    portS1.writeMicroseconds(1250);
    portS2.writeMicroseconds(1950);
}

void robot_right_turn() {
    portS1.writeMicroseconds(1900);
    portS2.writeMicroseconds(1305);
}

// fonctions de deplacement
void robot_stop() {
    portS1.writeMicroseconds(1500);
    portS2.writeMicroseconds(1500);
    delay(1000);
}

void robot_forward() {
    portS1.writeMicroseconds(1800);
    portS2.writeMicroseconds(1805);
}

```

Pour ce qui est du mode manuel, l'envoi d'une commande par Bluetooth met à jour la commande mise en œuvre par le robot, permettant de contrôler totalement sa trajectoire, grâce aux flèches au clavier.

La fonction dédiée au mode manuel est donnée ci-dessous :

```

while (mode) {
    if (Bluetooth.available()) {
        val = Bluetooth.read();
        if (val == '0') { //avance
            portS1.writeMicroseconds(1750);
            portS2.writeMicroseconds(1750);
        }
        if (val == '1') { //recule
            portS1.writeMicroseconds(1300);
            portS2.writeMicroseconds(1300);
        }
        if (val == '2') { //gauche
            portS1.writeMicroseconds(1300);
            portS2.writeMicroseconds(1900);
        }
        if (val == '3') { //droite
            portS1.writeMicroseconds(1900);
            portS2.writeMicroseconds(1300);
        }
        if (val == '4') { //stop
            portS1.writeMicroseconds(1500);
            portS2.writeMicroseconds(1500);
        }
        if (val == 'a') {
            portS1.writeMicroseconds(1500);
            portS2.writeMicroseconds(1500);
            mode = 0;
        }
    }
    delay(100);
}

```

Le robot, lorsqu'il fonctionne en mode manuel, ne permet pas de cartographier le terrain, cette fonctionnalité n'a pas été implémentée.

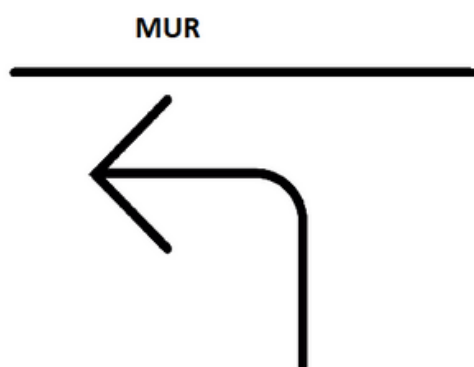
Il est préférable d'utiliser ce mode afin de remplacer le robot et de le relancer en mode automatique.

1 CONCEPTION

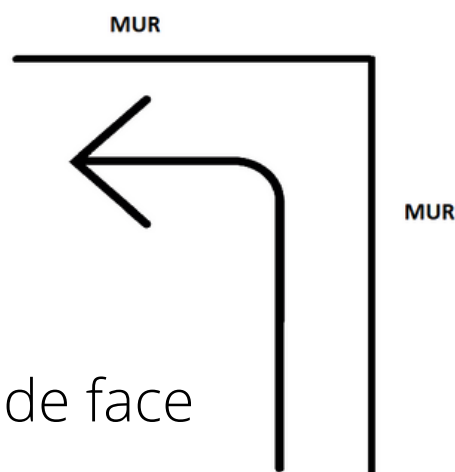
En mode automatique, le robot agit de manière autonome, en contrepartie d'un départ dans une configuration prédéfinie.

Il démarre dans un état d'initialisation et avance tout droit jusqu'à détecter un mur face à lui, il fait ensuite un virage à 90° à gauche pour positionner ce même mur à sa droite.

Il avance ensuite en ligne droite, tant qu'il ne rencontre pas d'obstacle, et tant qu'il perçoit toujours le mur sur sa droite. Le cas échéant, le robot tournera à droite s'il ne perçoit plus de mur à droite ou à gauche s'il détecte un obstacle face à lui.



Etat initial



Obstacle vu de face



Perte du mur droit

Ce comportement autonome est régi par la MEF présentée précédemment, codée dans Arduino en respectant les blocs F-M-G. Le basculement d'état se fait par le contrôle des variables internes, mises à jour par les mesures des capteurs. La boucle principale est cadencée afin de permettre l'envoi des trames Bluetooth sans encombre tout en vérifiant suffisamment souvent les capteurs. Une temporisation permet d'asservir le degré des virages.

Mise à jour des variables internes

```
a = sonar_face.ping_cm();
b = sonar_droite.ping_cm();

mf = ((a < 50) && (a > 0)); // Capte un mur devant
md = ((b < 50) && (b > 0)); // Capte un mur a droite
```

Exemples d'états

```
// bloc F
switch (EtatPresent) {
  case 0 : //Avance jusqu'a trouver un mur devant
    if (mf) {
      robot_stop();
      EtatSuivant = 2;
    }
    break;

  case 1 :
    if (mf) {
      EtatSuivant = 2;
      robot_stop();
    }
    else if (!mf && !md) {
      EtatSuivant = 4;
      robot_stop();
    }
    break;
```

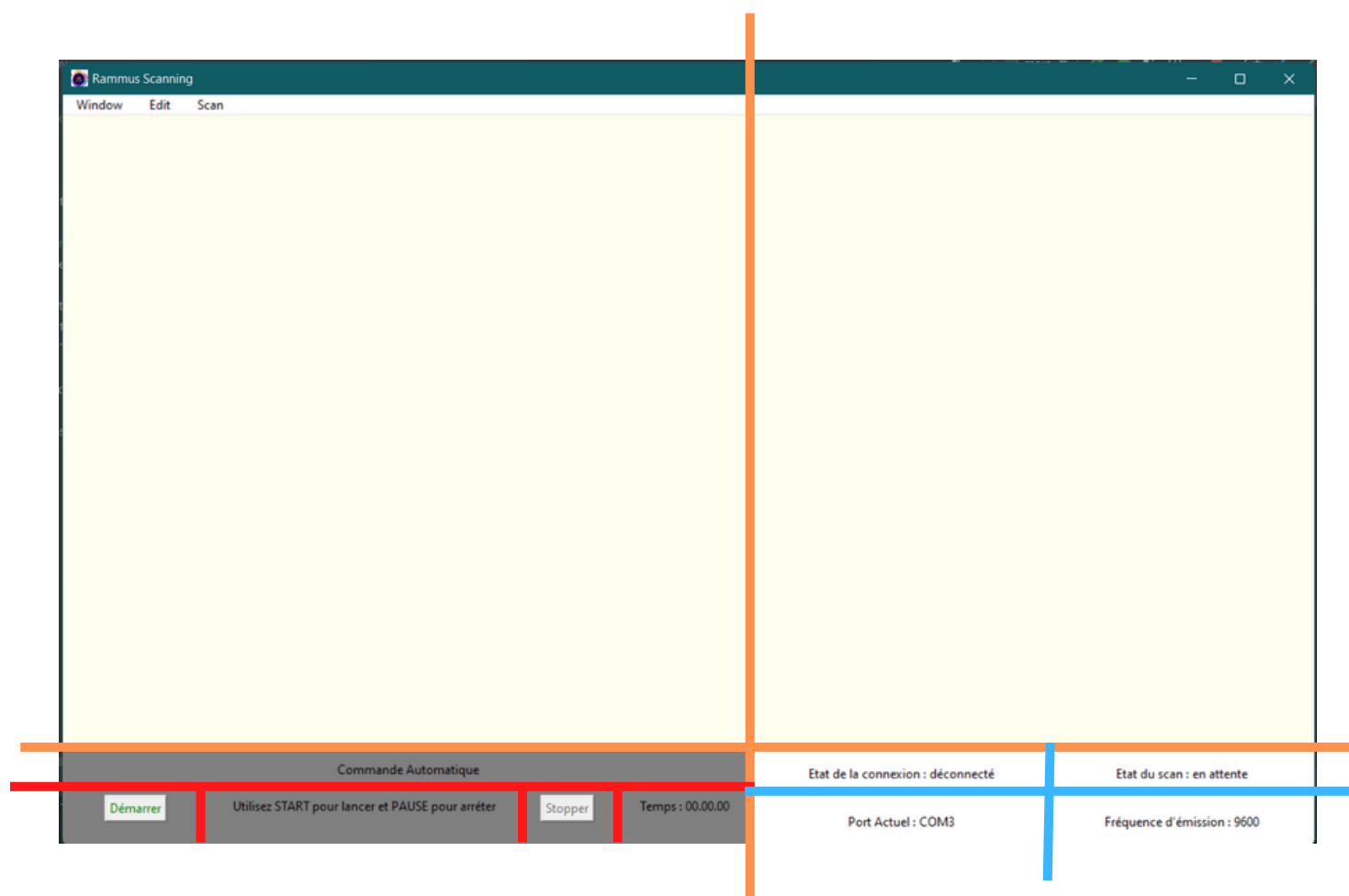
1.3 - Design de l'application

Bernat Loan

Nous avons fait le choix d'utiliser python 3.9 avec la librairie *tkinter*. Nous avons créé une fenêtre principalement composée d'un canvas et d'une bande de paramètres appelée "télécommande".

Notre principal objectif était la simplicité de prise en main et l'ergonomie. La position de la télécommande est au choix pour l'utilisateur, et les boutons de la télécommande s'adaptent au mode sélectionné. L'utilisateur est alerté des erreurs via des fenêtres d'alertes.

Enfin, pour placer les différents éléments sur la fenêtre, on utilise une grille virtuelle car c'est la solution la plus simple et pratique que propose tkinter.



1.4 - La communication Application-Rover

Bernat Loan & Soubirant Damien & Missier Samuel

Dans le matériel mis à notre disposition, nous avons un émetteur/récepteur Bluetooth. Après l'avoir configuré, nous avons fait le choix de transmettre des coordonnées du robot vers l'application et une lettre ou chiffre selon les instructions de l'application vers le robot.

Pour éviter de surcharger le module, l'envoi des coordonnées se fait uniquement lorsque le robot avance en marche avant (états 1 & 3)



Passage en Mode Auto : **"a"**



Passage en Mode Manuel : **"m"**



MODE AUTOMATIQUE

Démarrer : **"p"**



Stopper : **"s"**



"\$10\$20\$" : Renvoi des Coordonnées:



MODE MANUEL

Avancer : **"0"**



Tourner à droite : **"2"**



Tourner à gauche : **"3"**



Reculer : **"1"**



Arrêt sur place : **"4"**

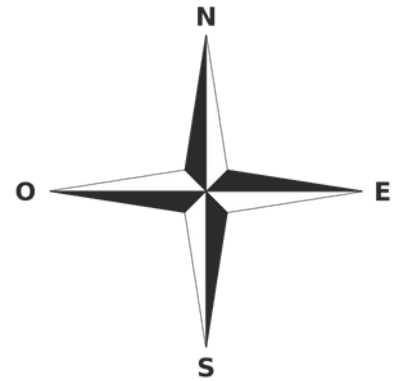


1.5 - Le Mapping

Soubirant Damien

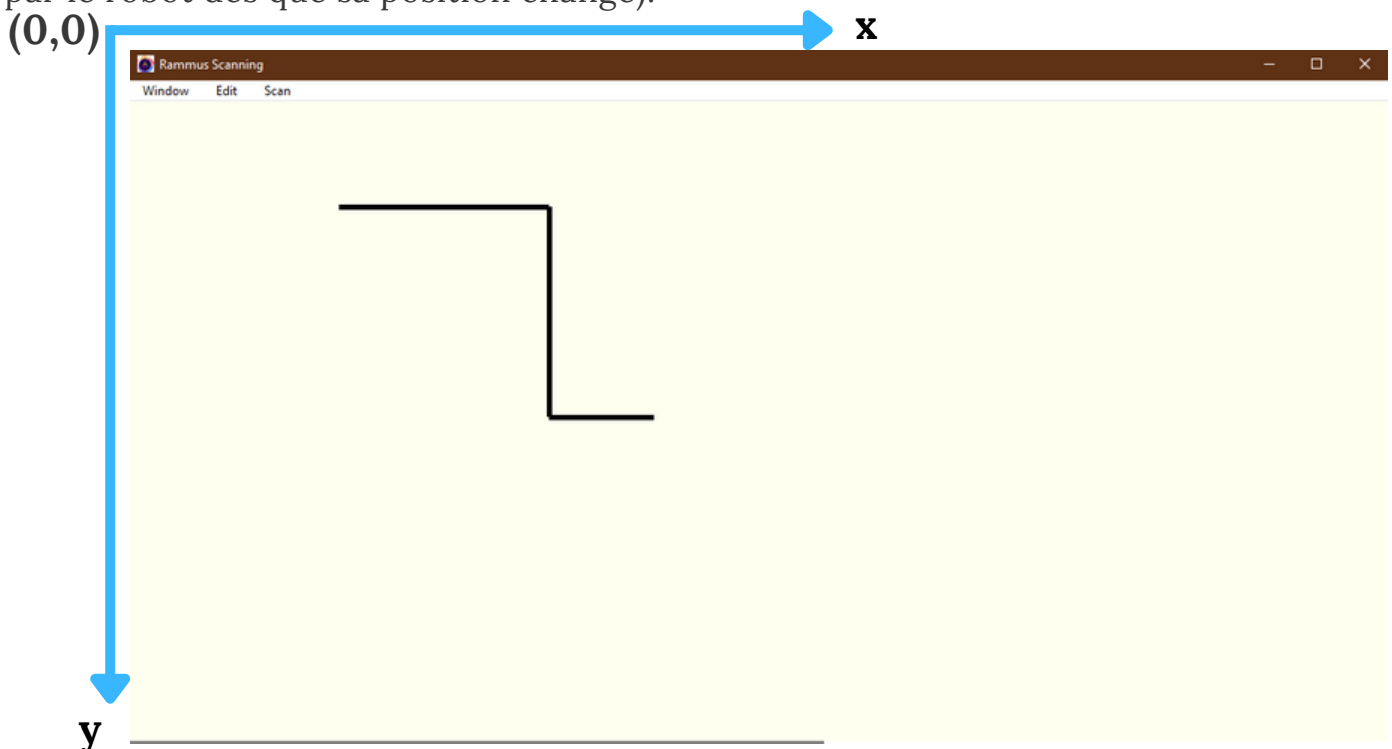
Pour la cartographie de la salle, nous avons opté pour une solution simple et efficace : implémenter deux variables de position et une variable d'orientation au robot.

Les variables de position permettent de suivre le robot selon les axes x et y d'un repère et sont incrémentées à chaque fois que le robot avance dans une direction. Pour savoir la direction dans laquelle le robot avance, nous avons la variable orientation qui est par défaut à NORD et qui change à chaque fois que le robot tourne. Par exemple si le robot tourne à droite l'orientation passera à EST ou si le robot tourne à gauche l'orientation passera à OUEST.



Coté application, nous n'avons donc plus qu'à afficher la position du robot car ce dernier est capable de suivre les murs grâce au mode automatique.

Nous avons fait le choix d'utiliser la librairie *tkinter* et la fonctionnalité canvas qui est utile pour tout ce qui est gestion de visuels. Pour dessiner le tracé sur le canva, il nous faut les coordonnées de deux points et le canva s'occupe de tracer la ligne entre les deux. Nous avons donc décidé de tracer un trait entre l'ancienne position du robot et la nouvelle position à chaque fois que l'on reçoit une trame par Bluetooth (trame envoyée par le robot dès que sa position change).



2

MANUEL D'UTILISATION

2.1 - Prendre en main le logiciel

Rammus Scanning a été conçu de manière à permettre la prise en main la plus simple et rapide possible. Les zones sont clairement délimitées :



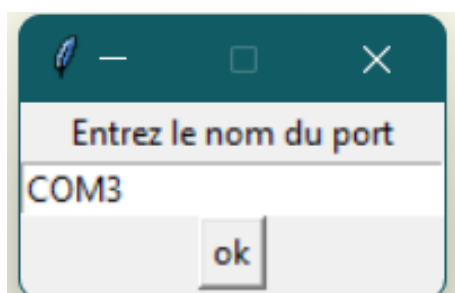
Il est possible de changer l'emplacement de la télécommande en la mettant en format vertical.

Par défaut le logiciel et le robot démarre toujours avec des paramètres prédéfinis :

- Bluetooth connecté au COM3
- Mode automatique par défaut
- Nom de la pièce : defaultRoom

2.2 - Paramétrer le scan

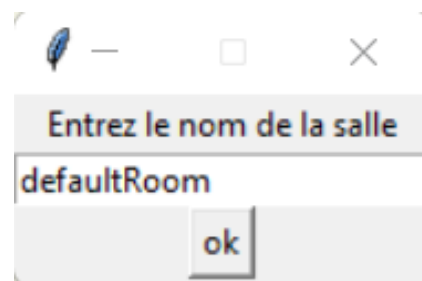
2.2.1 - La connexion Bluetooth



La fréquence est fixée à 9600 bauds et n'est pas réglable. Selon votre appareil, le port Série utilisé par l'application peut changer. Il vous est donc possible de sélectionner le port adéquat depuis le logiciel via le menu **EDIT > Changer le port de communication**.

2.2.2 - Le nom de la pièce

Le nom du fichier .eps que l'on extrait du scan est par défaut "defaultRoom". Vous pouvez le régler dans **SCAN > Changer le nom de la salle**



2.3 - Se connecter au robot

Le robot doit être alimenté et la LED de la carte Bluetooth doit clignoter rapidement. Ensuite, sélectionner le menu **WINDOW > SE CONNECTER**.

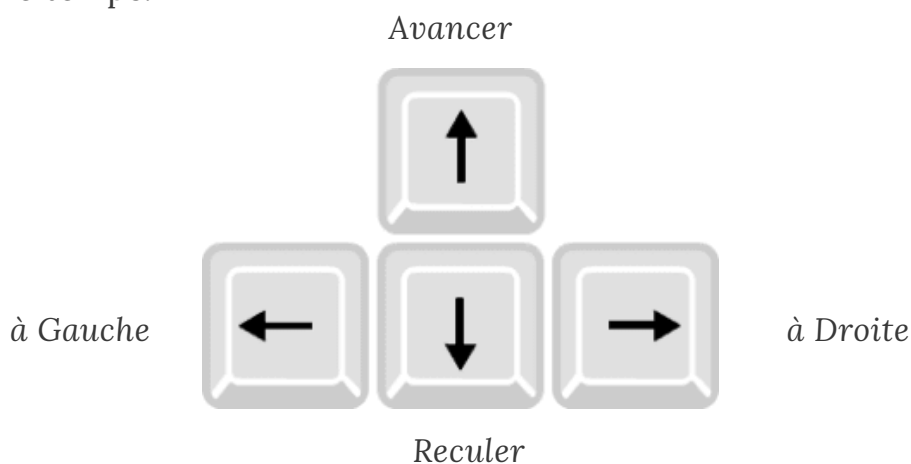
Un écran de chargement va s'ouvrir et le logiciel va tenter de se connecter au robot. En cas d'erreur une notification sera affichée à l'écran. Sinon, la LED du robot se mettra à clignoter moins vite.

2.4 - Commander le robot

2.4.1 - Le mode Manuel

Il est nécessaire d'être en monde manuel pour utiliser les fonctionnalités manuelles du robot **EDIT > Mode Manuel** et être connecté au robot.

Si toutes ces conditions là sont respectées, vous pouvez diriger le robot via les flèches directionnelles. Il faut rester appuyé sur la touche pour que la commande s'exécute. Si vous relâchez la touche le robot s'arrête. Il n'est pas possible d'utiliser plusieurs touches en même temps.

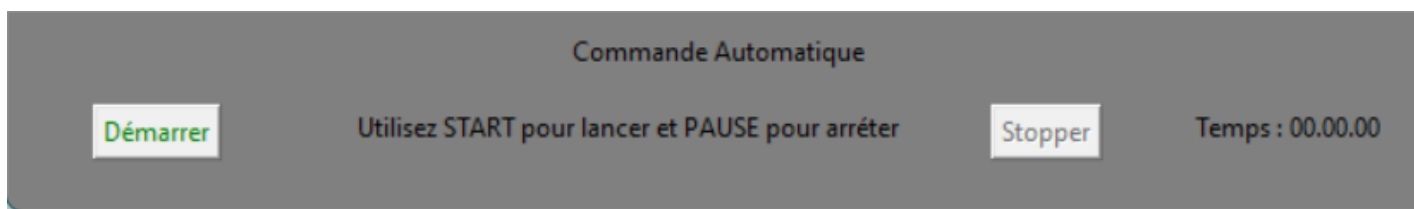


2.4.2 - Le mode Automatique

Il est nécessaire d'être en mode automatique pour utiliser les fonctionnalités manuelles du robot **EDIT > Mode Automatique** et être connecté au robot.

Vous pouvez lancer le mode automatique via le bouton **Démarrer** et le mettre en pause via le bouton **Stopper**. Idéalement toujours stopper le robot avant de changer de mode ou de se déconnecter.

Le bouton stop est également le bouton d'arrêt d'urgence du robot : il stoppe l'action en cours du robot pour éviter tout dommage sur le robot.



2.4.3 - Le mapping

En mode automatique, le robot scanne la pièce et modélise sur l'écran les murs de la pièce. Il est possible d'exporter la salle au format .eps ou bien de nettoyer le scan sur l'écran (effacer les murs actuels) respectivement **SCAN > Exporter Room** et **SCAN > Supprimer Room**

2.5 - Hardware

La première étape pour utiliser le robot est de connecter le cordon pour batterie. Cela va alimenter le contrôleur moteur à partir de la batterie 12V située dans le robot.

Ensuite, si vous souhaitez utiliser le mode automatique positionnez votre robot à 2m du mur de façon à avoir l'avant du robot regardant le mur (perpendiculairement).

Sinon posez juste le robot au sol.

Enfin pour alimenter la carte arduino, les capteurs et la carte bluetooth il faut mettre l'interrupteur situé à l'arrière du robot en position "On"

Utilisez le logiciel fourni pour utiliser correctement le robot.



Cordon batterie



Interrupteur général

3

EXPÉRIENCES PERSONNELLES

BERNAT Loan

J'ai trouvé cette partie du PFR relativement intense. On a peu de temps, le sujet est quand même assez vague et on a une certaine autonomie qui pose problème lorsque l'on rencontre des erreurs.

Hormis cette problématique, je me suis concentré sur la réalisation de l'application ainsi que son design. Après avoir terminé, j'ai apporté mon aide sur la communication Bluetooth et l'optimisation du mode automatique.

Comme durant les premières phases du PFR, j'ai apprécié travailler en groupe et je pense que chacun peut être fier du travail effectué.

GAUDILLAT Eliott

j'ai trouvé cette troisième phase du projet très enrichissante. j'ai pu m'intéresser à la partie hardware et donc progresser dans un domaine que je ne maîtrisais pas. On a également été mis en situation de temps de projet assez court ce qui m'a permis de revoir ma manière de travailler dans certaines conditions.

Pour finir je suis fier du travail réalisé par toute l'équipe durant les 3 phases du PFR.

SOUBIRANT Damien

Me concernant, j'ai bien aimé travailler sur ce projet car il m'a demandé une palette de compétences plus large que les autres phases du PFR. La partie IHM m'a, par exemple, demandé des compétences en programmation python (langage que j'ai du apprendre sur le tas). La partie programmation Arduino mobilisait quant à elle des compétences en langage C++ ainsi que des compétences en SED avec la mise en place de la machine à états pour le mode automatique. Enfin, il y avait la partie hardware qui mobilisait des compétences en électronique. J'ai vraiment apprécié travailler avec les membres de mon équipe car ce sont des personnes efficaces, débrouillardes et avec qui le travail en groupe est plus qu'agréable.

MISSIER Samuel

J'ai personnellement beaucoup apprécié ce projet malgré les difficultés rencontrées. J'ai participé à développer le comportement global du robot, que ce soit en mode automatique ou manuel, et cette étape a été rythmée par de nombreux tests à chaque modification du programme ou du système global. Cette méthode de travail m'a plu et je suis content des résultats que mon groupe et moi avons réussi à obtenir, notre système est fonctionnel et efficace. J'ai pu mettre à profit des compétences développées plus tôt dans mes études mais surtout en construire de nouvelles au sein d'un groupe très efficace et disponible, avec qui le travail est et restera très agréable.

PHAN Quentin

Durant le projet, j'ai eu de nouveau l'occasion de travailler avec ma superbe équipe, le fait de pouvoir travailler sur un projet comme celui-ci m'a permis de mieux appréhender le travail que l'on peut effectuer en entreprise.

J'ai aimé travailler sur ce projet car le fait de faire des recherches sur le matériel et pouvoir par la suite mettre en œuvre les différents composants est pour moi très instructif et intéressant. Pouvoir voir marcher le robot, en ayant participé à la réalisation et à la compréhension du code me permet de voir la multitude de possibilités que l'on peut effectuer dans le monde de la robotique. Je me suis alors rendu compte de la richesse de ma formation et de tout ce qu'elle m'a apportée durant cette année.

BRISSAUD Tobias

De mon côté, j'ai travaillé sur le traitement des données dans le but de développer les fonctions du mode automatique et de la cartographie en collaboration avec Samuel. J'ai découvert le prototypage avec Arduino (C++), sa simplicité et sa complexité à la fois car c'est un système qui est très capricieux. J'ai apprécié cette partie du projet bien que le temps imparti était légèrement court, j'espérais pouvoir communiquer plus d'informations avec l'application et automatiser encore plus le robot pour qu'il soit compatible avec encore plus de situations mais les solutions que nous avons trouvées sont tout à fait satisfaisantes. J'ai personnellement développé mes compétences en traitement des données appliqué à la robotique en pratiquant sur ce projet, cela m'a donné une idée concrète de ce qui était possible de faire, bien que nous ayons travaillé sur du matériel de prototypage. Je souhaite aussi remercier le groupe pour lequel le travail de groupe a été une réussite et ce malgré l'intégration d'un nouveau membre.