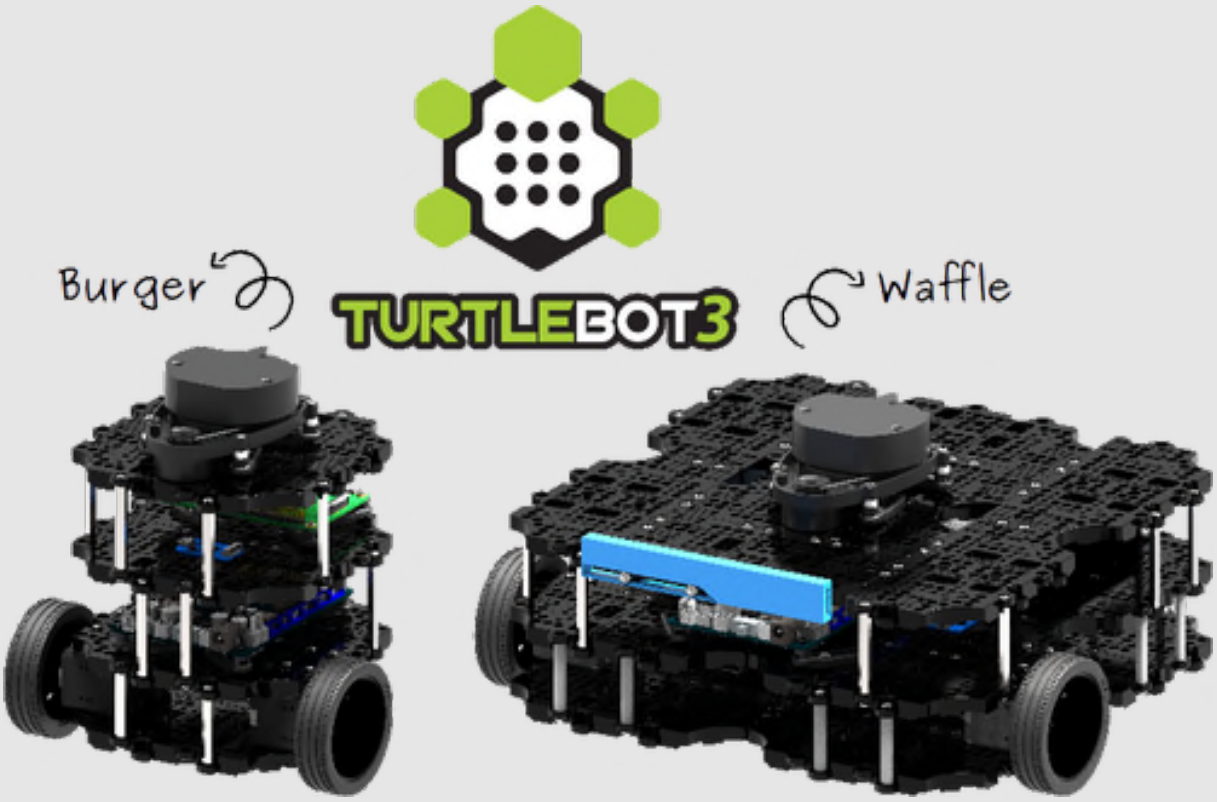
TW53 : TURTLE BOT



Bellini – Cedomirovic - Delorme

IMSI : Printemps 2018

Objectif

Assembler et paramétrer le Turtlebot Waffle afin de réaliser une démonstration vidéo de ces fonctionnalités dans un cadre logistique.

Introduction

Figure 1 : Robot tortue

L'utilisation de robot est de plus en plus importante dans les usines, ils occupent des tâches d'assemblage majoritairement mais participent également de plus en plus en logistique. Parmi ces robots logistiques, les plus courant sont les robots tortus.

Ceux-ci permettent d'acheminer les marchandises au travers des entrepôts et d'aider les ouvriers sur des chaines de montages.

Ces robots sont capables de prendre des éléments et de cartographier un entrepôt ou une usine pour se repérer et accomplir des tâches logistiques.
Ces robots sont équipés de processeur permettant d'éviter les collisions et d'analyser précisément leurs environnements en direct.

Ces robots seront certainement extrêmement présent dans l'usine du futur permettant de soulager les ouvrier de porter des charges lourdes ou d'effectuer des erreurs de trie ou de sélection d'article.

L’Equipe

Florian Bellini

Gaëtan Delorme

Fabian Cedomirovic

Table des matières

Présentation du Turtlebot 3 waffle pi

TurtleBot Waffle pi est un kit de robot personnel équipé d'un logiciel open-source, celui-ci a été créé pour la première version en 2010

Le logiciel intégré ROS offre de nombreuses utilisations de par la possibilité d'ajouter des modules et de réaliser des programmes sur le robot via le Raspeberry pi intégré. Les utilisations de celui sont donc très nombreuse et il peut tout à fait simuler un robot logistique du fait qu'il puisse supporter 30 kilos de charges.

Pour la partie hardware le TurtleBot se compose d'une base mobile offerte grâce à deux servomoteurs reliés à 2 roues motrice, d'un capteur de distance 2D / 3D : le capteur de distance 2D via la caméra et le 3D via le télémètre laser LiDAR.

Le robot se connecte au PC pilote grâce au Bluetooth et au wifi intégré. Il est nécessaire d'installer Ubuntu et différent module ROS sur l'ordinateur pour l'utiliser.

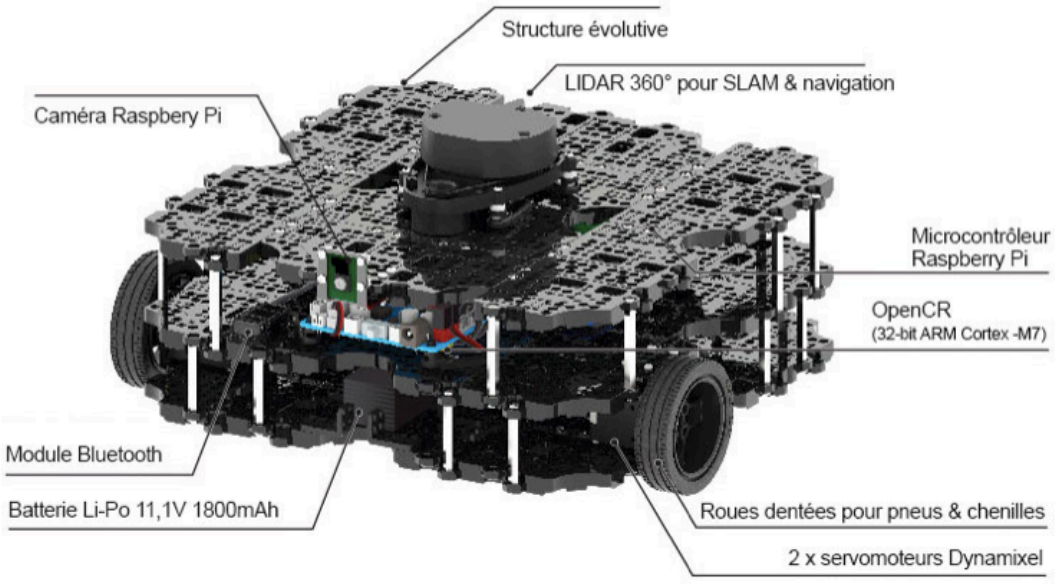


Figure 2 : Composant du TurtleBot

Montage du robot

Pour la première partie du projet, l'objectif était d'assembler le robot Waffle Bot. L'assemblage du robot a été réalisé en suivant le livret d'assemblage fournit, les premières étapes consistait en l'assemblage des couches pastique du robot.

Les secondes se concentraient sur la fixation de ces différentes couches entre elle tout en y ajoutant les composants grâce à des tiges en aluminium.

Enfin la dernière étape était la mise en place des différents composants sur le châssis du robot :

-Raspeberry Pi
-Camera
-Télémètre laser

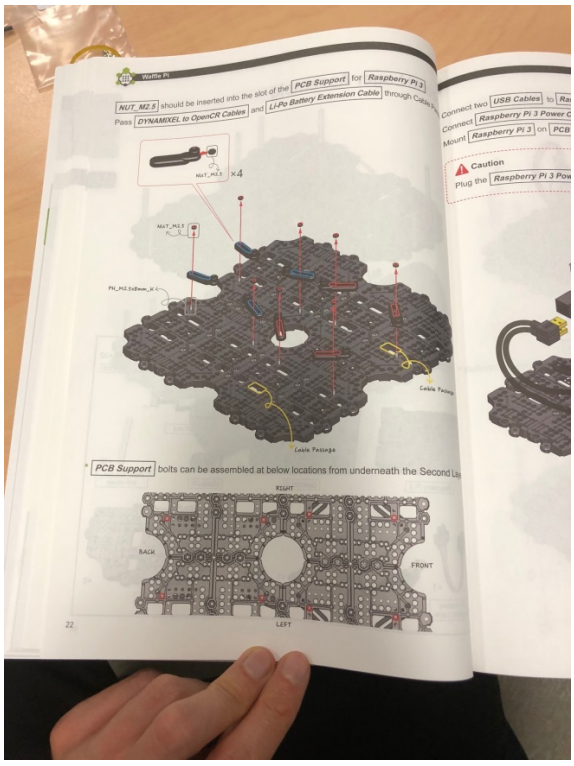


Figure 3 : Plan d'assemblage du robot

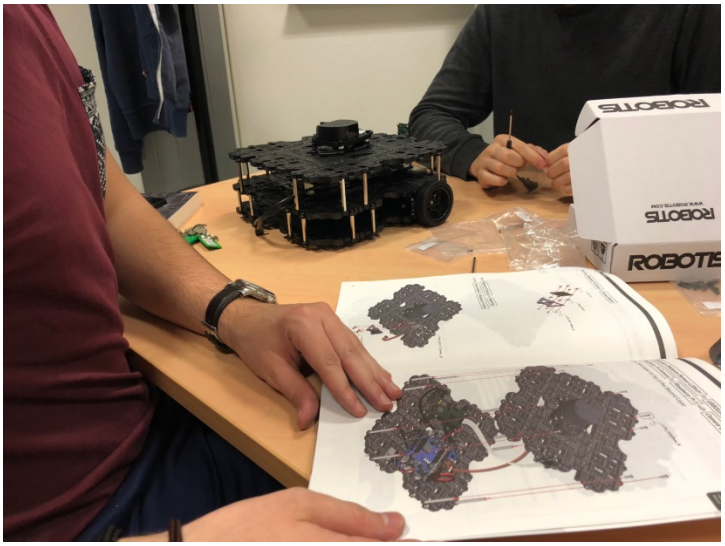


Figure 4 : Assemblage du robot

Pour réaliser l'assemblage, nous avons travaillé à 3. Le kit disposant de 3 tournevis, chacun se concentrait sur un élément du robot que nous avons ensuite mis en commun.

Une fois le robot assemblé nous avons procédé à des tests avec la télécommande pour vérifier le bon fonctionnement des différents composants.

Figure 5 : photo du robot fini

Nous avons ainsi vérifié que le robot roulait correctement, que les batteries et les LEDs étaient connecté etc…

Mise en place du logiciel

Installation d'Ubuntu

L'ensemble des étapes réalisé est présent sur le lien suivant : **http://k6.re/M-g4p**

Le logiciel, ROS, que nous souhaitons installer n'est disponible que sous Linux. Ainsi, afin d'appareillé le PC au robot il est nécessaire dans un premier temps d'installer Ubuntu. Pour cela nous nous sommes d'abord concentrés sur l'installation d'une machine virtuelle au sein d'un PC.

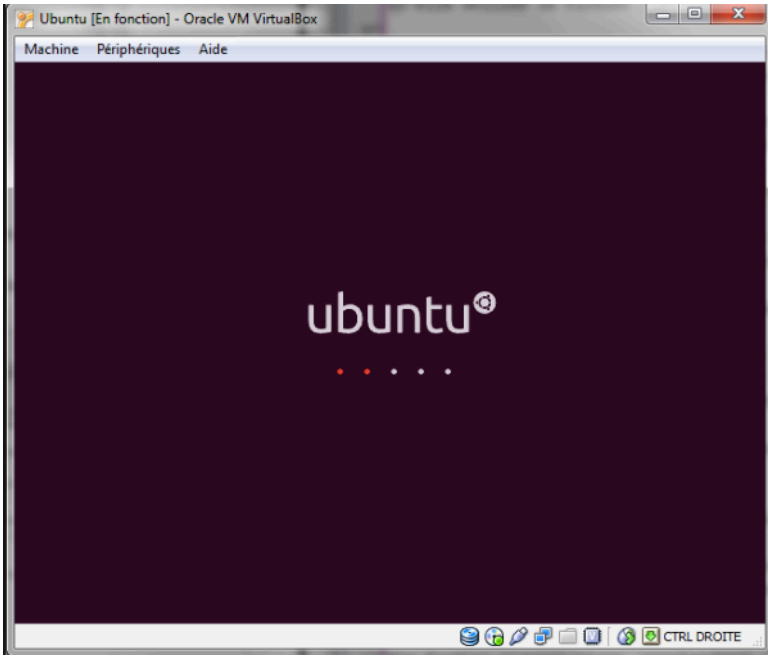


Figure 6 : Installation d'Ubuntu sur machine virtuelle

Nous avons donc installé Oracle sur notre ordinateur. Oracle permet d'allouer de la mémoire vive de notre disque dur dans le but de créer un second ordinateur qui tournerait sous Linux.

Malheureusement après plusieurs essais nous n'avons pas réussi à finaliser l'installation d'Ubuntu sur notre PC.

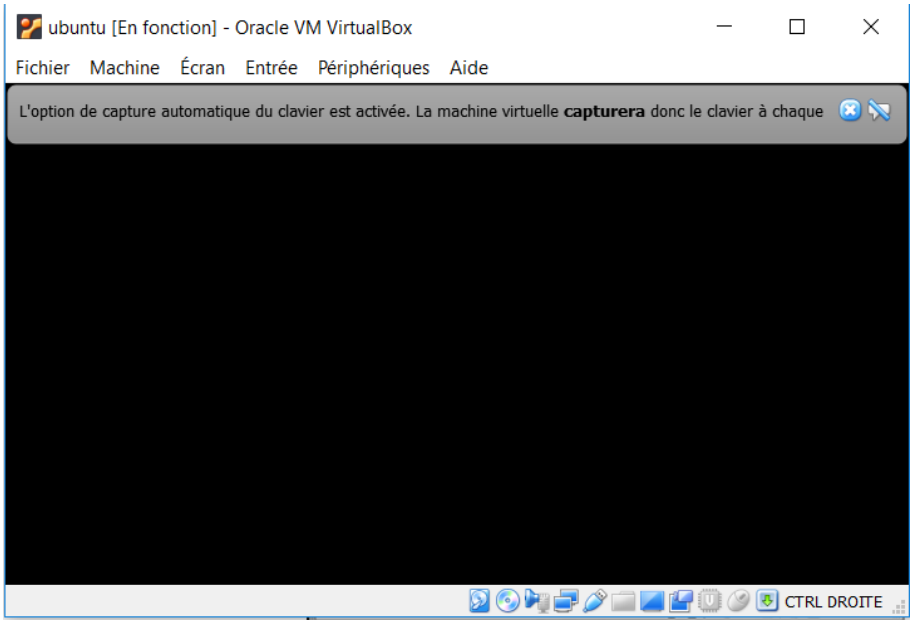


Figure 7 : capture d'écran de l'état de la machine virtuelle

Il nous a donc été nécessaire d'emprunté un PC dédié sur lequel Ubuntu était déjà installé afin de pouvoir installer ROS et le relier à notre Turtle Bot

Installation de ROS

L'installation de ROS a été réaliser en suivant l'ensemble des opérations regroupé sur ce lien : http://k6.re/OzFvc

Pour relier le PC au Robot il a été nécessaire d'installer ROS pour ce faire il a été nécessaire d'aller sur la fenêtre de commande d'Ubuntu que nous ouvrons avec CTRL+ALT+T.
Nous sommes ensuite allés sur le site de :
http://emanual.robotis.com/docs/en/platform/turtlebot3.
Ce site contient l'ensemble des commandes à entrer afin de connecter le robot et le relie

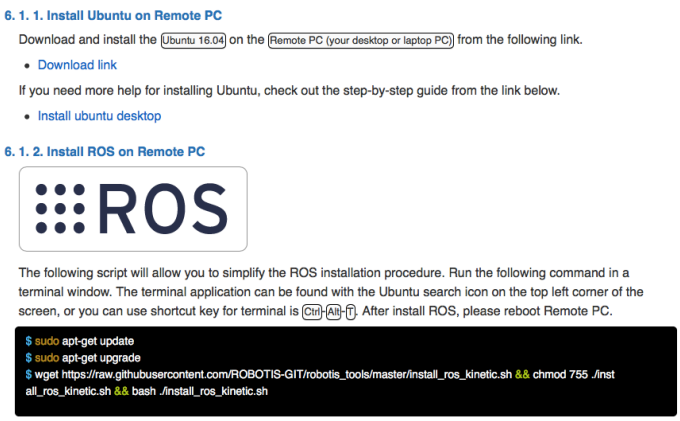


Figure 8 : Site ROS avec les commandes à renter

Premiers essais

Avant de passer à la partie cartographie et afin de mieux appréhender les différentes fonctionnalités du robot, nous avons commencé par quelques « exercices » de prise en main dont voici quelques exemples.

On distinguera 2 types de périphériques :

- ⇨ Le PC télécommande, c'est lui qui va donner les instructions au robot
- ⇨ Le raspberry PI qui va piloter les différents composants du robot ainsi que renvoyer les informations au PC commande.

Ces deux périphériques communiquent entre eux par l'intermédiaire du Wi-Fi, ainsi il faut prêter attention à leurs adresses IP ainsi que leurs fuseaux horaires.
L'adresse IP va nous permettre de savoir à quel service se connecter depuis le PC contrôle.
Le fuseau horaire doit être identique dans un souci de synchronisation entre commande et exécution de la commande.

Une fois ces paramètres vérifiés, on peut passer à la partie commande. Toutes les commandes se feront via l'intermédiaire du terminal. Nous aurons besoins de plusieurs de ces fenêtres, chacune exécutant des opérations différentes.

- ⇨ La première fenêtre nous permettra d'exécuter « Robot Opération System » de son acronyme ROS, qui est le système d'exploitation que nous utilisons pour notre robot.
- ⇨ La seconde fenêtre nous permettre, grâce à l'adresse IP du robot et via la connexion en Wi-Fi d'obtenir la fenêtre de commandes du robot, sur notre ordinateur commande. (de la même manière qu'un informaticien peut se connecter à votre PC à distance à l'aide de quelques informations)
- ⇨ La troisième fenêtre est celle de notre PC contrôle depuis laquelle nous allons renseigner les différents ordres.

De plus, avant de commencer les différents exercices, il faudra renseigner le modèle du robot à notre PC télécommande pour que les commandes émises correspondent bien au type de robot piloté.

La base des différents codes utilisés est directement disponible sur le site (en anglais) de turtlebot3 : https://lc.cx/m6XQ C'est pourquoi, dans un souci de lisibilité nous ne ferons pas de copié/collé des lignes de code dans ce rapport.

Place aux exercices !

Les marqueurs interactifs

Ici l'objectif est de diriger le robot grâce à sa représentation virtuelle. En exécutant les codes correspondants (celui des marqueurs ainsi que de celui de la représentation du robot) on obtiendra le résultat suivant :

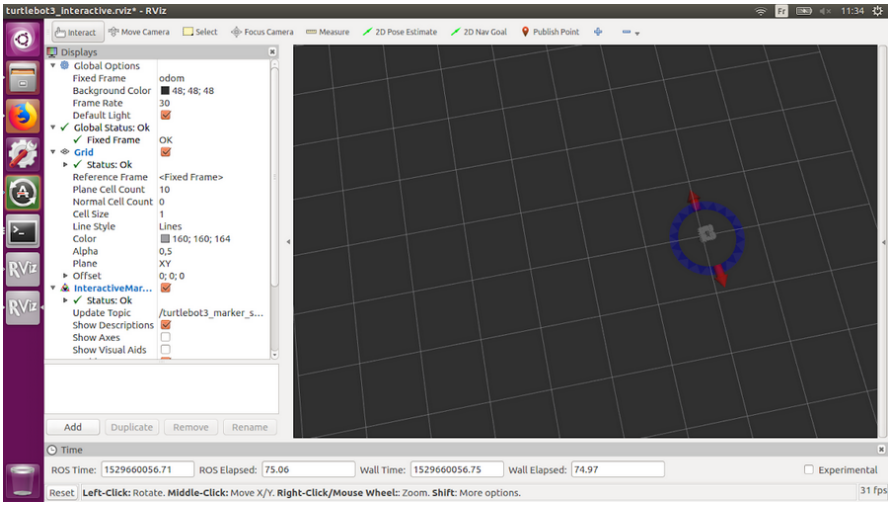


Figure 9 : exercice 1

Sur la partie de gauche se retrouve différentes informations relatives à la représentation.
Sur la partie de droite nous avons la représentation à proprement parlé du robot. A l'aide de la souris en cliquant sur les flèches ou bien la couronne bleu, on va pouvoir en temps réel commander le mouvement du turtlebot.

La détection d'obstacle

Le second exercice est intéressant puisqu'il se rapproche d'un cas réel. Effectivement, bien qu'un robot puisse être programmé pour aller d'un point A à un point B en suivant un parcours prédéfini, il peut s'avérer que se présente sur son trajet des obstacles non prévus/mobiles.

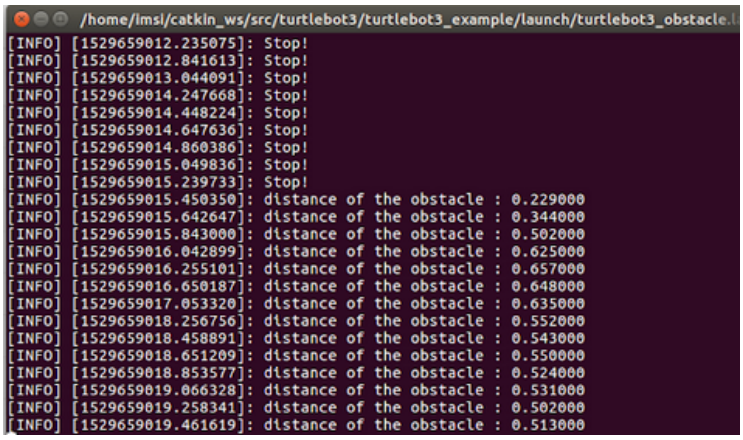


Figure 10 : exercice 2

Cette fenêtre nous retransmet les informations du robot. Ainsi sur la première partie on observe que celui est trop proche d'un obstacle et ce met donc à l'arrêt.
Puis lorsqu'on éloigne, le robot va reprendre son trajet tout en calculant continuellement la distance à laquelle il se trouve de l'obstacle grâce au LIDAR qui vous sera présenté par la suite.

Joint à ce rapport vous trouverez une vidéo que nous avons réalisé qui permettre d'imager notre projet et, en partie, ces exercices.

Cartographie

Principe de base

En robotique, de nombreux moyens peuvent être utilisés afin de se repérer dans l'espace. Ainsi l'odométrie, le principe du « *laser detection and ranging* » ou encore le capteur ccd sont divers moyens qui peuvent être utilisés.

Chacun présentant ses avantages et ses inconvénients, le Turtle Bot est équipé de deux moyens de repérages dans l'espace que sont l'odométrie et le *laser detection and ranging* ou aussi appelé LiDAR.

L'utilisation de ses deux moyens de repérage s'explique par le fait que l'odométrie permet de situer le robot dans un environnement, tandis que le LiDAR permet de repérer l'environnement par rapport au robot.

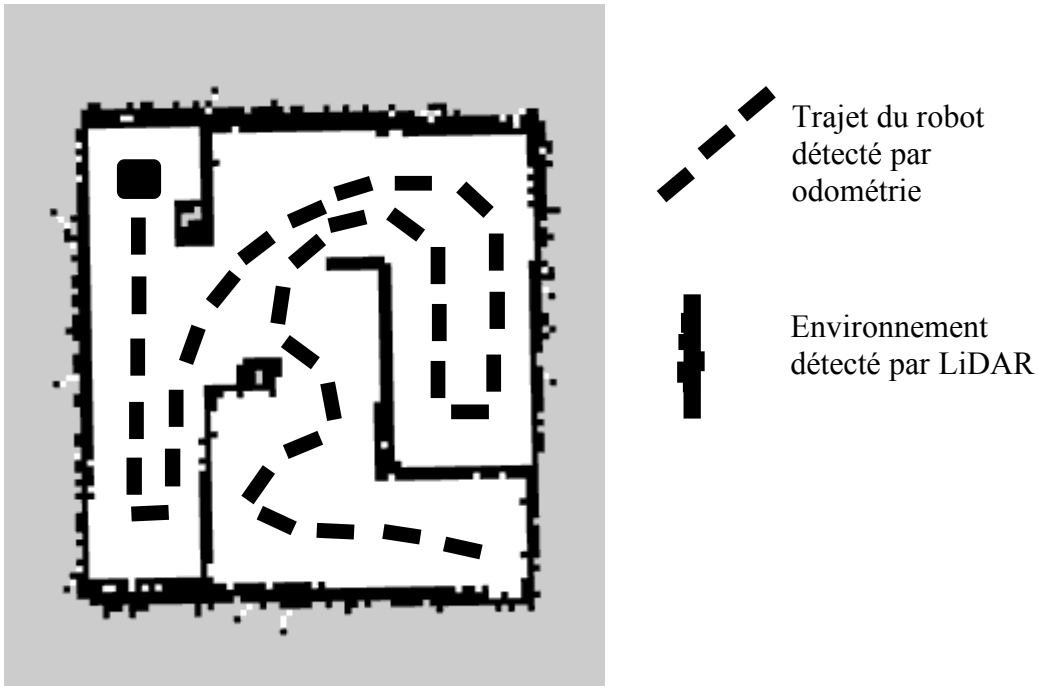


Figure 11 : schéma récapitulant les divers moyens de repérage dans l'espace

A cela sera aussi ajouté comme autre moyen de repérage dans l'espace la caméra afin de détecter un QR code et de pouvoir le suivre. Ce moyen de repérage est utilisé sur les robots suiveurs. Nôtre robot étant en tête de fil, nous ne traiterons donc pas de ce moyen dans notre rapport.

Le LiDAR

La technologie LiDAR utilise le principe de réflexion de la lumière afin de déterminer la géométrie de l'espace autour du robot.

Pour ce faire, un laser émet un faisceau de lumière, dans le cas du Turtle Bot c'est une lumière du domaine infrarouge à 785 nm. La lumière se réfléchira sur l'espace environnant et sera capté par un récepteur photovoltaïque. La distance sera alors calculée en appliquant la formule $d=\frac{v*t}{2}$.

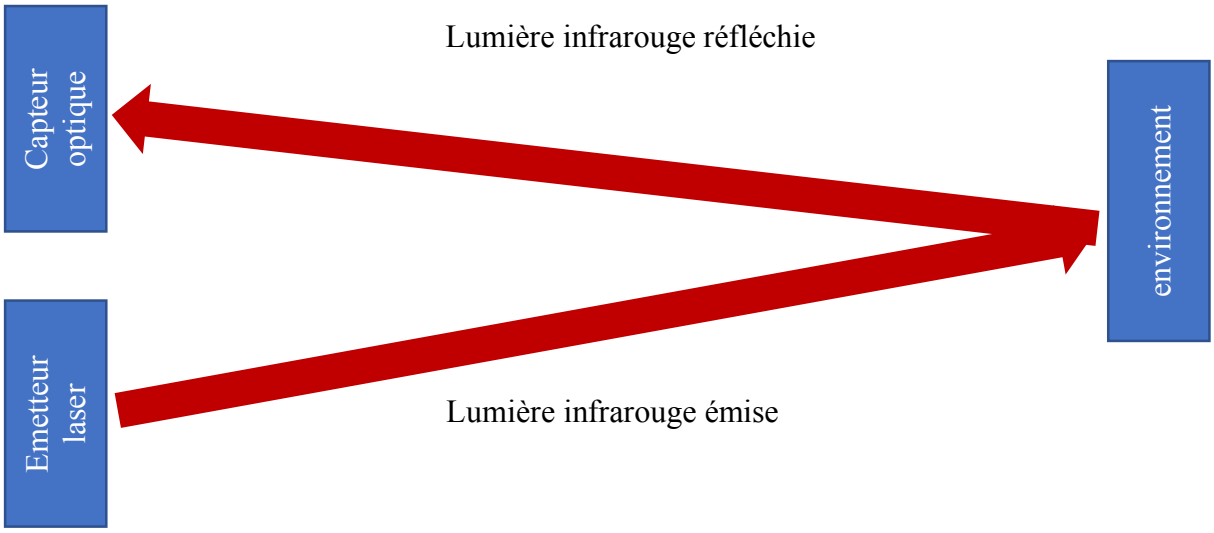


Figure 12 : Schéma explicatif du fonctionnement du LiDAR

En tournant sur lui-même il peut ainsi détecter l'ensemble de l'environnement et sa distance à 360 degrés.

Côté hardware, le LiDAR utilisé est le 360 Laser Distance Sensor LDS-01. Etant fournit avec le robot, il est imposé.

Figure 13 : LiDAR 360 Laser Distance Sensor LDS-01

Ainsi ce LiDAR a certaines spécifications qui vont entrer en jeu lors de la prise en main du robot. Parmi elle, la plus importante est sa plage de détection.

Items	Specifications
Distance Range	120 ~ 3,500mm
Distance Accuracy (120mm ~ 499mm)	±15mm
Distance Accuracy(500mm ~ 3,500mm)	±5.0%
Distance Precision(120mm ~ 499mm)	±10mm
Distance Precision(500mm ~ 3,500mm)	±3.5%
Scan Rate	300±10 rpm
Angular Range	360°
Angular Resolution	1°

Figure 14 : Plage de détection du LiDAR

On voit donc que le robot peut détecter un objet à une distance maximale de 3,5 m. De plus, plus il s'éloigne de l'objet, moins sa perception est précise.

L'odométrie

Le principe de l'odométrie consiste à déterminer la position du robot en fonction du nombre de tour effectué par les moteurs ou par les roues. Ainsi le robot peut déterminer la

distance qu'il parcourt et se repérer par rapport à une position de base. Les deux servomoteurs étant indépendant, il sera également possible de déterminer l'angle d'un virage pour positionner précisément le robot.

Pour faire cela, notre robot utilise des servomoteurs Dynamixel.

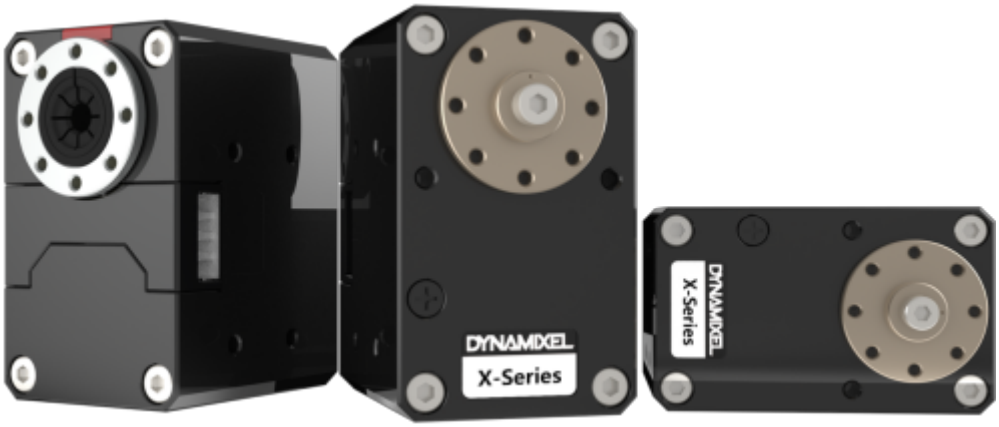


Figure 15 : Servomoteur XM430-W210 utilisé dans le Turtlbot

Ces moteurs ont un degré de précision de 0.088 degrés. Ils sont donc extrêmement précis.

Ainsi la combinaison des deux principes permet de repérer la position du robot ainsi que celle de son environnement et de les afficher sur l'écran de contrôle.

Mise en œuvre

Nous avons réalisé cette cartographie entre le bureau et la salle d'automatique à l'UTBM

Figure 16 : Mur fabriqué pour cartographier la zone

Pour mettre en place notre cartographie nous avons installé les plugins via le site de Turtle Bot.

L'ensemble des opérations réalisé sont situé sur la partie nommé SLAM sur le site de

TurtleBot : http://k6.re/1GwI1

Après installation nous avons lancer la cartographie sur l'ordinateur.

Le robot était ensuite piloté avec la télécommande. Nous avons dû mettre en place des murs pour relier le bureau à la salle d'automatique.

Il a été nécessaire de piloter le robot de manière à ce que celui détecte l'ensemble de son environnement.
Le robot ne repère cependant pas les vitres, il est donc nécessaire de les lui indiquer en plaçant des cartons sur celle-ci.

La grande portée du télémètre permet de balayer largement la zone.
La difficulté était de ne pas taper les murs avec le robot celui-ci identifiant les chocs et s'arrêtant immédiatement de cartographié pour protéger son matériel.
La cartographie du Turtle Bot retire automatiquement les éléments en mouvement de la carte pour ainsi nous pouvons circuler devant le robot lorsque nous cartographions la zone.
Après la cartographie nous obtenons une carte de la zone analysée que le robot pourra exploiter pour lancer des simulations.

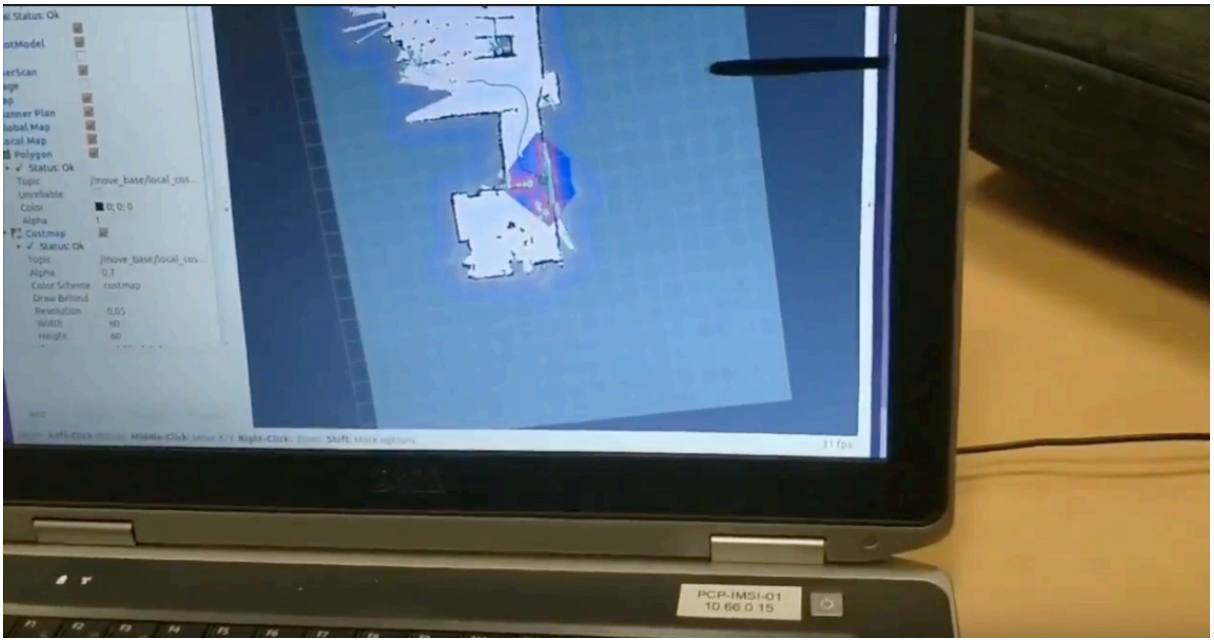


Figure 17 : Le robot détecte son environnement

Mise en place de la démo

Afin de tester notre cartographie et l'autonomie du robot nous avons lancé un test avec celui-ci.

L'ensemble des commandes pour paramétrer se pilotage est regroupé ici :

http://k6.re/XNE10

Le parcours définit était d'aller de la salle des professeurs vers la salle d'automatique. Nous avons lancé donc des tests.
Pour ce faire nous avons défini un parcours sur la carte précédemment cartographié sur l'ordinateur.
Le robot connecté en Bluetooth s'exécute et effectue le tracé comme demandé, en cas d'obstacle il analyse son environnement et calcul automatiquement un autre itinéraire.
Nous avons filmé son parcours et les interactions avec des éléments du décors qui seront placé dans la vidéo de présentation.
En raison du manque de temps nous n'avons pas pu élaborer un scénario avec récupération d'objet par des opérateur par exemple.
Ce scénario aurait été rendu possible du fait que le robot dispose d'une charge maximale de 30 kilos nous permettant de l'utiliser comme un petit AGV.

Conception d’un support pour AR Markers

Une mission secondaire nous a été demandée, celle de réaliser un support facilement retirable pour placer des cartes QR sur le robot.
Les cartes QR permettent aux différents robots de se reconnaitre.
En effet une autre simulation que nous devions réaliser étaient de faire en sorte que les robots puissent se suivrent lors de la réalisation d’un parcours définit.
Cette fonctionnalité a été réalisé mais ne sera pas inclus dans le rapport ni dans la vidéo de ce projet en raison du temps disponible.

La première étape pour la réalisation du support a été une phase de réflexion pour savoir comment allait être conçu notre support.
Nous avons procédé à un brainstorming et l’idée d’utiliser un porte carte nous est apparus intéressante celui-ci pouvant être facilement retirable.

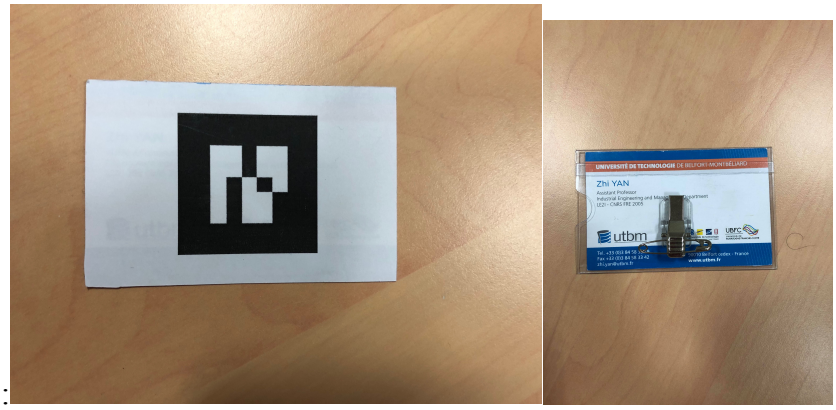
8 : 

Figure 18 : Carte QR code et porte carte

Nous souhaitions pour réaliser ce support qu’il soit utilisable sur les deux robots (Burger et Waffle Pi)
Notre solution permettra une fixation sur les deux robots.

Après discutions nous avons décidé de réaliser notre plaque de fixation en duralium, que nous découperons à la machine jet d’eau.
Cette plaque utilisera les fixations de vis des plaques du robots pour s’attacher.

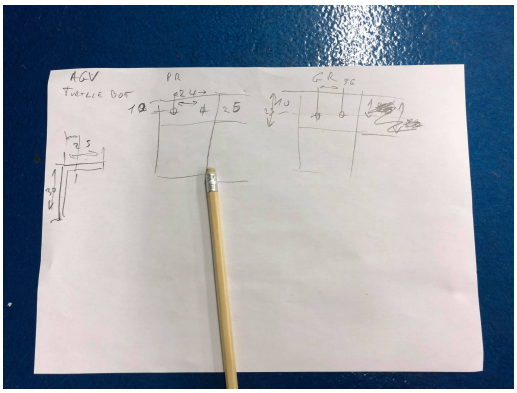


Figure 19 : Schéma de fabrication des plaques de fixations

La pièce est une équerre qui vient se fixer au châssis du robot dont l'un des bords permet la fixation de la pince du porte carte.
Il a également fallu tenir compte des ports du robot pour ne pas empêcher le rechargement ou la connexion de celui-ci à l'ordinateur.

Nous avons donc avec l'aide du technicien tracé le plan sur la machine jet d'eau et définis l'angle d'attaque.

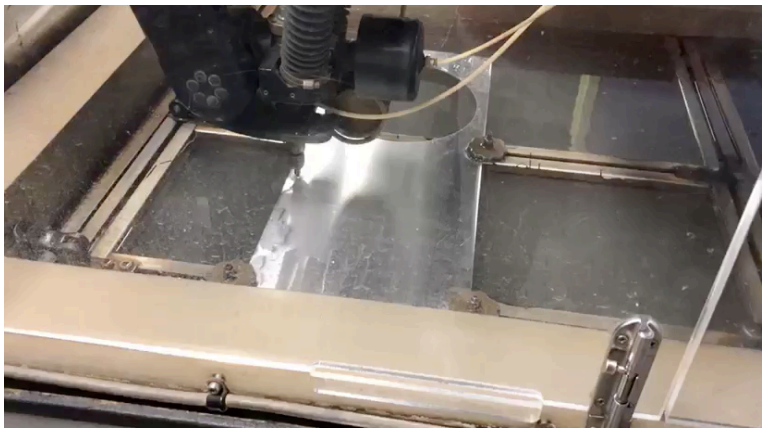


Figure 20 : Découpage de la pièce au jet d'eau.

Nous avons par la suite plié notre pièce avec la plieuse.

Le rendu final est observable sur les photos ci-dessous :

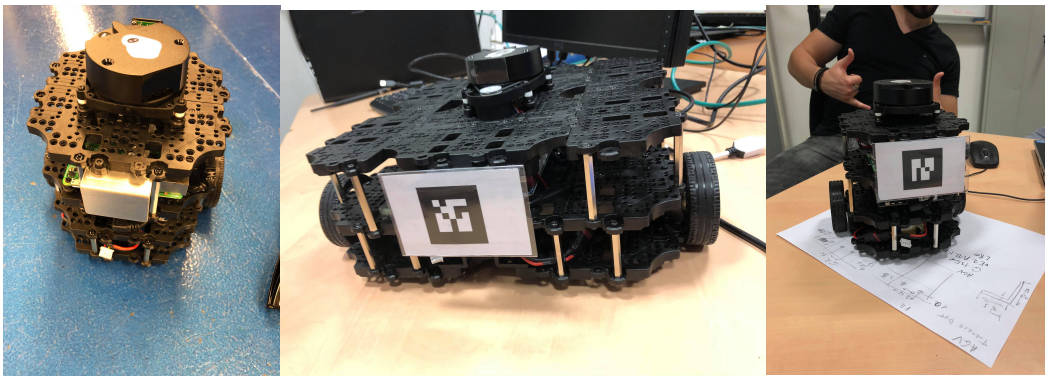


Figure 21 : Support de QRcode

Pour conclure cette étape nous a permis de déployer nos compétences en innovation et fabrication.
Nous avons conçu un support ergonomique et facilement retirable permettant de fixer de manière propre le QR code sur les différents robots.

Conclusion

En conclusion se projet nous a permis de nous familiariser avec des problématiques informatiques.

Nous avons pu découvrir comment fonctionnait un AGV, en fabriquer un et le paramétrer.

Nous avons également pu mettre en pratique nos compétence en fabrication afin de réaliser un support innovant sur le robot.

Enfin nous avons également dû manipuler un logiciel de montage afin de réaliser la vidéo de présentation du projet.