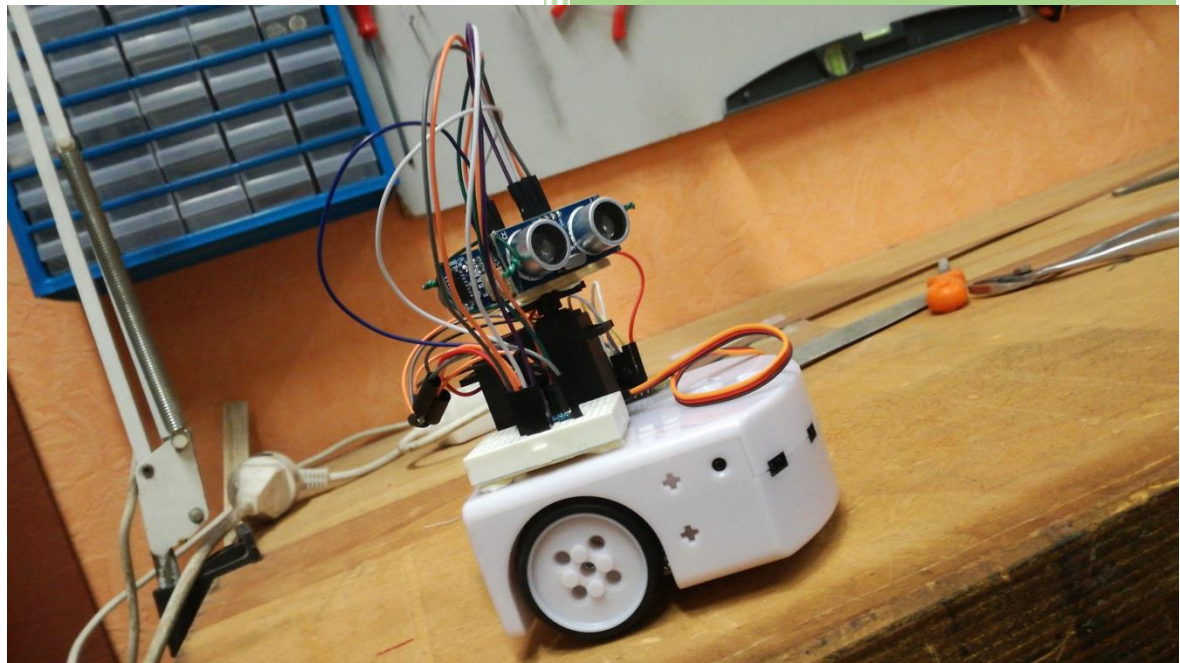


2019

# Conduite et Gestion de Projet



Etudiant :

JELLAD Amin  
LAAROUSSI Maher  
OUFKIR Hamid  
RAHERINIAINA Lucky  
COSTA Arnaud

Client :  
BODINI Olivier

16/04/2019

# TABLE DES MATIERES

1	But du rapport.....	3
2	Description du problème.....	3
2.1	Contraintes.....	3
3	Mise en place des outils .....	4
3.1	Choix des outils.....	4
3.1.1	Cartographier le lieu.....	4
3.1.2	Visualiser le résultat en temps réel.....	4
3.2	Programmer le tout.....	5
3.3	Stocker et partager les résultats.....	5
4	Conception.....	5
4.1	Réflexion .....	5
4.2	gestion du travail.....	5
4.2.1	Calibrage du robot .....	5
4.2.2	Equipe A : déplacement et localisation .....	6
4.2.3	Equipe B : évitement d'obstacle .....	6
4.2.4	Equipe C : capteurs ultrasons, servomoteur et serveur web.....	7
5	difficultés rencontres .....	10
5.1	Equipe A : déplacement et localisation .....	10
5.1.1	Problème de calibrage et de parcours :.....	10
5.1.2	Solution : procéder à un calibrage théorique .....	10
5.2	Equipe B : évitement d'obstacle .....	10
5.2.1	Problème des capteurs de proximité .....	10
5.2.2	Solution : faire une moyenne par distance.....	11
5.3	Equipe C : capteurs ultrasons et servomoteur .....	11
5.3.1	Problème de câblage.....	11
5.3.2	Solution : optimisation .....	12
5.3.3	Problème de composant.....	12
5.3.4	Solution : rachat.....	12
5.4	Problème commun équipe A et C : calcul des distances .....	12
5.4.1	Equipe B : déplacer le robot en utilisant le calcul de l'angle dans un triangle rectangle 12	
5.4.2	Equipe C : dessiner le résultat obtenu.....	13
5.4.3	Solution : erreur de conversion et confusion de calcul.....	13
6	Conclusion.....	13

7	Référence des composants.....	13
---	-------------------------------	----

# 1 BUT DU RAPPORT

---

Vous trouverez dans ce rapport une explication détaillée notre gestion du travail. Le projet en lui-même n'étant pas terminé (ce n'est de toute façon pas le but) nous expliqueront notre méthodologie de travail.

## 2 DESCRIPTION DU PROBLEME

---

Monsieur Bodini, client de ce projet, souhaitait cartographier un lieu avec son robot Thymio. Le client ne nous imposait pas de contraintes particulières. Le Thymio devait pouvoir cartographier un lieu et le client devait pouvoir visualiser le résultat en temps réel. Le client souhaitait également avoir la possibilité d'interagir avec le robot. L'équipe en charge du projet avait l'aval du client pour rajouter des composants (carte Raspberry Pi Zero W, capteur ultrason...).

### 2.1 CONTRAINTES

Nous avons défini un objectif précis étant donné que nous n'avions pas eu de contraintes fixées par le client (à part ne pas casser ni démonter le robot). Notre objectif était de concevoir un algorithme permettant de cartographier un lieu dans un environnement contenant des obstacles à indiquer sur la carte.

Le lieu était défini comme étant plat avec des obstacles géométriques simples. Le lieu était soit rectangulaire ou carré. La définition précise de ce lieu nous permettait de simplifier l'algorithme. L'une des difficultés quand on souhaite cartographier un environnement est l'architecture. Quand elle est simple cela n'est pas un souci mais quand elle devient complexe, il est alors difficile de définir qui est quoi.

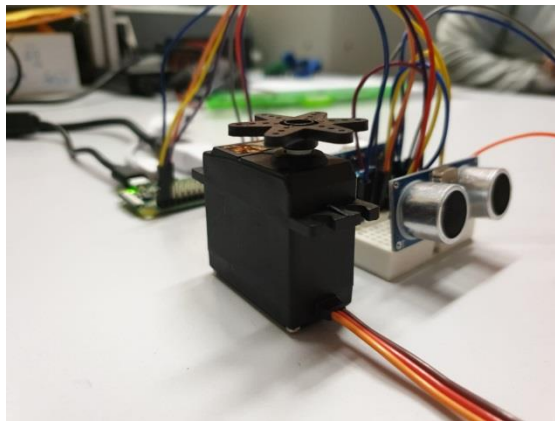
## 3 MISE EN PLACE DES OUTILS

---

### 3.1 CHOIX DES OUTILS

#### 3.1.1 Cartographier le lieu

Pour cartographier un lieu, nous avons décidé d'utiliser deux capteurs ultrasons montés sur un servomoteur. Les capteurs ultrasons ont une portée maximum de 4 mètres.



*Figure 1 le servomoteur avec les capteurs*

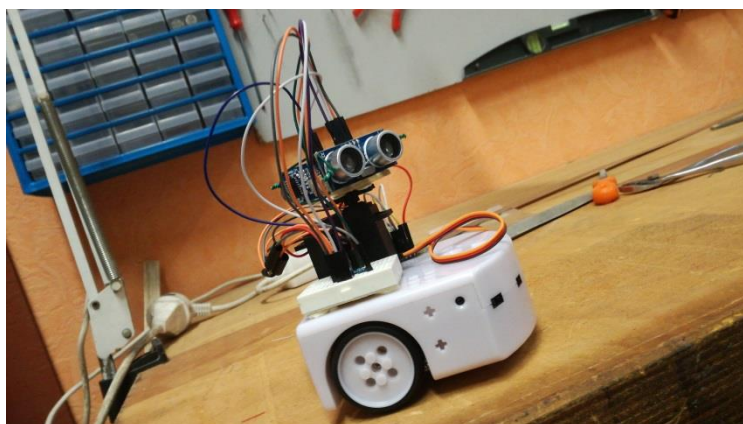
L'objectif étant que le servomoteur effectue une rotation de 180 degrés pour que les deux capteurs puissent analyser le lieu.

#### 3.1.2 Visualiser le résultat en temps réel

Pour visualiser le résultat en temps réel, nous avons utilisé une carte Raspberry Pi avec une interface web hébergée localement et accessible en externe via un hotspot wifi lancé à chaque démarrage.

Avec cela, le client pourra se connecter au serveur avec son Smartphone (ou autre) et peut demander au robot de démarrer la cartographie. Il pourra visualiser en temps réel toujours depuis son support, la construction de la carte.

De plus, la carte Raspberry Pi étant un ordinateur, il aura la responsabilité de diriger le robot ainsi que des équipements que nous lui avons ajouté.



*Figure 2 Montage final*

La référence de chaque composant se situe à la fin de ce rapport.

## 3.2 PROGRAMMER LE TOUT

Le Thymio possède un logiciel permettant de programmer soit par une interface graphique soit par un petit logiciel permettant de faire de petit script. Mais nous avons décidé de ne pas utiliser ces outils et nous nous sommes tournés vers python. La principale raison était qu'avec python nous étions libres sur la manière de programmer mais surtout que c'était plus simple étant donné que la programmation du servomoteur et des capteurs pouvaient se faire par python. Accessoirement, ça permettait aussi d'améliorer nos compétences en python.

L'interface web sera quant à lui programmé en HTML, CSS, jQuery, Ajax et PHP et nous avons utilisé la librairie GD pour la manipulation d'image et de dessin sur l'interface web.

## 3.3 STOCKER ET PARTAGER LES RESULTATS

Nous avons décidé d'utiliser GitLab afin de coordonner nos équipes et organiser nos fichiers et notre planning. De ce fait, nous avons aussi choisi de rendre ce projet public afin que n'importe qui puisse profiter de notre travail.

# 4 CONCEPTION

---

## 4.1 REFLEXION

Une des premières choses que nous avons faites est la décomposition du problème en sous-problèmes plus simple à régler (diviser pour mieux régner), et nous avons discerné trois sous problèmes :

- Le robot doit se déplacer dans l'espace et doit pouvoir se localiser
- Le robot doit pouvoir éviter les obstacles en utilisant ses propres capteurs
- Les capteurs ultrasons doivent pouvoir être synchronisés et fonctionnels et retournent des valeurs sous forme de dessin.

Pour pallier aux problèmes, nous avons divisé notre équipe en trois groupes pour faire le travail en parallèle et ainsi gagner du temps.

Une fois ces trois problèmes résolus, il fallait les rassembler et créer l'algorithme pour déplacer et cartographier.

## 4.2 GESTION DU TRAVAIL

### 4.2.1 Calibrage du robot

Avant de s'attaquer à la partie algorithmique du problème, il nous fallait calibrer les roues du robot et nous avons deux solutions.

Première solution : effectuer une batterie de test et trouver la bonne vitesse pour chaque roue du robot afin que le robot puisse avancer et tourner correctement.

Deuxième solution : une récente mise à jour dans le firmware du robot permet de calibrer les roues sans passer par du code. Il faut juste choisir le mode calibrage et corriger la direction du robot quand il ne va pas droit.

Au final nous avons utilisé les deux solutions car nous nous sommes rendus compte que calibrer le robot via la deuxième solution n'était tout le temps optimale, de ce fait nous avons dû calibrer légèrement la vitesse des roues dans le programme.

#### 4.2.2 Equipe A : déplacement et localisation

L'objectif était de créer des fonctions hauts niveaux pour le déplacement. L'équipe A devait aussi créer des fonctions à la demande des deux autres groupes du projet. L'initiative était que les deux autres groupes ne devaient pas utiliser des fonctions des bibliothèques données par le robot mais bien des fonctions données par cette équipe.

Mais l'équipe A avait un autre objectif : calibrer le robot comme expliqué dans la partie 4.3.

#### 4.2.3 Equipe B : évitement d'obstacle

Avec les capteurs de proximité du robot, l'équipe B devait concevoir un algorithme permettant de contourner un obstacle. Elle avait dans un premier temps effectué une batterie de test sur les capteurs et l'équipe B s'est rendue compte que pour une même distance les valeurs retournées par les capteurs étaient différentes. Pour pallier ce problème, elle avait conçu un graphique résultant de 500 résultats de tests des capteurs pour différentes distances.



Test des capteurs

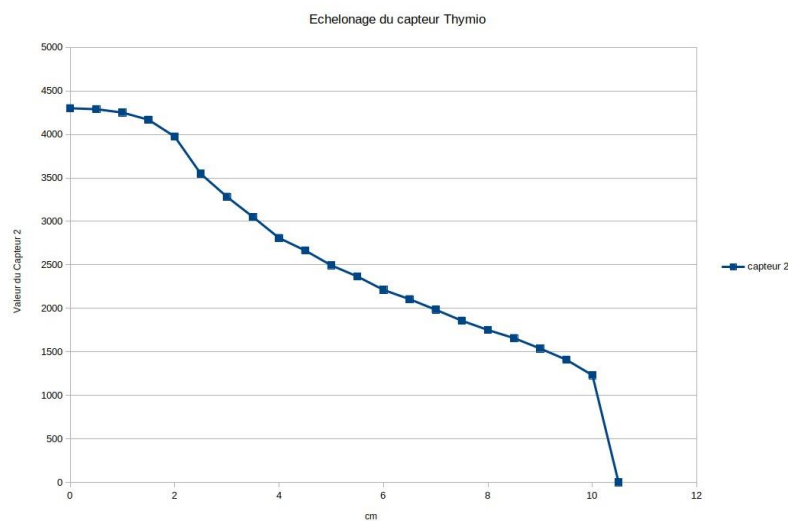


Figure 3 Un exemple d'échelonnage sur l'un des quatre capteurs

Ainsi, L'équipe B avait juste à traduire ce graphe en une structure sous python pour l'utiliser dans la fonction d'évitement.

#### 4.2.4 Equipe C : capteurs ultrasons, servomoteur et serveur web

##### 4.2.4.1 Configuration de la carte Raspberry

L'équipe C a commencé par installer le système d'exploitation Raspbian sur une carte SD de 32 go branché à la carte Raspberry Pi. Elle a ensuite installer les différentes bibliothèques nécessaires au bon fonctionnement de cette dernière. Puis il a fallu faire la création d'un dépôt GitLab et clonage des fichiers dans le Raspberry Pi.

##### 4.2.4.2 Paramétrer les capteurs

Pour paramétrer les capteurs, l'équipe C a utilisé une platine d'expérimentation afin de régler la tension des capteurs. Elle a ensuite fait de premiers tests afin de récupérer les valeurs convertie en cm. Durant cette phase, l'équipe C a remarqué que la précision des capteurs était de 0.1cm

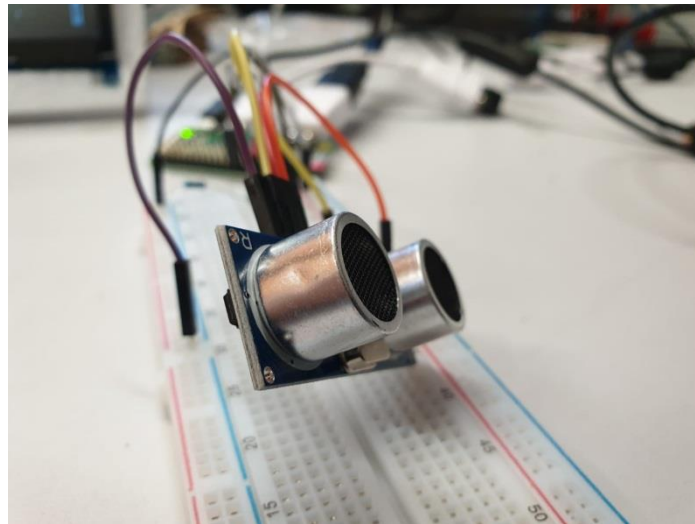


Figure 4 Un capteur sur une platine d'expérimentation

L'optimisation n'étant pas le plus important à ce stage, cette première version était encombrante le temps d'avoir une structure fonctionnelle (cela est expliqué dans la section 5.3).

##### 4.2.4.3 Fusion du servomoteur avec les capteurs

Une fois les capteurs paramétrés ainsi que le servomoteur, l'équipe C a rassemblé le capteur avec le servomoteur.

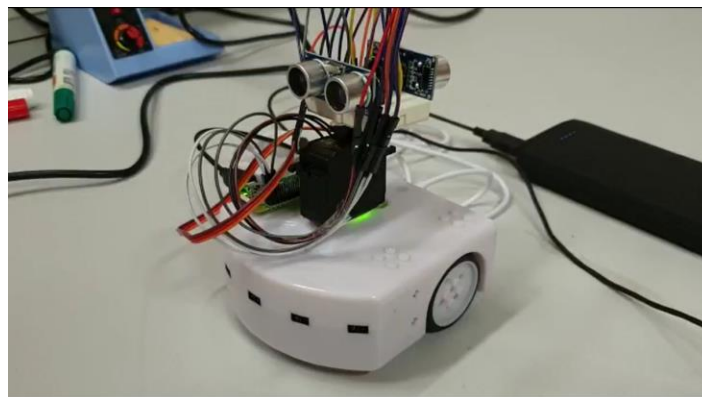


Figure 5 les capteurs montés sur le servomoteur



#### 4.2.4.4 Création des modules pour chaque composant

L'équipe C a fait trois modules :

- Un module qui manipule les capteurs ultrasons
- Un module qui manipule le servomoteur
- Un module qui manipule les deux premiers modules. Son but est de faire un scan : le servomoteur fait une rotation de 180 degrés et les capteurs s'exécutent. Une fois le scan effectué, un algorithme lisse les données obtenues, les corrige et renvoie le résultat.

#### 4.2.4.5 Mise en place d'un hotspot wifi pour l'interface web

Pour cette partie, l'équipe C a installé et configuré une interface web avec apache2 et PHP. L'équipe C a choisi l'IP 10.3.141.3. Puis elle configuré un hotspot wifi pour se connecter à l'interface web. L'équipe C a choisi l'identifiant Mapping et le mot de passe cartographie.

#### 4.2.4.6 Test du scan et du rendu de l'image

Après avoir tout configuré, l'équipe C a décidé de faire de premiers scans. Les résultats obtenus étaient décevants à cause d'un problème de calcul des angles explicité dans la section 5.4. De plus le rendu de l'image était décevant mais après plusieurs tests l'équipe C a pu résoudre ce problème.

#### 4.2.4.7 Mise en place de quatre capteurs

Durant la conception, l'équipe C a réfléchi à un moyen de rendre le scan plus rapide et efficace. Pour cela, elle a décidé de doubler le nombre de capteurs.

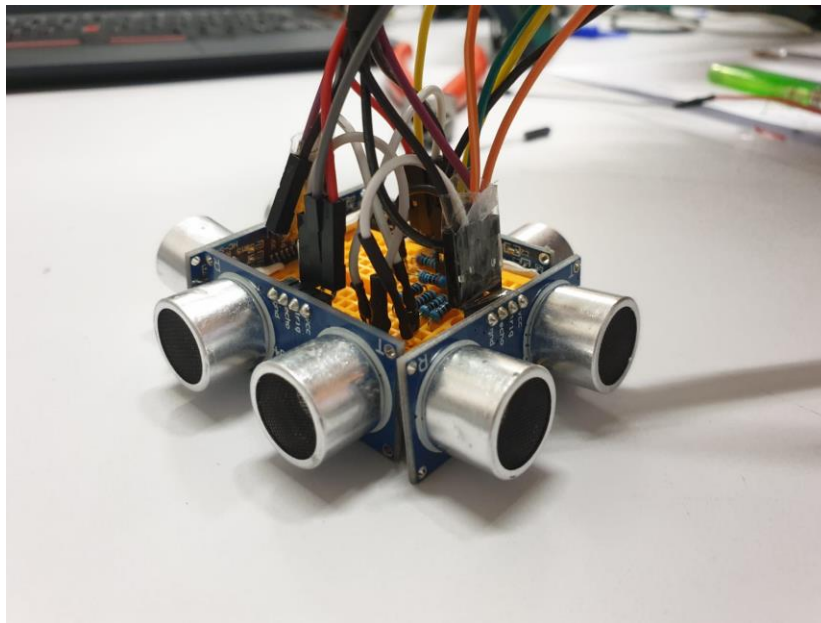


Figure 6 les quatre capteurs

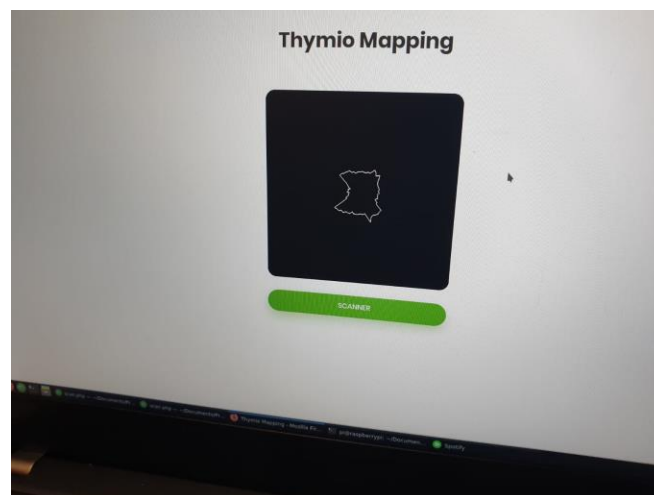
Mais il y avait un problème : cette nouvelle configuration demandait trop de ressource et il fallait une alimentation à la fois pour la carte Raspberry Pi et les capteurs. L'équipe a donc décidé de revenir sur ces pas et de remettre les deux capteurs.

#### 4.2.4.8 Lissage des données pour avoir un résultat proche de la réalité

L'équipe C devait lisser les données de sorte à ce que les formes soit proche de la réalité. Prenons un exemple. L'équipe C a voulu cartographier un carré simple.



Avec l'algorithme de lissage, le résultat obtenu étant ceci :



Ce n'est exactement un carré mais il y a du progrès à faire.

## 5 DIFFICULTES RENCONTRES

---

Durant la phase de conception, chaque équipe a rencontrée des difficultés qui ont pris le plus de temps à se résoudre.

### 5.1 EQUIPE A : DEPLACEMENT ET LOCALISATION

#### 5.1.1 Problème de calibrage et de parcours :

Le fait que le client souhaite cartographier avec un Thymio est à la fois un avantage et un inconvénient. Le Thymio est déjà construit ce qui supprime énormément de travail en robotique mais nous n'avons pas accès aux composants du robot. De ce fait, quand le robot est en action, il ne va jamais exactement à l'endroit prévu. En général, ce qu'on fait pour pallier ce problème, on compte la rotation des roues pour corriger les erreurs de parcours ou bien on utilise un accéléromètre. Le problème c'est qu'on n'a rien qui nous permet de compter et l'accéléromètre du Thymio permet uniquement de calculer sa trajectoire sur une surface 3D. De plus, nous n'avons pas la permission de démonter et modifier le robot.

#### 5.1.2 Solution : procéder à un calibrage théorique

Pour résoudre partiellement le problème, l'équipe A avait fait une calibration théorique du robot. Le robot arrive approximativement à l'endroit prévu et tourne presque correctement. Après une optimisation le robot se déplace et tourne avec une petite marge d'erreur. Mais ce n'est pas assez pour négliger cette marge d'erreur.

### 5.2 EQUIPE B : EVITEMENT D'OBSTACLE

#### 5.2.1 Problème des capteurs de proximité

Le robot possède 5 capteurs lui permettant de détecter des obstacles frontaux. Le problème, c'est qu'avec un faible coût de fabrication, les capteurs ne sont pas assez performants pour donner un même résultat pour une même distance.

### 5.2.2 Solution : faire une moyenne par distance

L'équipe B avait donc décidé de faire une moyenne pour un intervalle de 0.5 cm. L'équipe savait qu'à partir de 10.5 cm le robot ne captait plus rien.

mm	capteur 0	capteur 1	capteur 2	capteur 3	capteur 4
0	4223.333333	4094.666667	4300.111111	4297.111111	4351.555556
0.5	3256.222222	4107.666667	4289.222222	4196.666667	3132.444444
1.0	2950.222222	4048.111111	4250.666667	3907.777778	2773.111111
1.5	2595.777778	3787.666667	4168.0	3470.222222	2620.555556
2.0	2259.222222	3386.111111	3974.777778	3111.555556	2556.444444
2.5	2064.555556	3074.0	3546.777778	2825.888889	2360.0
3.0	1859.888889	2806.888889	3283.0	2707.111111	2236.111111
3.5	1642.666667	2552.666667	3050.333333	2589.888889	2124.444444
4.0	1647.888889	2476.0	2808.666667	2393.111111	1870.0
4.5	1387.666667	2269.222222	2665.222222	2383.0	1783.888889
5.0	1181.888889	2126.222222	2494.666667	2255.888889	1629.777778
5.5	0.0	2016.888889	2366.111111	2024.333333	1383.0
6.0	0.0	1867.777778	2211.888889	1957.777778	1167.222222
6.5	0.0	1757.111111	2105.333333	1772.444444	0.0
7.0	0.0	1620.0	1984.333333	1658.555556	0.0
7.5	0.0	1496.333333	1858.555556	1497.666667	0.0
8.0	0.0	1367.888889	1751.111111	1358.555556	0.0
8.5	0.0	1356.444444	1658.0	864.0	0.0
9.0	0.0	1181.777778	1538.555556	0.0	0.0
9.5	0.0	0.0	1409.666667	0.0	0.0
10.0	0.0	0.0	1231.666667	0.0	0.0
10.5	0.0	0.0	0.0	0.0	0.0

Figure 7 Moyenne des mesures en mm convertie en cm plus tard

## 5.3 EQUIPE C : CAPTEURS ULTRASONS ET SERVOMOTEUR

### 5.3.1 Problème de câblage

L'équipe C avait bien paramétré les capteurs.

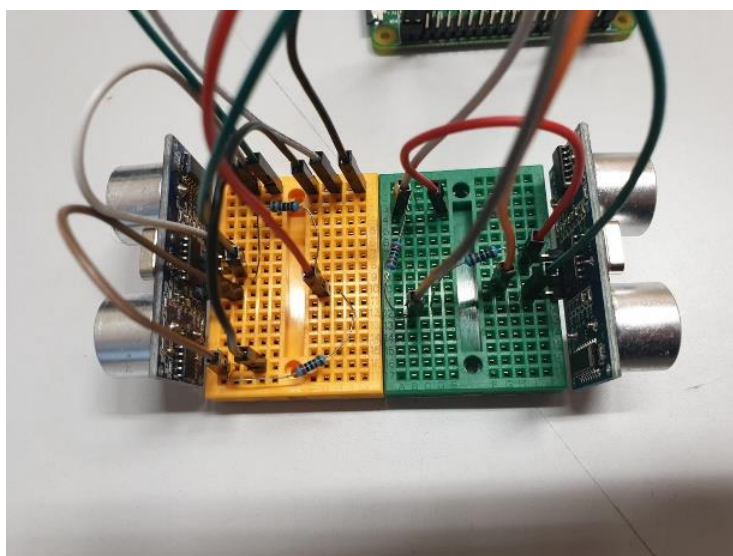


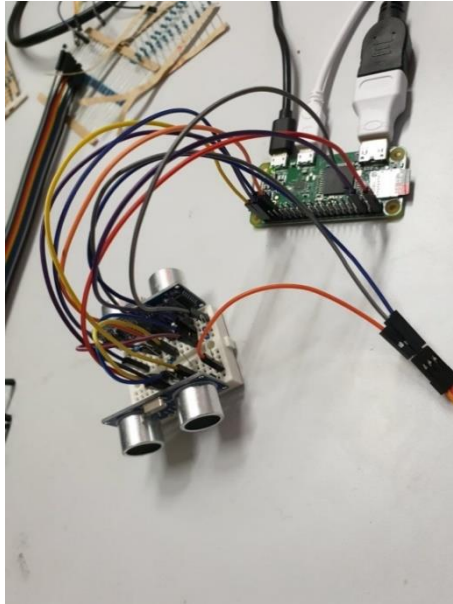
Figure 8 Une première version des capteurs

Le résultat est là mais cela reste problématique pour deux raisons.

- La taille obtenue après configuration est encombrante car l'objectif final est de rassembler le tout sur le Thymio
- La taille des fils est encombrant à la fois pour les capteurs lors de la rotation que dans la configuration finale. De plus les câbles encombraient les capteurs les empêchant de donner de bons résultats.

### 5.3.2 Solution : optimisation

Dans un premier temps, l'équipe a réorganisé la structure pour qu'elle soit plus compacte.



Puis elle a réduit la longueur des fils.

### 5.3.3 Problème de composant

Durant la phase de conception la plupart des composants étaient défaillant. En particulier le servomoteur. Ce dernier ne tournait plus exactement à 180 degrés et avait un dérèglement mécanique ce qui le rend inutilisable pour le scannage du lieu.

### 5.3.4 Solution : rachat

La solution était de racheter le composant.

Malgré cet achat, le deuxième servomoteur n'était pas fonctionnelle (erreur de fabrication) nous avons dû en racheter un troisième (un modèle différent) tout en poursuivant notre projet avec le premier servomoteur défaillant.

## 5.4 PROBLEME COMMUN EQUIPE A ET C : CALCUL DES DISTANCES

### 5.4.1 Equipe B : déplacer le robot en utilisant le calcul de l'angle dans un triangle rectangle

Pour déplacer le robot, l'équipe A a choisi d'utiliser Pythagore pour le déplacement car le déplacement est rapide et optimisé. Le but était de créer une fonction **AllerA** qui permet de déplacer un robot d'un point A à un point B peu importe son positionnement dans l'espace. Pour ce faire l'équipe A a créé un repère propre au robot pour son déplacement et a utilisé le calcul de l'angle pour positionner et déplacer le robot. Mais durant la mise en place de cette idée l'équipe A a eu un souci majeur similaire à l'équipe C.

#### 5.4.2 Equipe C : dessiner le résultat obtenu

L'équipe C devait transformer les données des capteurs en un dessin. Les capteurs s'exécutent à chaque fois que le serveur moteur fait 10 degrés car on souhaite avoir un bon compris entre la vitesse du scan et des données reçues et 10 degrés était un bon choix. La conséquence de cela est qu'il manque des données comme dit précédemment. Pour pallier ce problème il suffit de tracer les droites manquantes en utilisant le même principe que pour l'équipe A

#### 5.4.3 Solution : erreur de conversion et confusion de calcul

Les deux équipes avaient oublié de convertir leurs résultats en degré et l'équipe A s'était trompé dans les calculs mathématiques des angles.

## 6 CONCLUSION

---

Ce fut une expérience intéressante sur plusieurs points.

La matière gestion de projet fait partie des rares matières dans notre enseignement qui soit pratique. Ce n'est pas tant nos connaissances en informatique qui ont été une difficulté pour nous, mais plus notre aptitude à travailler comme si nous travaillions dans le monde professionnel. Il a fallu beaucoup réfléchir sur comment organiser notre travail en analysant la problématique et en déduire une approche satisfaisante pour nous. Il a fallu aussi réagir quand nous avons des imprévus à gérer.

Choisir ce projet c'est aussi apprendre la robotique. Pour certain ce fut une découverte et pour d'autre une révision voire une amélioration de ce qu'il savait déjà. D'ailleurs, parce que nous faisons de la robotique, nous avons pu utiliser les outils et locaux de club de robotique de l'Institut Galilée.

L'épilogue de cette histoire est que ce projet n'est pas fini car d'ici peu il y aura une rencontre de différents clubs de robotique le 10 mai à l'Institut Galilée et nous y sommes conviés. Il sera donc aboutit avant cette date.

## 7 REFERENCE DES COMPOSANTS

---

Servomoteur : SG5010



Deuxième servomoteur : MG996R



Robot : Thymio II



Capteur ultrason : SR04

