

Plateforme rover robotique

Rob Arts

RÉALISÉ PAR :

BELTRAN LUC - BOYEAU BENOÎT - BOUVET MATHIEU - JIMÉNEZ VEGA DIANA -
KHARITONOFF JOSÉPHINE - MERLET THOMAS

ENCADRÉ PAR :

LE KHANH VAN VALÉRY - PREVOT ALAIN - PERINET JEAN-MARC



École Nationale Supérieure d'Arts et Métiers
Campus de Châlons-en-Champagne
Rue Saint Dominique
51000, Châlons-en-Champagne



Sommaire

1	Présentation de l'équipe	2
2	Introduction	3
2.1	Objectifs du projet	3
2.2	Nos motivations	3
2.3	Répartition des tâches	3
2.4	Présentation du robot	3
3	Réalisation de l'aire de jeu	5
3.1	Spécifications fonctionnelles	5
3.2	Maquette numérique	5
3.3	Fabrication	6
3.3.1	Planéité, rugosité et résistance	6
3.3.2	Transport et mise en position	6
3.3.3	Sécurité du robot	7
3.3.4	Accessoires et balises mobiles	7
4	Travail sur le robot	8
4.1	Mise en oeuvre du robot	8
4.1.1	Moteur	8
4.1.2	Boussole	8
4.1.3	Line Follow	9
4.1.4	Capteur à ultrason	9
4.2	Liaison sans fil	10
4.3	Développement d'une application sur mobile	12
4.4	Problématique de localisation	15
4.4.1	Localisation par ultrasons	15
4.4.2	Localisation par infrarouge	16
4.4.3	Localisation à l'aide d'un laser	17
4.4.4	Localisation par triangulation à l'aide d'une caméra	18
5	Financement du projet	20
5.1	Budget	20
5.2	Recherche de sponsors et mécénat	20
5.2.1	Financement	20
5.2.2	Nos prestations	20
5.2.3	Support de communication	20
6	Promotion du projet	21
6.1	Apérobot	21
6.2	Visite d'entreprise	21
7	Travail à réaliser au second semestre	22
7.1	Exploitation du robot	22
7.2	Recherche de financement pour le projet	22
7.3	Promotion du projet	22
8	Annexes	23



1 Présentation de l'équipe

Pour mener à bien le projet que nous détaillerons par la suite, nous avons formé une équipe de 6 étudiants de deuxième année aux Arts et Métiers de Châlons-en-Champagne :

- **Luc BELTRAN**, responsable software et chef de projet.
- **Mathieu BOUVET**, responsable logistique et trésorier.
- **Benoît BOYEAU**, responsable communication et fabrication.
- **Diana JIMÉNEZ VEGA**, responsable des relations extérieures et vice-trésorière.
- **Joséphine KHARITONOFF**, designer et responsable maintenance et gestion de l'environnement du robot.
- **Thomas MERLET**, assistant chef de projet et responsable électronique.



Trois professeurs encadrent ce projet :

- **Valéry LE KHANH VAN**, professeur de Génie Électronique
- **Alain PREVOT**, professeur de Génie Mathématique
- **Jean-Marc PERINET**, professeur de Génie Mécanique

Le groupe de projet est scindé en deux équipes :

- **Équipe A : semestre GIM**
Luc BELTRAN
Mathieu BOUVET
Benoît BOYEAU
Thomas MERLET
- **Équipe B : semestre GIE**
Diana JIMÉNEZ VEGA
Joséphine KHARITONOFF



2 Introduction

2.1 Objectifs du projet

L'objectif global de notre projet est de créer une plateforme de robotique au sein du campus en mettant en place des solutions innovantes et pérennes pour que les étudiants des promotions suivantes puissent reprendre et perfectionner notre travail afin de participer à la Coupe de France de Robotique.

Ainsi, nos premiers objectifs sont :

- La création d'une aire de jeu pour permettre à notre robot de se déplacer en toute sécurité
- Une première exploitation du robot et la recherche de partenaire pour notre projet.

Ces activités seront détaillées par la suite.

2.2 Nos motivations

Nous nous sommes lancés dans ce projet dans l'optique de nous professionnaliser en abordant un sujet qui nous tient à cœur : la robotique. De plus, l'aspect pluridisciplinaire de ce projet nous a énormément motivés. En effet, ce projet permet de traiter parallèlement des domaines tels que l'électronique, la mécanique, l'informatique ou encore les mathématiques, tout en abordant les aspects communication et management aujourd'hui incontournables. Ce projet nous est donc apparu comme un excellent moyen d'élargir et affiner nos connaissances sur le sujet ainsi que d'apporter une certaine plus-value à notre projet professionnel.

2.3 Répartition des tâches

Le projet étant relativement riche et varié, nous avons décidé de réaliser la répartition des tâches de la façon suivante :

- **GIM** : Exploitation du robot, installation logiciels, utilisation des capteurs, développement de la transmission Bluetooth et mode de repérage.
- **GIE** : Maquette numérique de l'aire de jeu, design du logo, dossier et plaquette de sponsoring.

Les tâches suivantes étant vitales pour l'avancement du projet, elles ont été réalisées en collaboration avec les deux groupes :

- Fabrication de l'aire de jeu
- Participation à des événements de promotion
- Recherche d'aide financière ou en nature

2.4 Présentation du robot

Nous utilisons un robot Arduino, robot sur roues programmable et exploitable grâce à une interface de développement fournie (IDE Arduino).



FIGURE 1 – Robot Arduino



Ce robot est composé d'une carte de contrôle, Robot Control Board (étage supérieur) permettant de lire les capteurs et prendre les décisions, ainsi que d'une carte moteur Robot Motor Board (étage inférieur) utile à la commande des moteurs. Les deux tableaux sont équipés de microcontrôleurs Arduino facilement programmables grâce à l'interface de développement prévue à cet effet.

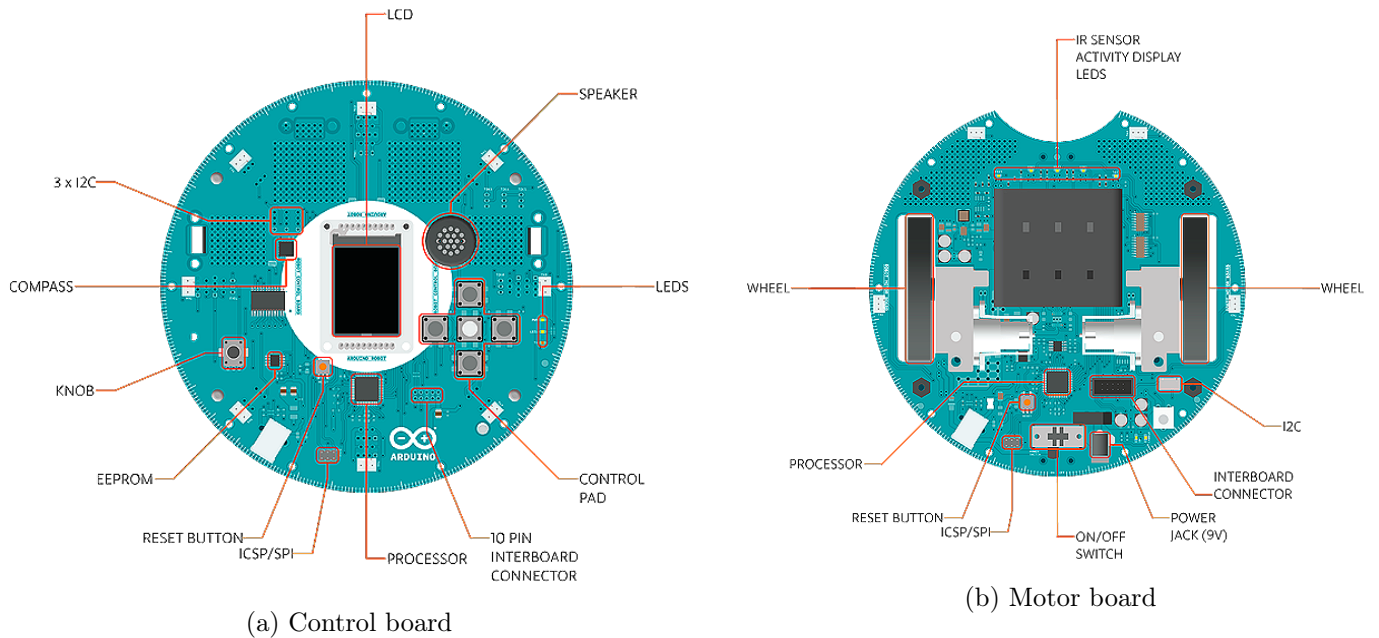


FIGURE 2 – Les deux étages du robot

On retrouve ainsi de nombreux capteurs tels qu'un capteur de proximité, un capteur à ultrason, une boussole, etc. Un écran LCD et un pad permettent d'interagir facilement avec le robot.



3 Réalisation de l'aire de jeu

3.1 Spécifications fonctionnelles

Spécifications	Performance attendu et contrôlé	Solution choisie	Solution technologique
Dimensions	Taille minimum : 2m * 80cm		
Planéité		Planche solide	Portes
Transportabilité	L'aire de jeu doit pouvoir être rangée dans une voiture	Diviser l'aire de jeu en deux parties	2 portes 5 lamellos 2 grenouillères
Résistant au transport	L'aire de jeu ne doit pas être endommagée lors du transport		
Modulable	L'aire de jeu doit permettre la mise en place et le déplacement de capteurs et de balises	Rail	Rail en métal sur le bord de l'aire de jeu
Rugosité	Le robot doit pouvoir rouler sur la surface		
Sécurité du robot	Le robot ne doit pas tomber de l'aire de jeu	Structure autour de l'aire de jeu	

Tableau 1 – Spécifications fonctionnelles

Pour un fonctionnement optimal, l'aire de jeu devra tenir compte des éléments extérieurs suivants : le robot, l'utilisateur, l'environnement lors de son utilisation et lors du transport, les techniciens lors de la maintenance.

Cette plateforme de jeu doit avant tout permettre au robot de se déplacer sans s'endommager. En effet, elle doit empêcher que le robot tombe (si elle est en hauteur) ou se déplace en dehors de la plateforme. Elle doit également résister aux chocs du robot et de l'utilisateur.

De plus, la plateforme doit nous permettre de déterminer la position du robot lors de son déplacement ainsi que de donner des éléments de repère au robot.

Enfin, l'utilisateur doit pouvoir monter et démonter facilement l'aire de jeu afin de faciliter son transport et son installation.

3.2 Maquette numérique

La maquette numérique de l'aire de jeu a été conçue à l'aide du logiciel Catia V5. Seuls le sol et les barrières de la plateforme ainsi que le rail et la mire pour un seul côté de la table, ont été modélisés.

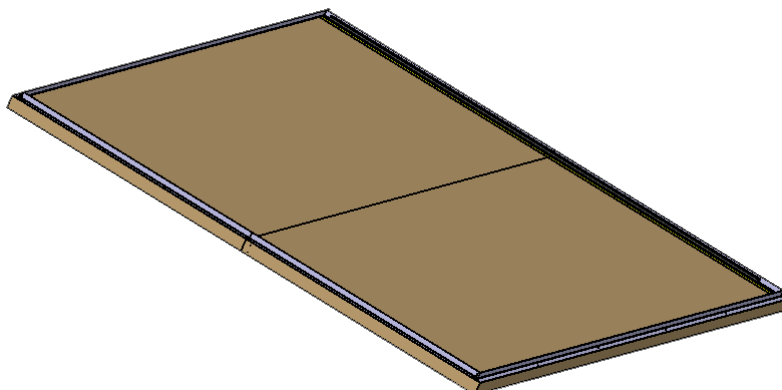


FIGURE 3 – Maquette numérique

Ce modèle a été réalisé en tenant compte du matériel disponible sur le campus (D'autres vues en annexe B).



3.3 Fabrication

Nous sommes rentrés en contact avec le menuisier de l'école afin qu'il nous aide à trouver des solutions qui puissent satisfaire notre cahier des charges.

3.3.1 Planéité, rugosité et résistance

Pour répondre aux spécifications fonctionnelles décrites ci-dessus nous avons choisi de réaliser une aire de jeu à l'aide de deux portes en bois de dimensions ($4 \times 83 \times 204 \text{ cm}^2$). Cette grande épaisseur nous permet d'éviter un moment de flexion trop important sans la présence de traverses (renforts) et ainsi d'assurer la pérennité de l'aire de jeu.

Nous avons décidé d'utiliser une machine à chanfreiner (figure 4) afin de retirer les chanfreins superflus de nos 2 plaques. Nous nous assurons ainsi qu'il n'y a pas d'espace entre les deux plaques lorsqu'elles seront liées.

Pour assurer une bonne mise en position de nos plaques, nous avons choisi d'utiliser des lamellos (figure 5). Il s'agit de petites plaques de bois placées à moitié dans chacun des deux éléments pour assurer le bon placement d'une plaque par rapport à l'autre. Nous avons pour cela utilisé une fraiseuse à lamelles sur un côté de chacune de nos portes puis placé nos lamellos avant d'emboîter nos deux portes.



FIGURE 4 – Machine à chanfreiner



FIGURE 5 – Lamello

3.3.2 Transport et mise en position

Afin d'assurer nos problématiques de transport et de maintien en position, nous avons choisi d'utiliser des grenouillères comme solutions techniques. Comme présenté dans la figure 6, il s'agit d'une attache métallique permettant une attache facile et démontable.



FIGURE 6 – Grenouillère



3.3.3 Sécurité du robot

Afin de s'assurer que le robot ne tombe pas de l'aire de jeu il a fallu réaliser un système de barrière tout autour de celle-ci. Nous avons alors disposé des cornières de protection d'angle le long des bords afin de faire office de barrière. Ces cornières sont en métal afin de résister aux chocs et sont vissées sur l'aire de jeu afin d'être démontable si nécessaire.

3.3.4 Accessoires et balises mobiles

Pour répondre à notre problématique de balises mobiles, nous avons utilisé des rails métalliques que nous avons usinés et vissés le long de notre aire de jeu. Ainsi, les balises peuvent glisser dans le rail.

L'ajout d'une réglette le long de l'aire de jeu a pour but de pouvoir repérer la position approximative du robot rapidement. Elle nous servira aussi par la suite à réaliser des expérimentations.



FIGURE 7 – Usinage des rails



FIGURE 8 – Mire, rails et barrières



4 Travail sur le robot

Fonction	Critère	Niveau
Le robot doit atteindre une position demandée	Se déplacer précisément	Précision de 1cm sur 10cm parcouru
Interface de communication avec le robot	Programmer et accès à distance	Accès à 10m
	Traiter localement puis à distance l'information	Accès à 10m
	Application mobile	Exportable sur Android et IOS
Recueillir des informations sur l'environnement	Stockage de l'information des capteurs	24h d'accessibilité
Communiquer et Promouvoir	Site web-CMS	Accessibilité universelle : responsive web design

Tableau 2 – Extrait du cahier des charges

4.1 Mise en oeuvre du robot

4.1.1 Moteur

On peut commander séparément deux moteurs (droite et gauche) dans les deux sens de rotations. La valeur de la commande pour un moteur est codée sur un octet soit de 0 à 255. Nous avons réalisé des essais pour voir l'influence de la commande sur la vitesse du robot. (cf. Annexe C.1).

Commande	Vitesse moyenne (mm/s)	Écart-type
255	592	2,5
240	529	37,8
230	525	37,8
220	513	1,2
210	504	1,7
200	494	7,4
190	477	7,2
180	459	4,3
170	440	4,8
160	406,5	3,0
150	388	5,8

Tableau 3 – Résultats expérimentaux

Par la suite, on considérera que la vitesse dépend linéairement de la commande (coefficient de corrélation : $R^2 = 0,96$).

4.1.2 Boussole

Le robot possède un module boussole qui lui permet de se repérer dans le plan grâce au champ magnétique terrestre. Ce capteur renvoie une valeur entre 0 et 359 (0 correspondant au Nord géographique). Lors de nos essais, nous avons remarqué que la précision de la boussole n'était pas assez fiable pour l'utiliser comme guide pour le robot. En effet, lorsque le robot n'est pas en mouvement, on observe une variation de l'angle de $\pm 5^\circ$. Et lorsque l'on demande au robot de s'orienter suivant un certain angle, on a une erreur d'orientation d'environ $\pm 20^\circ$.



4.1.3 Line Follow

Il existe un programme prédéfini pour le robot permettant de suivre une ligne ("Line Follow"). Cependant, lors de nos essais, nous avons vu qu'avec ce programme, le suivi n'était pas précis. Nous avons donc apporté des modifications à ce code pour améliorer la précision. (cf. C.3) Pour cela, nous utilisons les 5 capteurs infrarouges implémentés sur le robot. En utilisant les capteurs infrarouges, cela nécessite une ligne noire d'une largeur minimale de 3 cm.

4.1.4 Capteur à ultrason

Les capteurs à ultrasons permettent de mesurer des distances. Le capteur que nous utilisons peut mesurer des distances entre 2 cm et 4 m avec un angle de mesure de 30°.

Dans notre cas, on utilise le capteur pour éviter les obstacles. On a donc besoin que le délai entre deux mesures et de transmission soit très court. Avec ce capteur, le temps de mesure et de transmission est inférieur à 100 ms, ce qui est très satisfaisant.

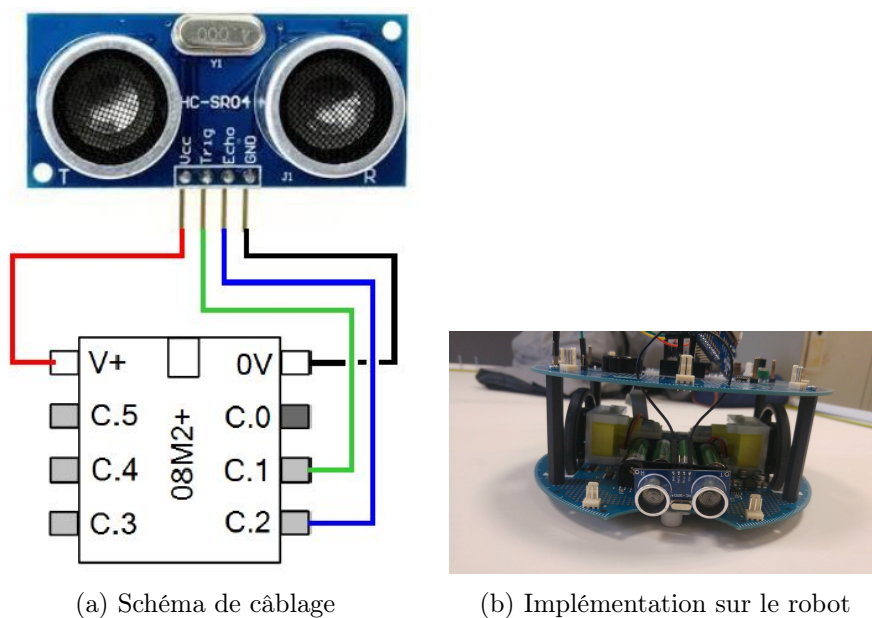


FIGURE 9 – Branchement du capteur à ultrason

En revanche, les mesures du capteur ne sont précises que pour un angle de 15°, ce qui pose un problème majeur puisque lors du déplacement du robot, les obstacles ne seront pas forcément placés devant lui.

Pour pallier à ce défaut, on peut monter le capteur sur un support lui-même monté sur un moteur pas à pas qui permettra de faire un balayage.



4.2 Liaison sans fil

Conformément au cahier des charges, le robot doit pouvoir être programmé et contrôlé à distance à l'aide d'une communication sans fil. Parmi les plusieurs moyens de communication possibles, les communications Bluetooth et Wi-Fi (norme IEEE 802.11) sont envisageables.

	Bluetooth Low Energy (BLE)	Wi-Fi (norme IEEE 802.11)
Portée	≈ 10 m	≈ 30 m en intérieur ≈ 100 m en extérieur
Fréquence d'émission (porteuse)	2,4 GHz	2,4 - 3,6 - 5 GHz
Débits de données	10 Mbit/s mesuré (24 Mbit/s théorique)	30 Mbit/s (54 Mbit/s théoriques)

Tableau 4 – Comparatif Bluetooth Low Energy - WiFi

C'est finalement la communication Bluetooth qui a été retenue, sa portée étant suffisante pour une utilisation dans la même pièce. Le Bluetooth a une bande passante plus limitée que le Wi-Fi, mais qui est amplement suffisante s'il s'agit seulement de contrôler le robot à distance. La communication Wifi serait pertinente uniquement si l'on souhaite envoyer l'image de la caméra vers le téléphone, ce qui n'est pas le cas ici.

La technologie Bluetooth Bluetooth®

Le Bluetooth est un standard de communication qui utilise les ondes radio Ultra Haute Fréquence (UHF) sur une bande de fréquence de 2,402 GHz - 2,48 GHz, sur 79 canaux chaque canal étant séparé d'1 MHz.

La technologie Bluetooth Low Energy (BLE)

Le BLE, ou Bluetooth à basse consommation, est un standard ouvert basé sur le Bluetooth 4.0, qu'il complète mais sans le remplacer. Il permet une consommation d'énergie 10 fois moindre, pour un débit identique au Bluetooth classique (1 Mb/s).

Avec la technologie Bluetooth Low Energy, un appareil Bluetooth Maître (dans notre cas le téléphone portable) va établir un lien avec un appareil Esclave (le robot).

Le BLE utilise des profils bluetooth spécifique appelé Generic Attribute ou GATT. Ces profils se divisent en trois niveau : Profil, Service et Caractéristique (Figure 10).

Un profil est un ensemble de Services prédéfini, contenant eux-mêmes un ensemble de Caractéristiques. La valeur incluse dans le Caractéristiques est la valeur que le client cherche en pratique à récupérer, comme par exemple la position du robot sur l'aire de déplacement. Lors de la connexion à un périphérique, il est nécessaire de sélectionner le Service qui sera utilisé lors de l'échange de données.

L'UUID : L'Universal Unique Identifier (UUID), codé sur 128 bits, permet d'identifier de manière unique le périphérique Bluetooth.

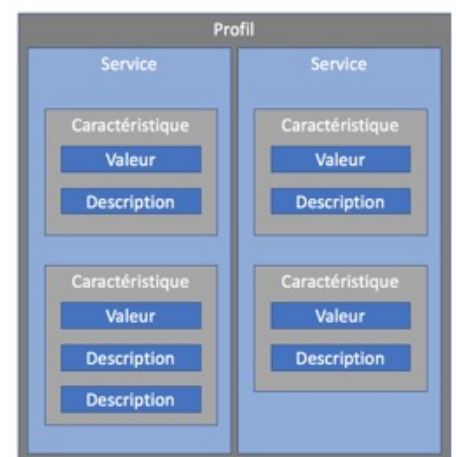


FIGURE 10 – Profils



Connexion à l'Arduino Control Board : La carte du robot n'étant pas équipée nativement du Bluetooth, il est nécessaire d'y connecter un module supportant ce protocole. Nous disposons du module Adafruit Bluefruit LE Shield. Le facteur de forme de cette carte fait que le Shield recouvre intégralement une carte Arduino Uno R3, et n'est donc pas adaptée en première approche à l'Arduino Control Board. Il a donc été nécessaire d'adapter la connexion. Le shield utilise une liaison de type SPI (Serial Peripheral Interface) afin de communiquer avec la carte. Le bus SPI utilise 4 signaux logique :

- **SCLK** — Serial Clock, Horloge (généré par le maître)
- **MOSI** — Master Output, Slave Input (généré par le maître)
- **MISO** — Master Input, Slave Output (généré par l'esclave)
- **SS** — Slave Select, indique quel carte est utilisé sur le bus SPI
- **IRQ** — Pin d'interruption qui indique quand une donnée est reçue par la carte Bluetooth

La connexion entre le robot et la carte Bluetooth se fait de la manière suivante :

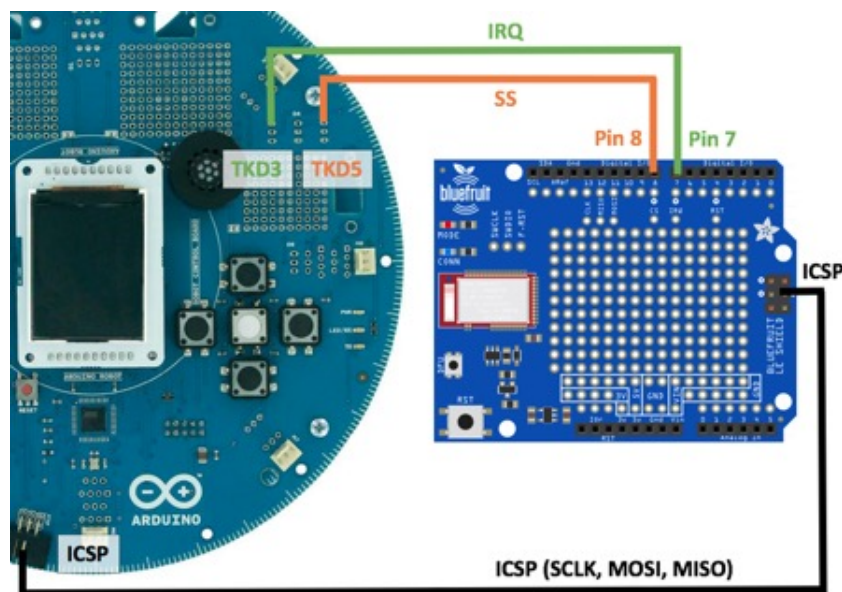


FIGURE 11 – Connexion entre le robot et la carte Bluetooth

Nous utilisons les entrées numérique TKD3 et TKD5 de la carte du Robot pour les signaux logique IRQ et SS.



4.3 Développement d'une application sur mobile

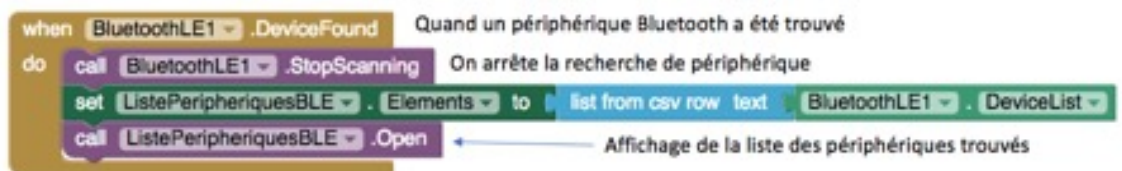
Afin de pouvoir contrôler le robot à distance et échanger des informations avec celui-ci, nous avons développé une application mobile. On utilise pour cela l'outil MIT App Inventor, développé par le Massachusetts Institute of Technology (MIT), qui simplifie le développement des applications sous le système d'exploitation Android.



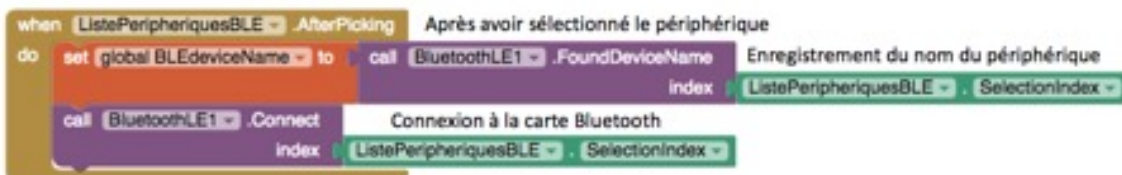
FIGURE 12 – Application sur App Inventor

Cette application permet pour l'instant de contrôler le déplacement du robot dans les 4 directions. A l'avenir, elle pourra recevoir les informations du robot (localisation, capteurs...).

Connexion à la carte Bluetooth :



(a) Scan des périphériques Bluetooth



(b) Selection et connexion du périphérique

FIGURE 13 – Connexion



Sélection du service : Après sélection de l'appareil dans la liste des périphériques Bluetooth, il est nécessaire de sélectionner le ServiceService. Dans le cas de notre carte Bluetooth, un Service particulier est créé par le programme Arduino, il s'agit du "Custom Service".



FIGURE 14 – Sélection du service

Boutons de déplacement : Les 4 boutons de déplacement sont programmés comme tel :



FIGURE 15 – Boutons de déplacement

Envoi de données sur la carte Bluetooth : La fonction envoiDirection est appelée par les actions sur les boutons de déplacement. Afin d'envoyer des données sur la carte Bluetooth, on utilise la fonction WriteStrings de la classe BluetoothLE. Elle prend pour argument :

- serviceUuid : l'UUID du Service (Universal Unique Identifier) enregistré préalablement dans l'Index sélectionné parmi la liste des Services afin de connaître le périphérique auquel envoyer la donnée.
- characteristicsUuid : l'adresse de la Caractéristique de Service sélectionnée.
- values : la valeur à envoyer

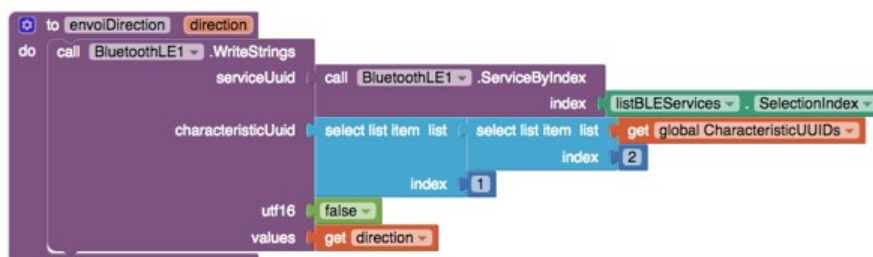


FIGURE 16 – Envoi de données sur la carte Bluetooth



Programmation du robot : Le robot fonctionne avec une carte Arduino, programmable à l'aide du logiciel Arduino . Nous utilisons la librairie fournie par fabricant de la carte (Adafruit) qui comprend l'ensemble des fonctions de la carte Bluetooth. Cette librairie étant conçue pour fonctionner avec une carte Arduino Uno, il a été nécessaire de l'adapter pour le robot. En effet, l'Arduino Control Board ne supporte pas la fonctionnalité dite de Software SPI, permettant de simuler une interface de connexion SPI à partir de pin digitaux de la carte. Il est donc nécessaire de supprimer ces fonctions de la librairie, afin que le programme compile.

Notre programme est structuré en différentes fonctions, en voici les principaux extraits :

— La fonction setup, appelée au démarrage du robot :

```

1 void setup(void)
2 { Robot.begin(); //Démarrage du Robot
3   pinMode(LED_PIN, OUTPUT); //Definition du pin de la LED d'activite du robot
4   ble.begin(); // Démarrage du module Bluetooth

```

— La fonction loop, exécutée en boucle lors du fonctionnement du robot :

```

1 void loop(void)
2 {digitalWrite(LED_PIN, HIGH); //On allume la LED d'activite, la connexion est
   etablie et le robot peut bouger
3   ble.println("AT+BLEUARTRX");
4   ble.readline(); //Verification de la reception de chaine de caractere de la carte
   Bluetooth
5   ble.waitForOK();
6   String BLEbuffer = ble.buffer; //Creation d'une variable BLEBuffer contenant la
   chaine de caractere recue
7   //Pour test : Si une donnee a ete recue, on l'affiche dans le Moniteur Serie
   visible quand l'Arduino est connecte a un ordinateur
8   if (BLEbuffer.length())
9     Serial.print(F("[Recu]_")); Serial.println(BLEbuffer);
10  //Si la commande recue
11  if (BLEbuffer.indexOf("avancer") >= 0) {
12    //Robot.motorsWrite(vitesseGauche,vitesseDroit) controle la vitesse des moteurs
   gauche et droit du robot.
13    //Les valeurs vitesseGauche et vitesseDroit sont comprises entre 0 (arret) et
   255 (vitesse maximale).
14    //Modulation a Largeur d'Impulsion (MLI). La PWM est en fait un signal numerique
   qui, a une frequence donnee, a un rapport cyclique qui change.
15    Robot.motorsWrite(130, 130); //Le robot avance vers l'avant, meme vitesse des 2
   moteurs
16    //Pour test : On affiche dans le moniteur Serie que la commande a bien ete recue
17    Serial.println(F("Commande_recue:_Avancer"));
18    ble.println("AT+BLEUARTRX");
19  }
20  //On fait de meme pour les autres commandes
21  BLEbuffer = ""; //On vide le Buffer en attente d'une nouvelle donnee
22 }

```

Ce programme est téléversé sur la carte du robot. Lorsque le robot est mis sous tension, la carte Bluetooth démarre et il suffit de lancer l'application depuis un téléphone mobile pour s'y connecter. Les commandes de déplacements fonctionnent correctement et permettent de déplacer le robot sur l'aire.

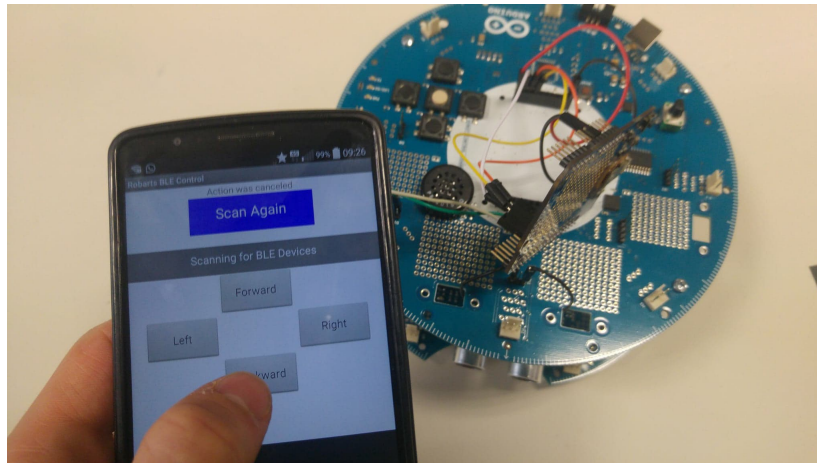


FIGURE 17 – Contrôle à distance du robot

4.4 Problématique de localisation

La problématique de localisation a été un des problèmes majeurs sur lesquels nous avons dû nous focaliser. Plusieurs choix technologiques s'offrent à nous :

4.4.1 Localisation par ultrasons

Le principe consiste à utiliser la technologie des ultrasons afin de calculer la distance entre le robot et un obstacle situé face à lui. Cependant, cette solution nécessite plusieurs capteurs pour pouvoir se repérer. En couplant l'utilisation de la boussole (pour s'orienter dans le sens de la table) et les ultrasons pour déterminer la distance avec les barrières, le robot aurait pu être capable de se repérer.

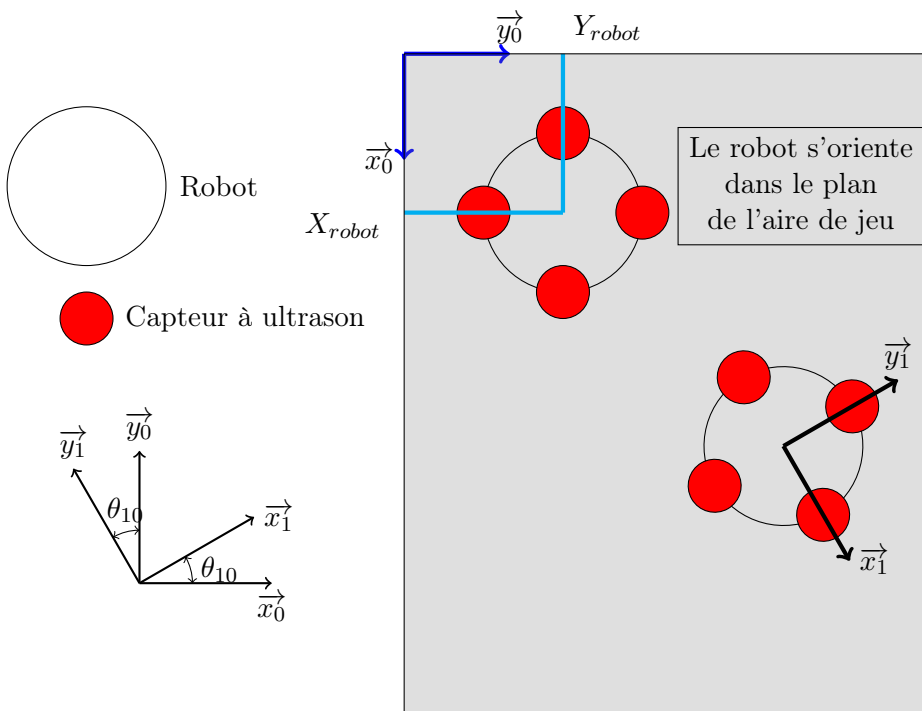


FIGURE 18 – Localisation par ultrason

Malheureusement, les tests et essais réalisés sur les capteurs n'ont pas été concluants. Le capteur à ultrasons



était peu précis et ne permettait pas d'obtenir une position correcte du robot. De plus la boussole gyromètre du robot est aussi peu précise (précision de l'ordre de 10° au vu des trajectoires du robot). Nous avons donc décidé de ne pas utiliser ce capteur sans réaliser une étude plus approfondie.

4.4.2 Localisation par infrarouge

Le principe consiste à placer des balises équipées d'émetteur autour de l'aire de jeu. Le robot, alors équipé de plusieurs récepteurs IR placés à 360° sur son bord, est capable de déterminer la disposition des balises en fonction du récepteur utilisé.

Cependant cette solution requiert de nombreux récepteurs IR pour obtenir une précision correcte. Le nombre de pin utilisables pour le fonctionnement de ces récepteurs étant limité, nous avons laissé cette solution de côté.

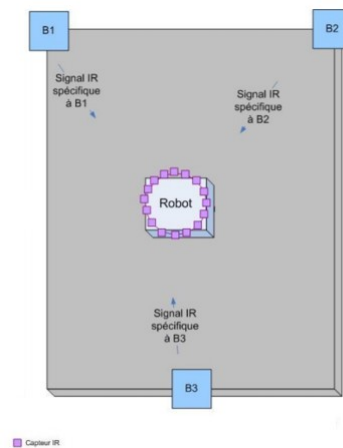


FIGURE 19 – Localisation par infrarouge



4.4.3 Localisation à l'aide d'un laser

Le principe consiste à faire tourner un laser au-dessus du robot de façon horizontale, à une vitesse connue. Les balises placées autour de l'aire de jeu, alors équipées de photodiodes adaptées au laser, renvoient l'information du passage du laser au système de localisation. En couplant un module d'horloge avec le capteur laser, il est possible de connaître le moment où a été reçu le signal. En renvoyant cette information à l'aide d'un module Bluetooth, le robot peut analyser l'information.



FIGURE 20 – Capteur laser

On en déduit ainsi les angles entre les balises qui nous permettent de déterminer la position du robot à l'aide du principe de triangulation (cf. Annexe D).

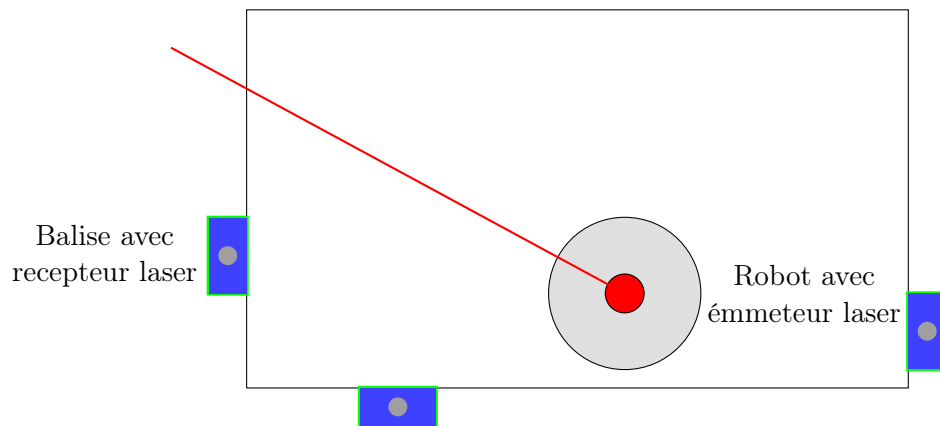


FIGURE 21 – Configuration - système de localisation par laser

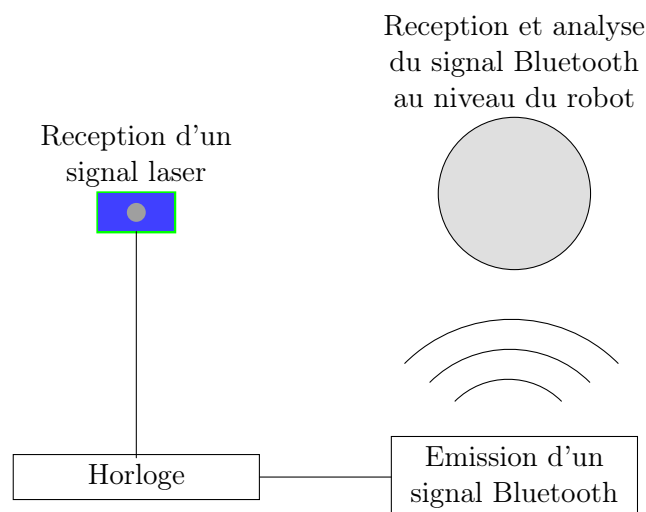


FIGURE 22 – Fonctionnement - système de localisation par laser



Cependant cette technique qui semble en théorie relativement simple est finalement très difficile à mettre en place. En effet, après quelque recherche sur le sujet, on s'aperçoit que cette solution technologique requiert une grande précision mécanique pour faire tourner le laser horizontalement. L'asservissement en vitesse et la connectique n'arrangent en rien le problème et la difficulté d'installation. De plus, afin de faire fonctionner les balises réceptrices, il aurait fallu les commander individuellement à l'aide d'une carte Arduino, installer un module Bluetooth et assurer la communication avec notre robot. C'est pour cela que nous avons préféré continuer nos recherches tout en mettant cette solution de côté.

4.4.4 Localisation par triangulation à l'aide d'une caméra

Le principe de cette méthode consiste à utiliser une Pixy cam, une caméra spécialement développée pour fonctionner avec Arduino. Nous avons choisi cette méthode car elle nous paraissait être la méthode la plus facile à mettre en place : on dispose d'une caméra qui peut effectuer des rotations autour de l'axe du robot.

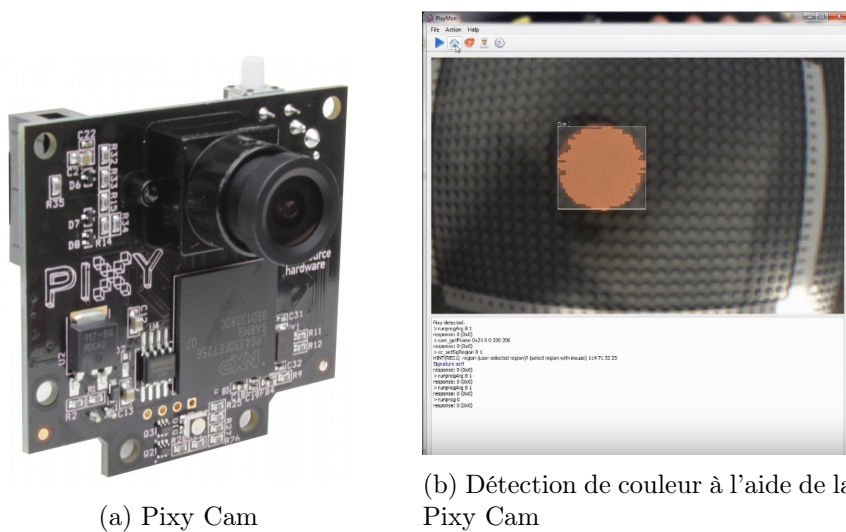


FIGURE 23 – Caméra

Cette dernière peut distinguer les couleurs. Ainsi nous souhaitons disposer de 3 balises de couleurs différentes sur notre aire de jeu avec des positions connues.

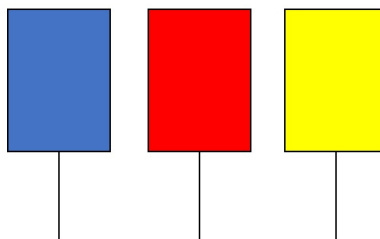


FIGURE 24 – Balises de différentes couleurs

La Pixy Cam peut repérer le centre de chaque balises et mesurer les angles entre les différentes balises grâce au système de rotation de la balise (qui tourne à l'aide d'un moteur pas à pas et qui est donc très précis)

N'ayant pas encore reçu la Pixy Cam, nous avons dans un premier temps réalisé la partie mathématique du problème (cf. Annexe D).

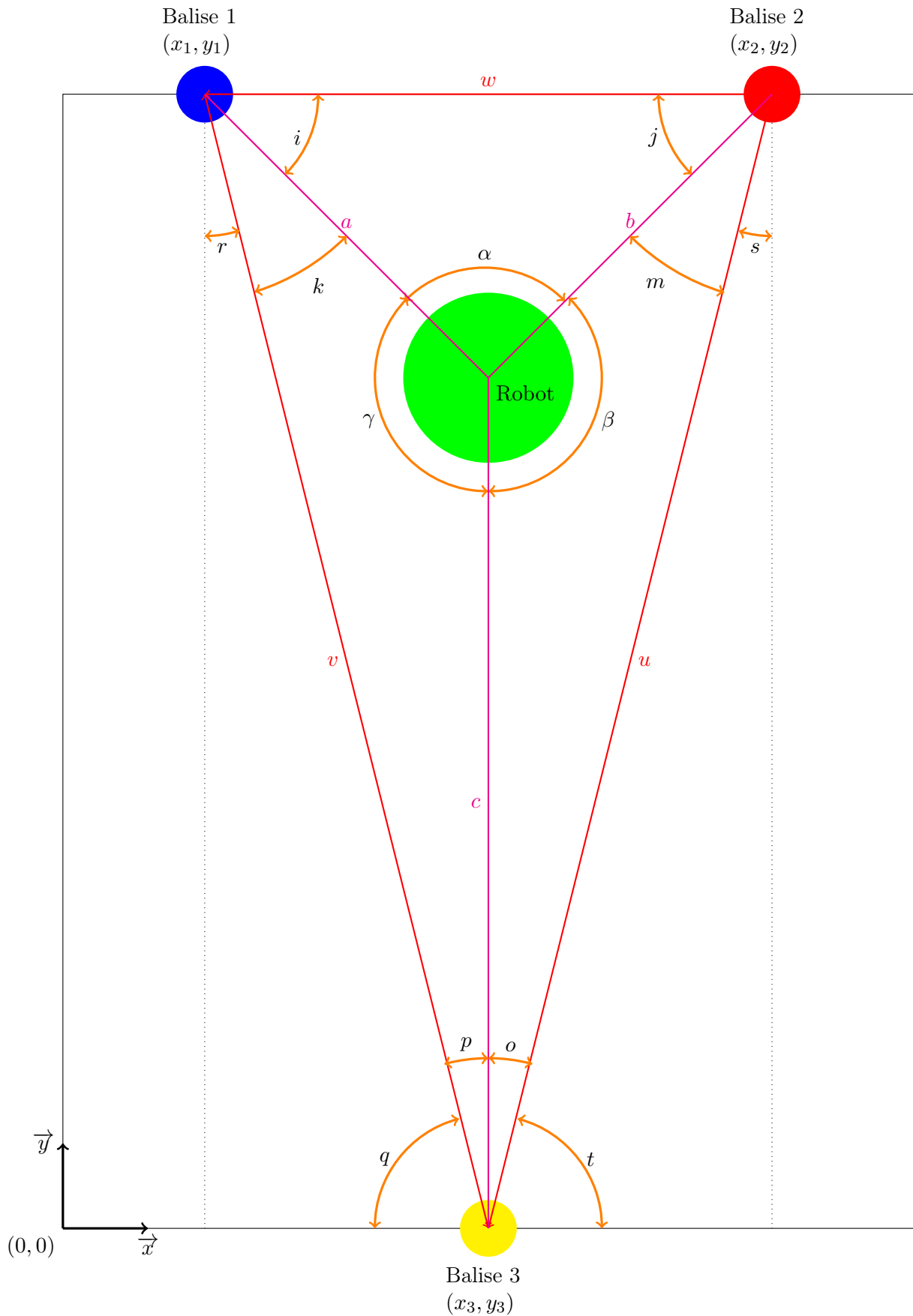


FIGURE 25 – Localisation par triangulation



5 Financement du projet

5.1 Budget

Le responsable du projet estime que ce projet pilote réclame un budget de fonctionnement de 8000 € tous moyens matériels confondus.

Les ressources mise à notre disposition (labo, atelier, matériel informatique, robot, ...) représentent la moitié de ce budget (soit 4000 €).

Le tableau en annexe F montre le budget nécessaire pour le bon déroulement du projet cette année. Les 3000 € restant serviront à l'achat de matériel plus performant et à couvrir les frais de déplacement pour les années suivantes.

5.2 Recherche de sponsors et mécénat

5.2.1 Financement

Nous sommes à la recherche d'entreprises ou organisations souhaitant soutenir le projet RobArts à hauteur de 4000 € pour que nous puissions atteindre le budget global de fonctionnement de ce projet de 8000 €. Nous contactons alors les entreprises dans une démarche de mécénat, c'est à dire qu'une entreprise choisissant de devenir mécène de notre projet pourra bénéficier d'une déduction fiscale de 60% du montant investi dans la limite de 0.5% de son chiffre d'affaire.

5.2.2 Nos prestations

Si les entreprises nous soutiennent lors du développement de notre projet, nous leur fournirons un espace dédié sur la coque de notre robot qu'elles pourront exploiter selon leur bon vouloir. Nous pouvons également réaliser une démonstration du robot sur demande. L'image de marque de l'entreprise sera ainsi mise en avant à chaque représentation et utilisation de notre robot, mettant en valeur l'entreprise et ses partenariats étudiants.

Finalement, soutenir les élèves des Arts et Métiers sera aussi l'occasion pour l'entreprise d'être mis en relation avec l'ensemble des acteurs de notre école afin d'amorcer ou d'entretenir une relation qui sera mutuellement bénéfique.

5.2.3 Support de communication

Nous avons également élaboré un soutien de communication. En effet, une plaquette explicative du projet a été faite par nos soins (cf. Annexe E). Elle nous servira lors du démarchage d'entreprises pour la recherche de financement.



6 Promotion du projet

6.1 Apérobot

Une soirée « Apérobot » est une réunion pour mettre en relation des passionnés ou curieux de la robotique. Nous avons décidé de nous y rendre afin de prendre contact avec les différents acteurs dans la robotique de la région ainsi que des particuliers intéressés à ce domaine.

Ainsi, nous avons pris contact avec la société Needabot qui organise l'évènement afin de présenter plus en détail notre projet et d'avoir plus de renseignements sur l'entreprise. Il nous a été également proposé de faire une visite guidée dans cette entreprise.

Peu d'acteurs étaient présents lors de cet Apérobot (seuls deux membres de Needabot étaient sur place). Cependant en discutant avec eux nous avons proposé de présenter notre projet lors du prochain Apérobot.

6.2 Visite d'entreprise

Nous envisageons de faire des visites d'entreprise dans les startups de l'agglomération de Châlons-en-Champagne afin de mettre en relation notre plateforme de robotique avec les différents acteurs de ce domaine à proximité.

La première visite organisée, le jeudi 8 février 2018 se fera chez Needabot à Châlons-en-Champagne. Cette entreprise est spécialisée dans le développement software de robots dans le domaine de l'aide à la personne. Après avoir pris contact avec Lucile Peuch, la directrice générale de Needabot, nous avons pu échanger avec elle et montrer notre intérêt pour son entreprise. Ainsi, la visite portera sur les sujets suivants :

- L'aspect technique, programmation, logiciels
- La gestion de projet
- Le lancement de la start up



FIGURE 26 – NeedaBot

Nous avons également pris contact avec l'entreprise Vitibot, spécialisée dans la fabrication de robots viticoles automatisés pour organiser une visite. Cependant, nous n'avons pas eu de réponse de leur part.



FIGURE 27 – Vitibot



7 Travail à réaliser au second semestre

Nous avons listé nos objectifs à atteindre lors du deuxième semestre :

7.1 Exploitation du robot

- Implémentation de capteurs tels que des capteurs de température et humidité.
- Gestion de l'envoi de données en direct par Bluetooth et réception par le robot.
- Enregistrement des données sur une carte SD sur une durée maximale de 24 heures.
- Programmation du robot de façon à atteindre une position donnée.

7.2 Recherche de financement pour le projet

Démarchage d'entreprises pour lever des fonds (financiers ou aide en nature).

7.3 Promotion du projet

- Création d'un site web pour le projet et CMS (Content Management System), une famille de logiciels destinés à la conception et la mise à jour d'un site web.
- Apérobot pour présenter le projet devant des experts ainsi qu'un public amateur.
- Présentation du robot au centre Leclerc Fagnières devant du public général.



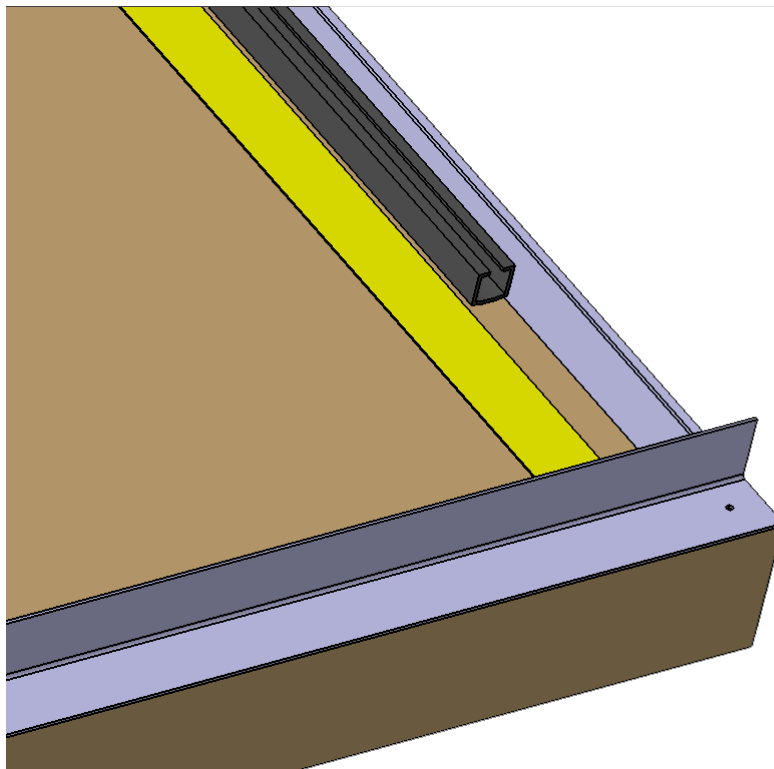
8 Annexes

A Cahier des charges

Fonctions	Critère	Niveau	Solution technologique proposée
S'adapter à l'aire de jeu	Défaut de planéité du sol	3 mm	Assemblage de 2 portes planes
	Adhérence	Ne pas glisser	Bon état de surface de l'aire de jeu
Résister à l'aire de jeu	Choc avec les barrières	Respecter l'aire de jeu	Capteur à ultrasons
Respecter l'aire de jeu	Ne doit pas endommager les barrières, capteurs, balises		Capteur à ultrasons
Resister à l'air	Température	0-40°C	
	Humidité	0-87%	
	Vent	Sans objet	
	Pression	Atmosphérique	
	Poussière	Sans objet	
Respecter l'air	Ne pas polluer	Tout électrique	Moteurs électriques
S'adapter au téléphone	Compatibilité	iOS/ Android	Développement d'application sur AppInventor
Communiquer avec le téléphone	Recevoir des informations	BLE	Adafruit Bluefruit LE Shield
	Envoyer des informations	BLE	Adafruit Bluefruit LE Shield
S'adapter à l'utilisateur	Toucher	Ergonomique	Coque réalisée en impression 3D
	Visuel	Communiquer	Ecran LCD
	Ouïe	Emettre un son	Haut-parleur
	Être utilisable par tous	Facile d'utilisation	Arduino
Respecter l'utilisateur et le public	Toucher	Ergonomique	Coque réalisée en impression 3D
	Visuel	Ne pas éblouir	LED/Capteurs
	Ouïe	0-70 dB	Haut-parleur
Resister à l'utilisateur	Toucher	Ne pas s'abimer lorsque pris en main	
Resister au public	Interférences	Utiliser des technologies peu perturbées par l'extérieur	
S'adapter aux balises	Repérer les balises	Repérer une gamme de 8 couleurs de balises	Utilisation d'une Pixy Cam
Utiliser les balises	Se repérer par triangulation grâce aux balises	1cm d'erreur	Utilisation d'une Pixy Cam
S'adapter à l'ordinateur	Compatibilité	Windows/iOS	
Communiquer avec l'ordinateur	Compiler et télécharger un programme, acquisition de données	cable usb type A/B	Arduino IDE, Sublime text
Utiliser l'énergie du réseau EDF	Tension	9V	Adaptateur chargeur
	Puissance Maxi	9W	
Avoir de l'autonomie	Batteries	4h	Piles rechargeables



B Maquette numérique





C Codes

C.1 Protocole expérimental - essai moteur

Le but de cette expérience est de vérifier que la vitesse du moteur est bien proportionnelle à la commande. Pour cela, on a placé une ligne de départ sur l'aire de jeu. On place ensuite le robot derrière la ligne de départ de telle sorte qu'il se déplace en ligne droite parallèlement à l'un des côtés de l'aire de jeu. Ainsi, grâce à la mire on peut mesurer la distance parcourue par le robot pendant 2 secondes. On en déduit la vitesse du robot :

$$v = \frac{d}{t}$$

Code utilisé pour les essais :

```

1 #include <ArduinoRobot.h>
2 #include <Wire.h>
3 void setup()
4 { Robot.begin(); // Intitialisation du robot
5   Robot.waitContinue(); // Le programme demarre quand on appuie sur le bouton centrale }
6 void loop()
7 { Robot.motorsWrite(150,150); // Le moteur avance en ligne droite avec la commande
   souhaitee
8   delay(2000);
9   Robot.motorsStop(); // Le robot s'arrete au bout de 2 sec
10  Robot.waitContinue(); // On recommence ce cycle apres appuie sur le bouton centrale}

```

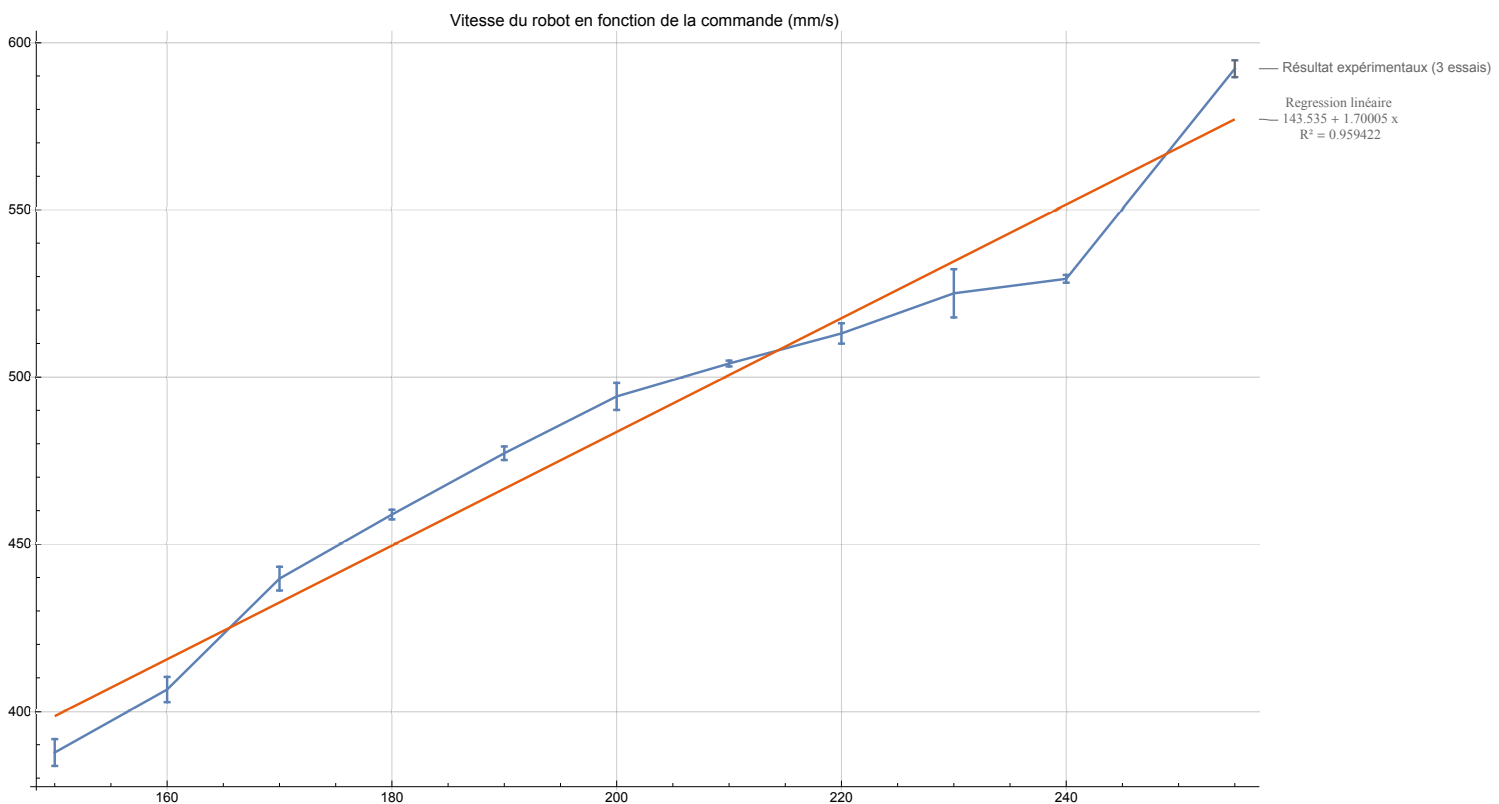


FIGURE 28 – Résultats expérimentaux



C.2 Boussole

```

1 int speedLeft;
2 int speedRight;
3 int compassValue;
4 int direc = 90; //Direction demandee (90 ==> vers l'est)
5 #include <ArduinoRobot.h>
6 #include <Wire.h>
7
8 void setup() {
9     // initialisation des modules
10    Robot.begin();
11    Robot.beginTFT();
12    Robot.beginSD();
13    Robot.displayLogos();
14 }
15
16 void loop() {
17     compassValue = Robot.compassRead(); \\ Lecture de la valeur de la boussole
18     int diff = compassValue - direc; \\ Difference entre la valeur reelle et la
19     valeur demandee
20
21     // on ramenne la difference entre -180 et 180
22     if (diff > 180) {
23         diff = -360 + diff;
24     } else if (diff < -180) {
25         diff = 360 + diff;
26     }
27
28     // Passage de l'intervalle [-180;180] (degres) ==> l'intervalle [-255;255] (
29     commande moteur)
30     diff = map(diff, -180, 180, -255, 255);
31
32     if (diff > 0) {
33         \\ difference > 0 ==> robot trop a gauche ==> on fait tourner le robot a
34         droite
35         speedLeft = 150 - diff;
36         speedRight = 150;
37     } else {
38         // situation inverse ...
39         speedLeft = 150;
40         speedRight = 150 + diff;
41     }
42     Robot.motorsWrite(speedLeft, speedRight); // On effectue le changement de
43     commande moteur
44
45     Robot.drawCompass(compassValue); \\ On montre la boussole sur l'ecran
46 }

```



C.3 Line Follow

```

1 #include <ArduinoRobot.h>
2 #include <Wire.h>
3 void setup()
4 {
5     //Initialisation des modules
6     Robot.begin();
7     Robot.beginTFT();
8     Robot.beginSD();
9     Robot.beginSpeaker();
10
11     Robot.waitContinue(); // Le programme démarre quand on appuie sur le bouton
        centrale
12
13 }
14 //Partie du code modifiée
15 void loop(){
16     Robot.updateIR(); //On met à jour les valeurs des capteurs infrarouge
17     int b = Robot.IRarray[1]; //On lit la valeur du capteur 2
18     int c = Robot.IRarray[2]; // On lit la valeur du capteur 3
19     int d = Robot.IRarray[3]; // On lit la valeur du capteur 4
20     // Le capteur infrarouge renvoie une valeur proche de 900 lorsqu'il est sur
        du blanc et de proche de 100 sur du noir
21
22     if(b>600){ //capteur 2 sur du blanc ==> robot trop à droite
23         Robot.motorsStop(); //On arrête le robot
24         Robot.motorsWrite(-100,100); // Le robot tourne sur sa gauche ...
25         delay(50); // ... Pendant 5 centièmes de seconde
26         Robot.motorsStop(); //On arrête le moteur
27     }
28     if(d>600){ //cas inverse
29         Robot.motorsStop();
30         Robot.motorsWrite(100,-100);
31         delay(50);
32         Robot.motorsStop();
33     }
34     if(b<600 && d<600){ //Si ni le capteur 2 ni le capteur 4 ne sont sur du blanc
        ==> le robot est bien sur la ligne ==> le robot avance tout droit pendant
        un dixième de seconde
35         Robot.motorsWrite(100,100);
36         delay(100);
37     }
38     if( b>600 && c>600 && d>600 ){ //si les 3 capteurs sont sur du blanc ==> on
        est arrivé à la fin de la ligne noire
39         Robot.motorsStop(); // Arrêt du robot
40         Robot.waitContinue(); // Tant que l'on appuie pas sur le bouton central
41     }
42 }
    
```



D Feuille de calcul - triangulation

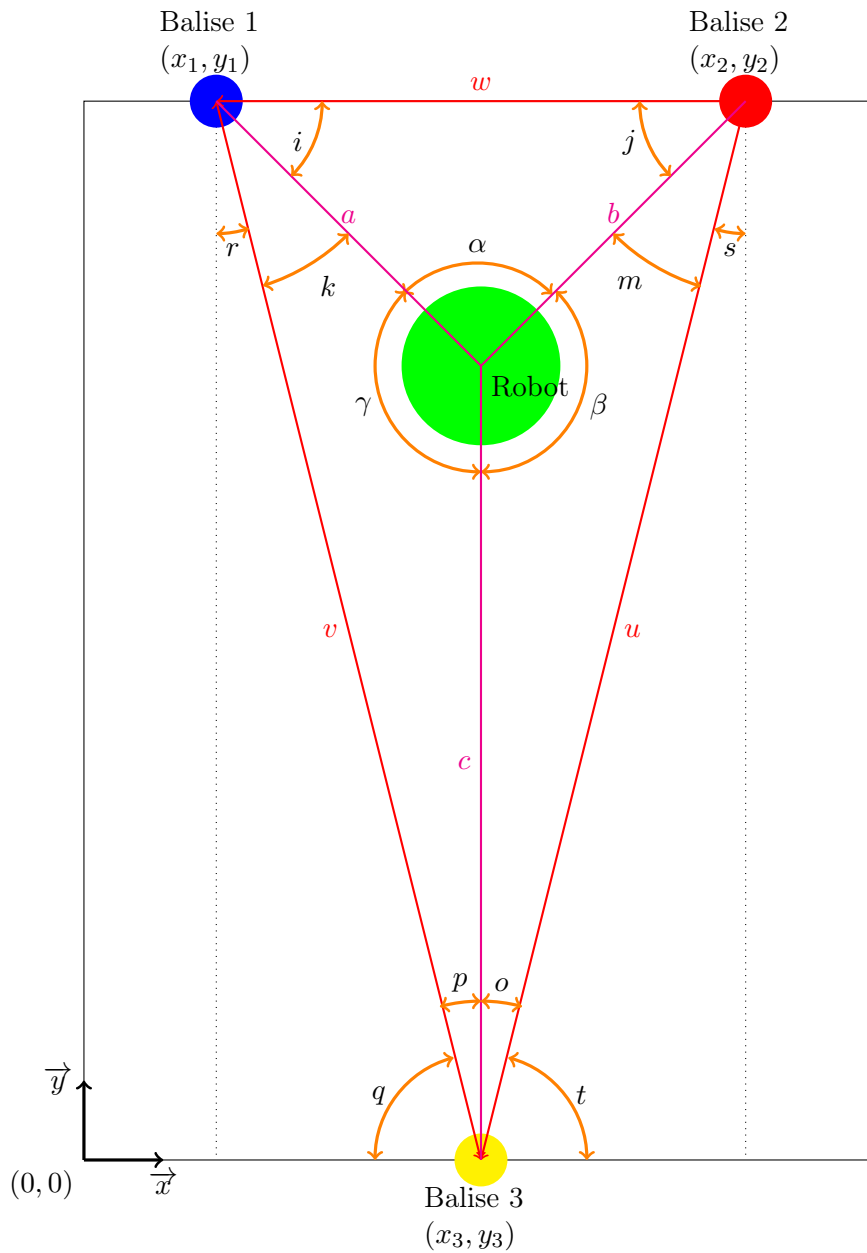


FIGURE 29 – Paramétrage

Formule du sinus :

$$\begin{cases} \frac{v}{\sin \gamma} = \frac{a}{\sin p} = \frac{c}{\sin k} \\ \frac{w}{\sin \alpha} = \frac{a}{\sin j} = \frac{b}{\sin i} \\ \frac{u}{\sin \beta} = \frac{b}{\sin o} = \frac{c}{\sin m} \end{cases}$$

On en déduit a , b et c en fonction des autres paramètres :

$$\begin{cases} a = \frac{v}{\sin \gamma} \cdot \sin p = \frac{w}{\sin \alpha} \cdot \sin j & (1) \\ b = \frac{w}{\sin \alpha} \cdot \sin i = \frac{u}{\sin \beta} \cdot \sin o & (2) \\ c = \frac{u}{\sin \beta} \cdot \sin m = \frac{v}{\sin \gamma} \cdot \sin k & (3) \end{cases}$$



Par définition :

$$\begin{cases} j + m = 90 - s \\ m = 180 - o - \beta \end{cases}$$

On en déduit :

$$o = j + s - \beta + 90$$

Et :

$$i = 180 - j - \alpha$$

On injecte dans (2) :

$$\frac{w}{\sin \alpha} \cdot \sin(180 - j - \alpha) = \frac{u}{\sin \beta} \cdot \sin(j + s - \beta + 90) \quad (4)$$

On pose :

$$\begin{cases} A = \frac{u}{\sin \beta} \\ B = \frac{w}{\sin \alpha} \\ \Phi_1 = s - \beta + 90 \\ \Phi_2 = \alpha - 180 \end{cases}$$

(4) s'écrit alors :

$$A \cdot \sin(j + \Phi_1) = -B \cdot \sin(j + \Phi_2)$$

En utilisant la formule de développement, on a :

$$U \cdot \sin j = -V \cdot \cos j$$

Avec :

$$\begin{cases} U = A \cdot \cos \Phi_1 + B \cdot \cos \Phi_2 \\ V = A \cdot \sin \Phi_1 + B \cdot \sin \Phi_2 \end{cases}$$

Or $\sin x = \sqrt{1 - \cos^2 x}$, d'où :

$$U \cdot \sqrt{1 - \cos^2 j} = -V \cos j$$

On en déduit :

$$\cos j = \sqrt{\frac{U^2}{U^2 + V^2}}$$

Comme $-1 \leq \sqrt{\frac{U^2}{U^2 + V^2}} \leq 1$:

$$j = \arccos\left(\sqrt{\frac{U^2}{U^2 + V^2}}\right)$$

Avec les paramètres du problème :

$$j = \arccos\left(\sqrt{\frac{[A \cdot \cos(s - \beta + 90) + B \cdot \cos(\alpha - 180)]^2}{[A \cdot \cos(s - \beta + 90) + B \cdot \cos(\alpha - 180)]^2 + [A \cdot \sin(s - \beta + 90) + B \cdot \sin(\alpha - 180)]^2}}\right)$$

On rappelle que :

$$\begin{cases} i = 180 - \alpha - j \\ a = \frac{w}{\sin \alpha} \cdot \sin j \\ b = \frac{w}{\sin \alpha} \cdot \sin i \end{cases}$$

On a donc les coordonnées du robot telle que :

$$\begin{cases} x = x_1 + \cos i \cdot a \\ y = y_1 + \sin i \cdot a \end{cases}$$



Programmation de la fonction de triangulation sous Arduino :

```

1 #include <stdio.h>
2 #include "math.h"
3
4 float x,y;
5
6
7 void position(float alpha,float beta,float gama) //Fonction de calcul de la position
  en fonction des angles inter balise
8 {
9     //Dimenssions et angles connues
10    float L = 2040; // longueur du terrain
11    float w = 1660; // longueur entre B1 et B2 = 2 x 830 mm
12    float v = sqrt(pow(830,2)+ pow(2040,2)); // longueur entre B1 et B3
13    float u = sqrt(pow(830,2)+ pow(2040,2)); // logueur entre B2 et B3
14    float r = atan(830/2040); // angle(0,B1,B3)
15    float q = (M_PI/2)-r; // angle(B1,B3,0)
16    float s = atan(830/2040); //angle
17    float t = (M_PI/2)-s; //angle
18
19    // Calcul des angles inconnus
20    float j = acos(sqrt(pow(((u/sin(beta))*cos(s-beta+M_PI/2)+(w/sin(alpha))*cos(alpha-
      M_PI)),2)/(pow(((u/sin(beta))*cos(s-beta+M_PI/2)+(w/sin(alpha))*cos(alpha-M_PI))
      ,2)+pow(((u/sin(beta))*sin(s-beta+M_PI/2)+(w/sin(alpha))*sin(alpha-M_PI)),2)))));
21    float i = M_PI-alpha-j;
22    printf("%.6f", j);
23    printf("\n");
24    printf("%.6f", i);
25    printf("\n");
26
27    //Calcul des longueur inconnus
28    float a = w*sin(j)/sin(alpha);
29    float b = w*sin(i)/sin(alpha);
30    printf("%.6f", a);
31    printf("\n");
32    printf("%.6f", b);
33    printf("\n");
34    //Calcul des coordonees
35    x = cos(i)*a;
36    y = L - sin(i)*a;
37 }
38
39 int main(int argc, const char * argv[]) {
40
41    position(M_PI/2,3*M_PI/4,3*M_PI/4); //Fonction de calcul de la position
42    printf("%.6f", x);
43    printf("\n");
44    printf("%.6f", y);
45    printf("\n");
46
47    return 0;
48 }
    
```



E Plaquette de communication (en cours d'élaboration)

À propos de nous

Notre équipe est composée de 6 étudiants ingénieurs d'Arts et Métiers ParisTech. Tous fascinés par la robotique, nous nous sommes lancés dans la création d'un projet avec un objectif commun, vivre notre passion.

TEAM WORK

Chacun apporte au projet ses compétences dans divers domaines, depuis la programmation jusqu'à la gestion de projet en passant par la communication et l'innovation.

Notre objectif est de fonder au sein de l'école un 'club robotique' qui soit une base solide sur laquelle les élèves pourront s'appuyer pour s'épanouir dans leurs projets pédagogiques.

Nous suivre

Retrouvez nous sur youtube nous postons régulièrement des vidéos sur avancées de notre projet.

L'équipe :

- Luc Beltran, chef de projet et responsable software ;
- Mathieu Bouvet, trésorier et responsable logistique ;
- Benoît Boyeau, responsable communication et fabrication ;
- Thomas Merlet, assistant chef de projet et responsable électronique ;
- Diana Jiménez Vega, vice-trésorière et responsable des relations extérieures ;
- Joséphine Kharitonoff, designer, responsable maintenance et gestion de l'environnement du robot.

Professeurs encadrant le projet :

- Alain Prévot, professeur de Génie Mathématique ;
- Jean-Marc Périnet, professeur de Génie Mécanique.

Retrouvez nous sur le campus Arts et Métiers de Châlons-en-Champagne, Rue Saint Dominique, 51000.

Pour plus d'informations, n'hésitez pas à nous contacter :

RobArts@gmail.com

06 70 26 27 92

**ARTS
ET MÉTIERS**
ParisTech

Rob Arts
La science pour se mouvoir

Projet étudiant

...

Recherche, innovation et développement technologique dans le domaine de la **robotique**.

...

Notre projet

Fabriquer notre robot ? Organiser notre projet ? Gérer la communication pour un travail de groupe ludique et efficace ? Notre projet répond à toutes ces questions et nous relevons le défi d'y répondre avec notre équipe !

Soutenir notre projet

Pourquoi devenir partenaire ?

- Soutenir un projet étudiant à fort potentiel ;
- Mettre en avant votre implication dans un projet innovant ;
- Investir sur les talents de demain.

Comment ?

- Via mécénat :

Bénéficiez d'une déduction fiscale de 60% du montant investi (dans la limite de 0,5% de votre chiffre d'affaires).

- Via sponsoring :

Votre logo sera présent sur nos supports de communication ainsi que sur le robot lors des présentations publiques telles que la coupe de France ou la coupe d'Europe.

Nos objectifs

- **Court terme**
Créer un robot capable de se repérer dans son environnement et de prendre des décisions directionnelles sur son itinéraire.
- **Long terme**
Participer à la coupe de France et à la coupe d'Europe de robotique.

Nos outils

Nous avons choisi comme support la technologie Arduino combinée aux capteurs Adafruit.

Concernant la programmation d'applications sous Android, nous exploitons l'interface App Inventor.

Pour la gestion de projet nous utilisons le logiciel Trello.

Notre budget

Au total notre budget s'élève à 2 000€.

Il comprend l'achat du matériel pour fabriquer le robot (1500€) ainsi que les frais annexes afin de participer au concours robotique (500€).

FIGURE 30 – Plaquette de communication



F Budget 2017/2018

Matériel	Quantité	Prix unitaire	Prix total
Aire de jeu			
Plaques bois 1250*2500*10 mm	2	60,00 €	120,00 €
Rails alu	4	13,90 €	55,60 €
Mètres	4	1,57 €	6,28 €
Fixation Grenouillère	1	3,45€	3,45 €
Robot			
Adafruit BLE	1	25,19 €	25,19 €
Moteur pas à pas	2	12,99 €	25,98 €
Batterie (4 ACCUS HR6 2500)	1	15,55€	15,55 €
Pixy (Cmucam5) Smart Vision Sensor	3	101,09 €	303,27 €
Support 2 axes avec servos pour Pixy Cam	3	59,99 €	179,97 €
Carte Arduino Nano x3	2	10,99 €	21,98 €
Carte Arduino Mega	2	34,99 €	69,98 €
Neuftech 3x Module à ultrasons	1	8,99 €	8,99 €
Capteur de température NTC DHT	1	4,30 €	4,30 €
Centrale inertielle MPU6050	1	2,22 €	2,22 €
Breadboard	1	7,59 €	7,59 €
Photodiode x3	1	5,79 €	5,79 €
Lecteur Carte SD	1	2,19 €	2,19 €
Arduino Module écran LCD couleur	1	6,30 €	6,30€
Capteur infrarouge	1	12,90 €	12,90 €
Robot Arduino	1	150,00 €	150,00 €
TOTAL :			1 027,53 €

Tableau 5 – Budget 2017/2018