
RAPPORT DE PROJET

Projet d'électronique n°4 : DRAWBOT

Auteurs :

Erwan	PLANET
Thibaud	DANSETTE
Hugo	FAIVRE
Léo-Pol	BERNARD CHAUSSEE

Enseignant :

Mme NEDRA

Ce projet a été conçu dans le contexte du cours sur les systèmes bouclés à l'ECE. L'objectif est de créer un robot autonome capable de tracer des formes au sol, baptisé le Drawbot. Pour cela, nous utilisons une carte NodeMCU ESP32 installée sur la plateforme Gyrobot, qui est équipée de moteurs avec encodeurs, d'une centrale inertielle (IMU) et d'un magnétomètre.

L'objectif est d'implémenter des boucles de rétroaction afin de maîtriser avec précision les mouvements du robot. Trois tracés sont à réaliser : un escalier, concevoir un cercle avec un rayon ajustable, et réaliser une flèche ou une rose des vents pointant vers le Nord. Chaque tracé requiert une interaction sans fil avec l'ordinateur, un traitement des données provenant des capteurs et un contrôle des moteurs à l'aide d'un régulateur PID.

Ce projet nous offre l'opportunité d'appliquer nos compétences en automatique, électronique et programmation sur microcontrôleur. Cela nous incite également à collaborer en groupe, à gérer notre emploi du temps efficacement et à consigner de manière appropriée chaque phase du développement. Un des défis majeurs est de concevoir un robot précis et stable malgré les contraintes matérielles, tout en produisant un résultat soigné et reproductible.

Nous attestons que ce travail est original, qu'il est le fruit d'un travail commun au groupe et qu'il a été rédigé de manière autonome.

Paris, le 15/05/2025

Table des matières

I. Objectif.....	3
II. Glossaire.....	3
A. Termes.....	3
B. Acronymes.....	4
III. L'équipe	5
A. Présentation de l'équipe	5
B. Organisation de l'équipe.....	5
C. Diagramme de Gantt	6
IV. Contexte et problématique	6
A. Contexte	6
B. Problématique.....	7
C. Spécifications techniques	7
V. Conception	8
A. Architecture fonctionnelle	8
B. Architecture matérielle.....	9
C. Architecture logicielle	11
VI. Développement	13
A. Module 1- Interface Wifi Bluetooth.....	13
B. Module 2- Le déplacement du robot	13
C. Module 3– Séquence 1 : L'escalier.....	15
D. Module 4 – Séquence 2 : Le cercle	15
E. Module 5 – Le magnétomètre	16
F. Module 6 – Séquence 3 : La rose des vents.....	16
G. Module 7 – Conception de la coque	17
VII. Tests et validation.....	18
A. Module 1- Interface Wifi Bluetooth.....	18
B. Module 2- Le déplacement du robot	19
C. Module 3– Séquence 1 : L'escalier.....	20
D. Module 4 – Séquence 2 : Le cercle	24
E. Module 5 – Le magnétomètre	26
F. Module 6 – Séquence 3 : La rose des vents.....	27
G. Module 7 – Conception de la coque.....	30
VIII. Bilan	31
A. État d'avancement.....	31
B. Pertinence de la solution technique	31
C. Bilan sur le travail d'équipe	31
IX. Bibliographie	33
X. Annexes	34

I. Objectif

Dans ce document, nous allons vous exposer l'ensemble de nos recherches menées pour développer au mieux notre projet.

Ce projet s'intitule Drawbot : il s'agit d'un robot que nous devons programmer afin de lui faire dessiner des motifs.

Vous trouverez dans ce document deux parties principales intitulées « Développement » et « Tests et validations ». Celles-ci vous présenteront en détail le déroulement de nos recherches pour concevoir ce robot dans les meilleures conditions.

II. Glossaire

A. Termes

Terme	Définition
DRAWBOT	Nom du projet qui consiste à concevoir un robot capable de dessiner des formes sur une surface à l'aide d'un feutre fixé sur sa structure commandé via wifi.
NodeMCU ESP32	Microcontrôleur avec Wi-Fi et Bluetooth intégré, utilisé pour piloter le robot et assurer la communication avec l'interface web.
Capteurs	Dispositifs permettant de mesurer des grandeurs physiques (vitesse, orientation, distance) et de les transmettre au microcontrôleur.
KiCad	Logiciel de conception de schémas électroniques et de circuits imprimés utilisé pour dessiner la schématique du Gyrobot.
Drivers	Circuits intégrés permettant de piloter les moteurs en amplifiant les signaux envoyés par le microcontrôleur.
Magnétomètre (MAG)	Capteur mesurant l'orientation du robot par rapport au champ magnétique terrestre, utilisé comme boussole dans notre projet.
Centrale inertielle (IMU)	Capteur combinant accéléromètre et gyroscope pour mesurer les accélérations et vitesses angulaires du robot.
Encodeur	Capteur placé sur l'axe du moteur pour mesurer les rotations de roue, permettant de calculer la distance parcourue.
LED	Diode électroluminescente.

Motoréducteur	Moteur électrique associé à un réducteur de vitesse pour obtenir un couple plus élevé et un contrôle précis du mouvement.
Switch	Interrupteur permettant d'allumer ou d'éteindre le robot avec les piles.
Condensateur	Composant électronique stockant de l'énergie sous forme électrique. Il est utilisé pour stabiliser la tension et filtrer les signaux.
Headers	Connecteurs femelles soudés sur la carte permettant d'insérer des broches du microcontrôleur (ESP32).
Asservissement	Méthode de contrôle permettant d'ajuster automatiquement une variable pour suivre une demande grâce à une boucle de rétroaction.
Boucle de rétroaction	Système de contrôle où la sortie est mesurée et réutilisée pour ajuster l'entrée.

B. Acronymes

Acronyme	Signification	Explication
I2C	Inter-Integrated Circuit	Protocole de communication série utilisé entre le microcontrôleur et les capteurs : IMU, magnétomètre dans notre projet.
GPIO	General Purpose Input, Output	Broche numérique du microcontrôleur pouvant être utilisée en entrée ou sortie.
Correction PID	Proportionnel, Intégral, Dérivatif	C'est un correcteur utilisé pour stabiliser le comportement du robot en ajustant la commande motrice en fonction de l'erreur mesurée.
PWM	Pulse Width Modulation	Technique de modulation utilisée pour contrôler la vitesse des moteurs en variant la largeur des impulsions envoyées.

III. L'équipe

A. Présentation de l'équipe

Notre équipe est composé de 4 étudiants, Erwan PLANET, Hugo FAIVRE, Thibaud DANSETTE ainsi que Léo-Pol BERNARD-CHAUSSEE.

Nous avons fait le choix de nous mettre dans le même groupe de projet car nous nous entendions bien, avec un bon esprit de travail.

Nom, Prénom	Compétences	Qualités
Erwan PLANET	Logique pour le code, soudure, recherches	Rigoureux, ambitieux
Hugo FAIVRE	KiCad, schématiques	Méticuleux, créatif
Thibaud DANSETTE	KiCad, code	Esprit d'analyse, persévérant
Léo-Pol BERNARD-CHAUSSEE	Recherches, Scilab	Communication, attentif

B. Organisation de l'équipe

Avant de commencer notre projet nous avons fait un brainstorming pour mettre en place une liste de toutes les tâches à réaliser, puis avons réparti le travail équitablement en fonction des compétences de chaque membre. Nous avons aussi fait en sorte que même si un membre ne fais pas une certaine tâche, la personne qui la réalise prenne le temps d'expliquer comment il a fait pour y parvenir et son fonctionnement. Cela permet à toute l'équipe de comprendre le travail réalisé par tous.

Voici comment nous nous sommes reparti les tâches :

Erwan s'est occupé de la séquence 1, les escaliers.

Hugo a pris en charge la deuxième séquence, les cercles.

Thibaud a travaillé sur la séquence 3 avec la rose des vents

Léo-Pol c'est occupé de la schématique sur KiCad, ainsi que du design du site.

Nous avons essayé de nous répartir équitablement les tâches.

C. Diagramme de Gantt

Pour organiser au mieux le temps consacré à notre projet, nous avons mis en place un diagramme de Gantt nous permettant de visualiser facilement les tâches principales à réaliser ainsi que la durée maximale et la date limite pour les finir.

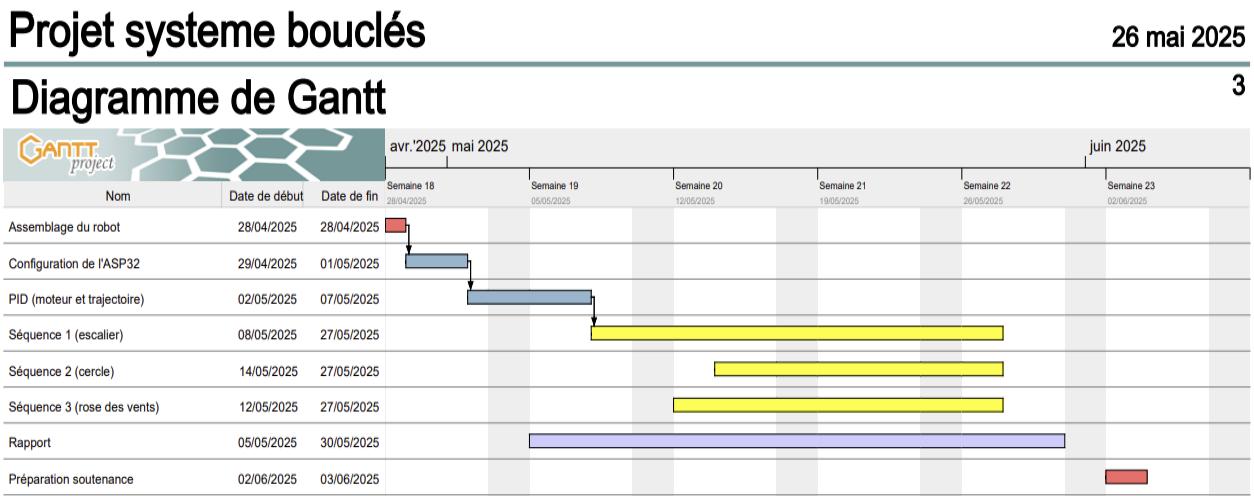


Figure 1 : Diagramme de Gantt

IV. Contexte et problématique

A. Contexte

Le projet s'inscrit dans un contexte éducatif et technologique. Il vise à nous former aux systèmes embarqués et à la commande en boucle fermée, compétences très demandées dans les secteurs de la robotique, de l'automatisation et de l'électronique.

Gyrobot a été développée par l'ECE pour le projet de système bouclée. Elle évolue chaque année avec de nouveaux cas d'usage. Le Drawbot s'appuie sur cette base, enrichie par l'ajout de capteurs dont le magnétomètre et la centrale inertielles et d'un contrôle PID pour améliorer la précision du mouvement.

B. Problématique

Comment concevoir un robot mobile capable d'exécuter des trajectoires de dessin précises, en interprétant des commandes à distance et en s'auto-corrigent à l'aide de capteurs embarqués, tout en respectant des contraintes de précision sur les distances, les angles et l'orientation ?

C. Spécifications techniques

Notre robot Drawbot a pour objectif principal de réaliser des dessins de manière autonome à partir de commandes envoyées depuis un ordinateur via une connexion sans fil. Il doit être capable d'exécuter différentes figures géométriques comme une séquence de lignes droites (forme d'escalier), un cercle de rayon variable et une flèche orientée vers le pôle Nord terrestre et en complément de cette flèche, en fonctionnalité avancée, une rose des vents. Chaque trajectoire doit être suivie précisément, en tenant compte des ordres reçus, tout en corrigeant les écarts de mouvement à l'aide de capteurs, d'un système de rétroaction (boucle fermée) et d'un correcteur PID.

Les composants utilisés pour réaliser ce projet sont les suivants :

- Un microcontrôleur, la carte nodeMCU ESP32
- Deux motoréducteurs N20 à 100 RPM avec encodeurs
- Drivers de moteurs
- Une centrale inertielle (LSM6DS3)
- Un magnétomètre (LIS3MDL)
- Une alimentation sur piles AA, un total de 6volts
- Un châssis servant de PCB
- 2 roues de 4.5cm de rayon
- Un feutre pour tableau blanc

Le robot doit respecter plusieurs critères de précision afin de valider son bon fonctionnement. Lors du dessin de trajectoires droites, une tolérance maximale de 1 cm est autorisée sur les distances parcourues. Pour les rotations, l'erreur angulaire ne doit pas dépasser 5°. Le cercle tracé doit être fermé avec une erreur de fermeture inférieure à 5 mm, et son rayon doit pouvoir être défini entre 2 et 20 cm, avec une tolérance de ± 5 mm. Enfin, pour la séquence orientée, la flèche dessinée doit pointer vers le Nord magnétique avec une précision de ± 10 °.

V. Conception

A. Architecture fonctionnelle

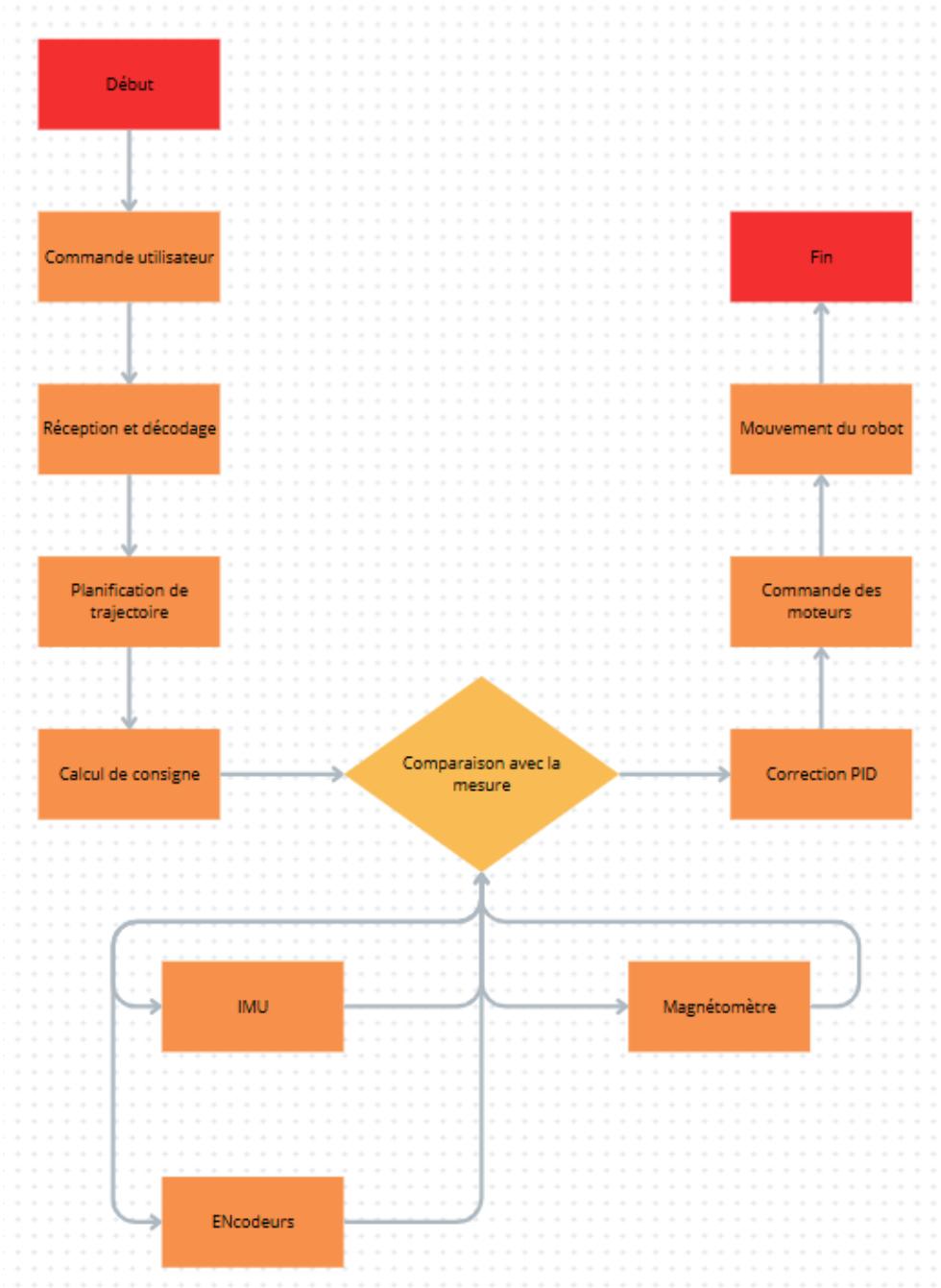


Figure 2 : Architecture fonctionnelle

B. Architecture matérielle

Pour le déroulement de notre projet, nous allons utiliser les composants ci-dessous afin de réaliser notre projet au mieux.

- Une carte de programmation ESP32 : elle est pratique pour sa vitesse, sa connectivité Wi-Fi ou Bluetooth et sa compatibilité avec les capteurs.
- Des moteurs avec réducteur et encodeur numérique : pour un mouvement précis et mesurable.
- Des capteurs LSM6DS3 et LIS3MDL : afin de connaître l'orientation et tracer en direction du Nord.
- Deux drivers DRV8837DSGR : pour gérer les moteurs avec précision.
- Un circuit imprimé (PCB) : permettant de relier tous les composants entre eux
- Et pour finir 4 piles de 1.5v pour alimenter le circuit.

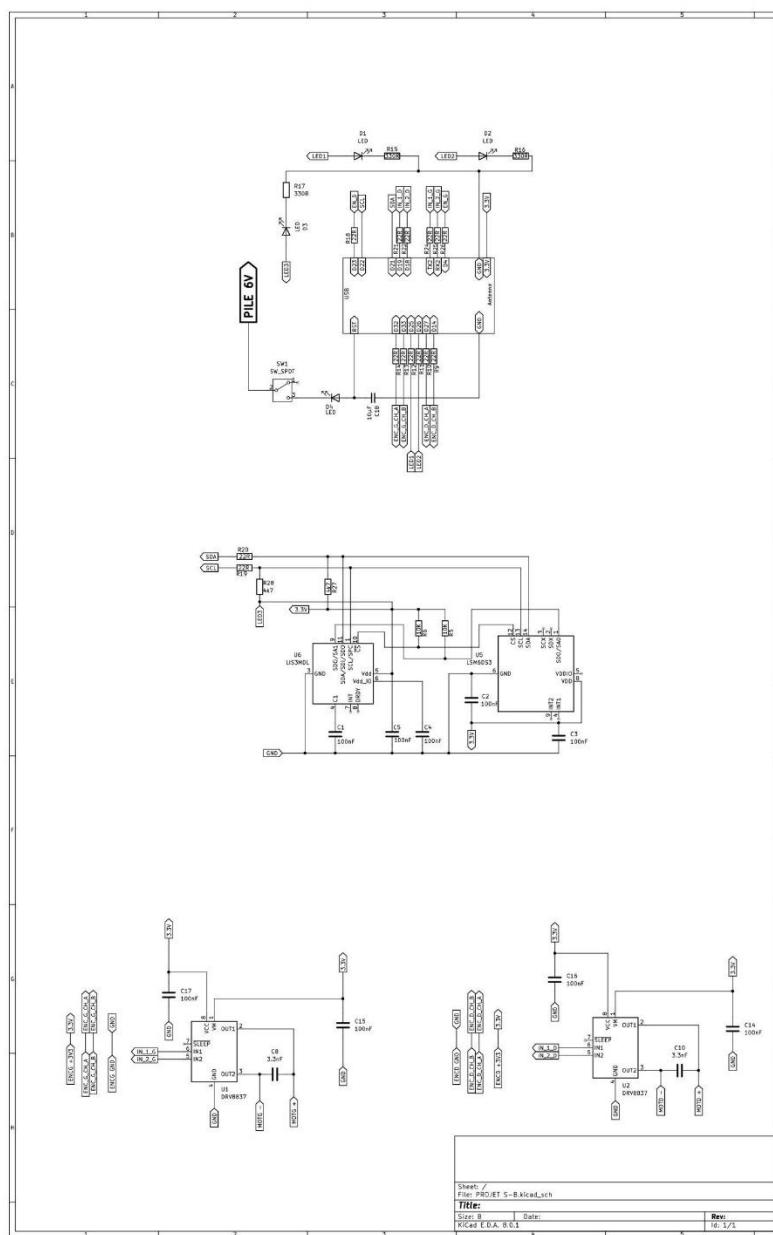


Figure 3 : Schématique du PCB de notre robot fait sur KiCad

Sur la figure 3, nous pouvons observer la schématique du robot. En bas, nous avons les deux drivers qui permettent de commander les moteurs. Au milieu de la figure, nous avons disposé deux capteurs (LIS3MDL, le magnétomètre ainsi que LSM6DS3, la centrale inertuelle), et en haut, nous avons schématisé la carte ESP32 avec les pins.

Comment les différentes briques techniques sont connectées entre elles ?

L'ordinateur communique par Wifi avec l'ESP32 qui centralise le système : il commande les moteurs via les drivers DRV8837, lit les encodeurs en direct, et récupère les données de la centrale inertuelle ainsi que du magnétomètre via le bus I2C (SDA/SCL). L'alimentation est assurée par des piles connectées au circuit imprimé.

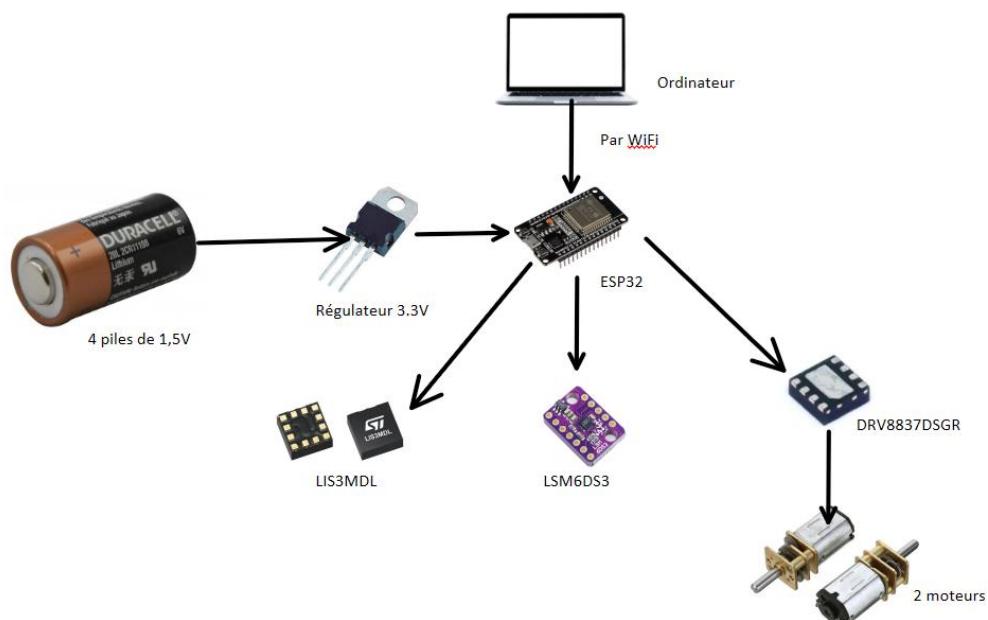


Figure 4 : Architecture matérielle

C. Architecture logicielle

Le programme embarqué sur l'ESP32 permet de contrôler le robot via une interface web en Wi-Fi. L'utilisateur peut envoyer des commandes pour faire avancer le robot, exécuter la figure de l'escalier ou orienter le robot vers le Nord magnétique puis faire une rose des vents.

Les moteurs sont pilotés par PWM et corrigés en temps réel grâce à deux correcteurs PID (ligne droite et rotation). Les encodeurs mesurent la distance parcourue, et le magnétomètre LIS3MDL permet de détecter l'orientation absolue 0°. L'ensemble assure un mouvement précis, fluide et autonome.

Algorigramme séquence 1 : escalier

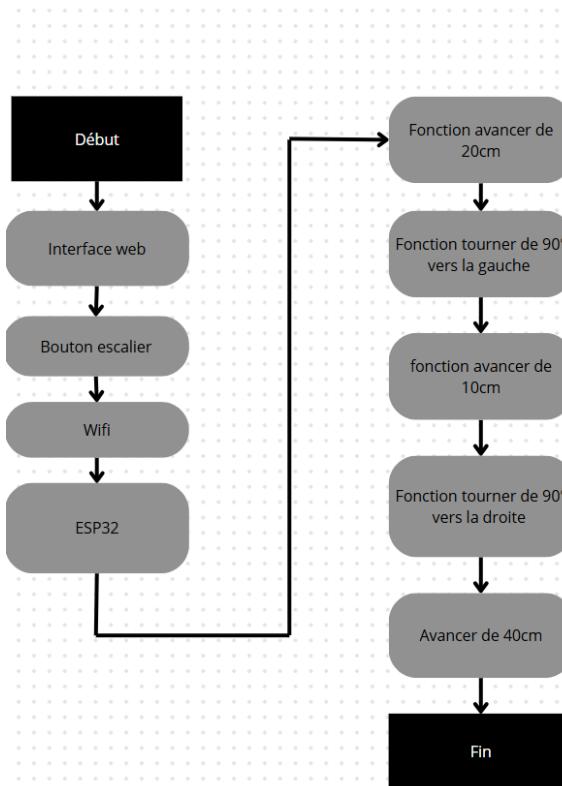


Figure 5 : Algorigramme de la première séquence, faire un escalier

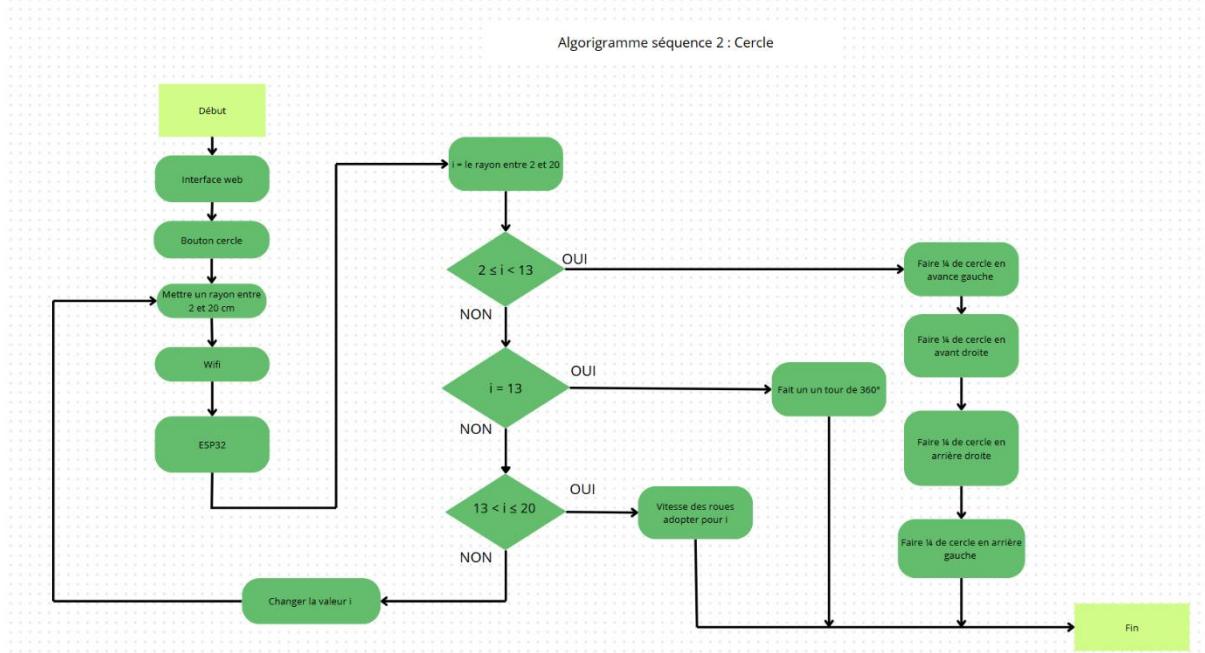


Figure 6 : Algorigramme de la deuxième séquence, faire un rond

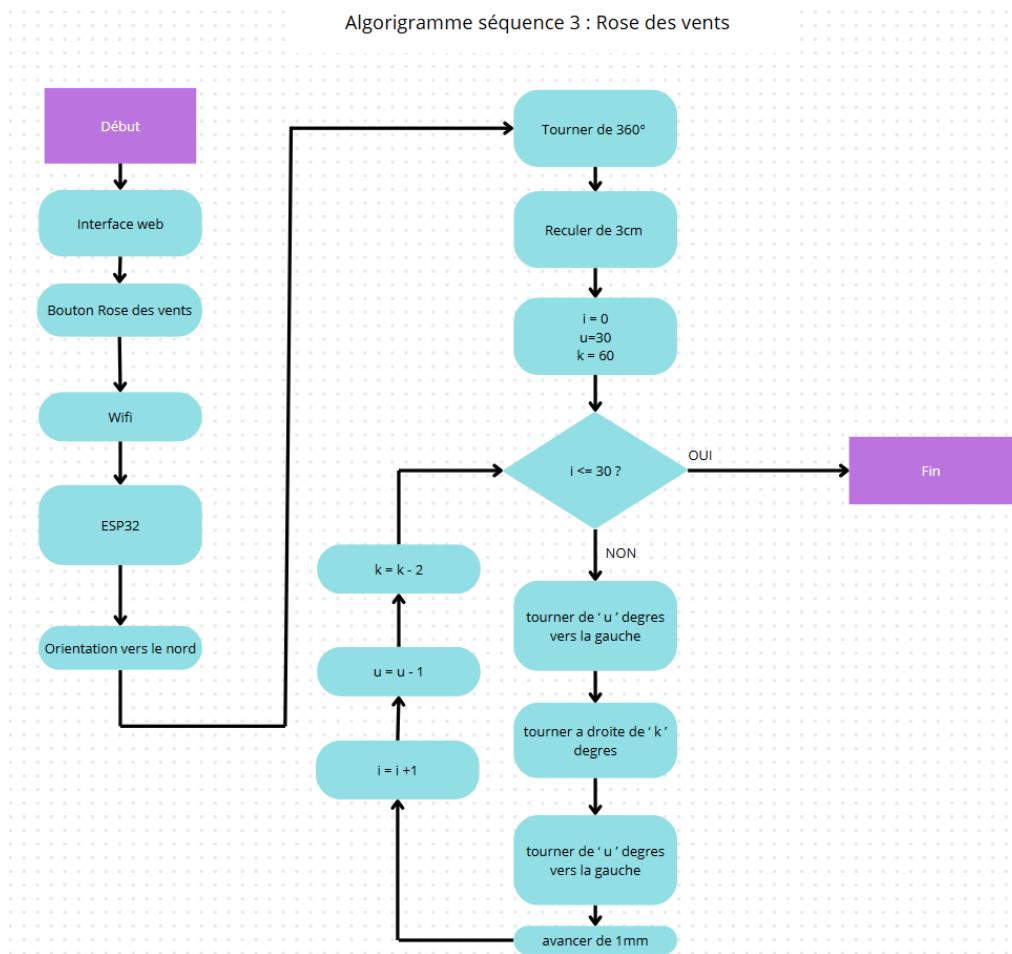


Figure 7 : Algorigramme de la troisième séquence, trouver le nord puis faire une rose des vents

VI. Développement

A. Module 1 - Interface Wifi Bluetooth

Pour permettre une communication entre l'utilisateur et le robot, nous avons choisi d'utiliser la connexion Wi-Fi intégrée à la carte ESP32. Ce choix s'est imposé car le Wi-Fi est directement supporté par l'ESP32, sans besoin de module externe, contrairement à l'Arduino Nano.

Nous avons opté pour une interface web : un petit serveur HTTP est hébergé sur l'ESP32, permettant à l'utilisateur d'entrer des commandes (par exemple : « d'avancer de 5 cm ») via une simple page HTML accessible depuis n'importe quel appareil connecté au même réseau. Cela rend le système très souple.

Le Bluetooth, bien que disponible sur l'ESP32, a été écarté pour ce projet car il nécessite souvent des applications spécifiques sur le téléphone, et la portée est plus limitée. Le Wi-Fi, lui, offre une meilleure portée, une plus grande compatibilité, et une interface plus ergonomique via navigateur.

En résumé, le choix du Wi-Fi permet :

- Une interface conviviale via navigateur web,
- Une mise en œuvre simple grâce aux bibliothèques intégrées à l'ESP32,
- Une souplesse d'utilisation sans installation d'application côté utilisateur.

B. Module 2 – Le déplacement du robot

Le déplacement repose sur deux motoréducteurs avec encodeurs intégrés. Chaque moteur est piloté indépendamment, ce qui permet au robot d'avancer, tourner ou effectuer des trajectoires complexes. Les encodeurs mesurent précisément la rotation des roues pour suivre la distance parcourue.

Pour assurer un déplacement précis et stable, nous avons implémenté un correcteur PID. Ce correcteur agit sur la vitesse des moteurs en comparant la consigne (vitesse ou position cible) à la valeur mesurée.

- Le correcteur proportionnel (K_p) augmente la réactivité du système. Trop élevé, il provoque des tremblements et de l'instabilité.
- Le correcteur intégral (K_i) corrige les erreurs résiduelles dans le temps. S'il est mal réglé, il peut causer un dépassement ou une perte de contrôle.
- Le correcteur dérivé (K_d) permet une anticipation et un freinage progressif, mais un K_d trop fort ralentit excessivement le robot.

Ce correcteur PID est appliqué en boucle pour ajuster en temps réel la puissance envoyée aux moteurs, garantissant ainsi une trajectoire fidèle aux commandes.

Paramètre	Effet	Si trop haut
Kp	Puissance de réponse	Instabilité, tremblement
Ki	Corrige les erreurs persistantes	Overshoot, perte de contrôle
Kd	Freinage, anticipation	Ralentissement excessif

Figure 8 : Tableau de l'effet du PID pendant les rotations du robot

Paramètre	Effet	Si trop haut
Kp	Correction instantanée de l'écart entre les 2 roues	Oscillation, signe d'instabilité
Ki	Correction d'un écart persistant dans le temps	Dépassemens, oscillations
Kd	Freine la correction, évite le dépassement	Freinage excessif, réaction lente

Figure 9 : Tableau de l'effet du PID pendant les lignes droites du robot

Afin de bien nous rendre compte de la différence de vitesse entre les deux moteurs, nous avons effectué des codes permettant de voir les vitesses des roues sur teleplot.

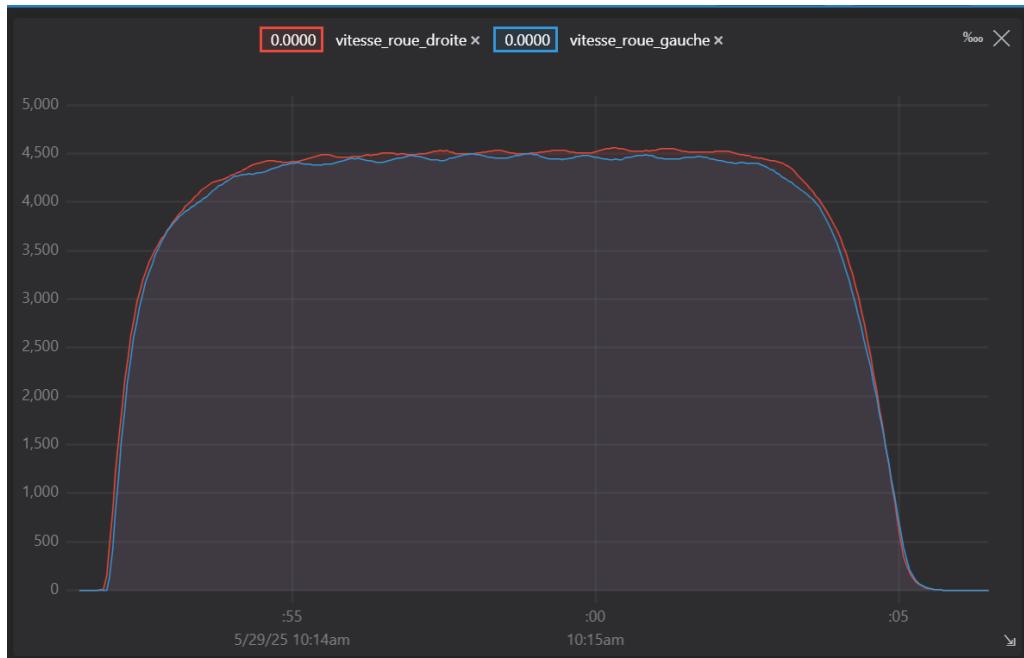


Figure 10 : Graphe montrant la vitesse des deux roues sur teleplot.

Comme nous pouvons le voir sur la figure ci-dessus, la vitesse de rotation des deux roues n'est pas identique alors qu'il possède le même signal PWM en entrée.

Nous allons donc devoir corriger cela à l'aide d'un correcteur PID.

C. Module 3 – Séquence 1 : L'escalier

La séquence de l'escalier est la première figure imposée de notre projet. Elle a pour but de valider la capacité du robot à exécuter une série de mouvements avec précision. Le tracé consiste à avancer avec précision de 20cm, effectuer un trait perpendiculaire vers la gauche d'une longueur de 10cm, puis un dernier trait de 40cm perpendiculaire au trait de 10cm vers la droite.

Pour cela on va utiliser :

- Les encodeurs, pour mesurer avec précision la distance parcourue grâce aux tics,
- Le correcteur PID va nous permettre de stabiliser notre robot et de lui permettre de tracer un trait parfaitement droit.

Les mouvements sont découpés en étapes, chacune étant exécutée après validation de la précédente. Ce fonctionnement en enchaînement contrôlé est essentiel pour garantir que la figure finale respecte les tolérances exigées : ± 1 cm pour les longueurs et $\pm 5^\circ$ pour les angles.

D. Module 4 – Séquence 2 : Le cercle

La séquence du cercle vise à démontrer la capacité du robot à suivre une trajectoire courbe de manière fluide et précise. Le rayon du cercle est un paramètre envoyé via l'interface Wi-Fi, ce qui ajoute une dimension dynamique à l'exécution.

Pour réaliser cette forme, le robot doit avancer tout en tournant, ce qui implique de faire tourner les roues à des vitesses différentes. La roue intérieure au cercle tourne plus lentement que la roue extérieure. Cette différence est calculée en fonction du rayon demandé et de l'empattement du robot.

Les éléments techniques mobilisés sont :

- Les encodeurs, qui assurent le suivi des distances parcourues par chaque roue,
- Le correcteur PID, qui régule précisément la vitesse des moteurs pour maintenir le bon ratio entre les deux roues, garantissant une courbure régulière.

Le critère de réussite est la fermeture du cercle avec une tolérance de 5 mm entre le point de départ et d'arrivée. Plusieurs tests sont réalisés avec des rayons différents (2 à 20 cm) pour valider la robustesse de l'algorithme.

Cette séquence met en valeur le bon équilibrage entre commande moteur, retour capteur et régulation, éléments essentiels pour obtenir un tracé propre et précis.

E. Module 5 – Le magnétomètre



Figure 11 : Magnétomètre LIS3MDL

Le magnétomètre LIS3MDL est utilisé pour déterminer l'orientation du robot par rapport au Nord magnétique. Ce module est essentiel pour la séquence de la rose des vents, qui exige que le robot trace une flèche orientée précisément vers le pôle Nord.

Le magnétomètre communique avec l'ESP32 via le bus I2C, ce qui permet une lecture régulière des champs magnétiques en X, Y et Z. Seuls les champs X et Y sont utilisés pour calculer l'angle absolu par rapport au Nord, selon une formule arc tangente : $\text{atan2}(x_{\text{cal}}, -y_{\text{cal}}) * 180 / \pi$, avec x_{cal} et $-y_{\text{cal}}$ des variables indiquant un champ magnétique après traitement et calibration.

Les choix techniques suivants ont été faits :

- Le capteur est fixé au robot, ce qui permet une mesure directe de son orientation,
- Une phase de calibration est réalisée pour compenser les perturbations magnétiques locales,
- Les mesures sont lissées pour éviter les sauts brusques d'orientation dus au bruit et aux interférences.

F. Module 6 – Séquence 3 : La rose des vents

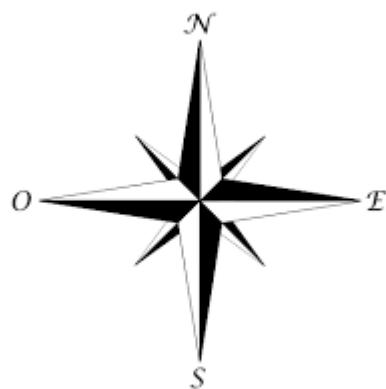


Figure 12 : Image d'une rose des vents

Cette séquence a pour objectif de faire tracer au robot une flèche orientée vers le Nord magnétique. Elle mobilise particulièrement le magnétomètre LIS3MDL, indispensable pour détecter la direction absolue du Nord.

Avant de commencer le tracé, le robot lit les données du magnétomètre (axes X et Y) pour calculer son orientation actuelle. Si nécessaire, il effectue une rotation jusqu'à ce que son axe soit aligné avec le Nord. Cette réorientation garantit que la flèche soit bien dirigée, peu importe la position initiale du robot.

Une fois orienté, le robot trace une flèche, en pointant le Nord, comparable à un triangle et exprime avec les autres points cardinaux (E, S, O) ainsi que les intermédiaires (NE, SE, SO, NO) avec des tirets sur un cercle. La qualité du tracé dépend de :

- La précision de l'orientation grâce au magnétomètre,
- La stabilité du mouvement assurée par les encodeurs et le correcteur PID,
- L'efficacité du découpage des étapes de dessin (pointe, cercle et tirets).

Cette séquence est la plus symbolique du projet, car elle relie directement le robot à son environnement réel. Elle démontre l'intégration réussie entre capteurs, commande moteur et logique de navigation spatiale.

G. Module 7 – Conception de la coque

Afin de finaliser au mieux ce projet, nous avons fait le choix de concevoir une coque pour le robot.

Nous allons utiliser Fusion 360, qui est un logiciel de modélisation 3D fourni par l'école.



Figure 13 : Logo du logiciel utilisé pour la conception 3D

VII. Tests et validation

A. Module 1 - Interface Wifi Bluetooth

Dans ce module, nous avons développé une interface web hébergée directement sur l'ESP32. Cette interface est accessible via un navigateur web en se connectant au réseau Wi-Fi du robot. Elle permet à l'utilisateur de saisir une commande à distance, comme la distance à parcourir, ou de déclencher des fonctions prédéfinies.

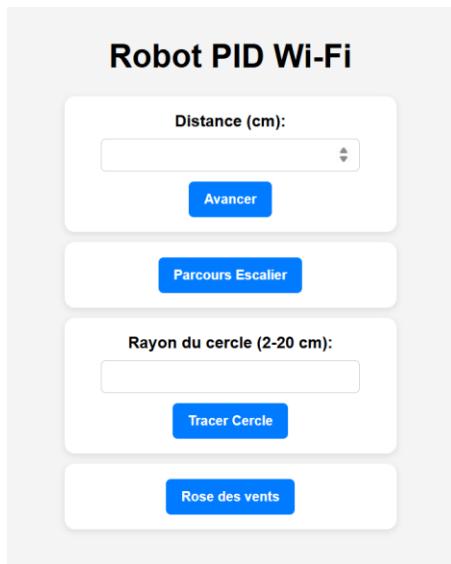
L'objectif de ce test était de valider le bon fonctionnement de la communication entre l'utilisateur et l'ESP32 via Wi-Fi, ainsi que le bon affichage et la réactivité de la page web.

Les critères de réussite étaient les suivants :

- La page web doit s'afficher correctement sur différents navigateurs (Chrome, Firefox, Safari...).
- Tous les éléments HTML (zone de texte, boutons) doivent s'afficher correctement et être interactifs.
- Les requêtes HTTP générées par l'interface (ex : clic sur "Avancer") doivent bien être reçues par l'ESP32.
- Chaque action déclenchée depuis l'interface doit être interprétée correctement par le microcontrôleur.

Les tests ont été effectués avec succès :

- La page web s'affiche instantanément sur smartphone et ordinateur dès la connexion au réseau du robot.
- Les boutons sont fonctionnels et visuellement bien intégrés (voir Image 8).
- À chaque clic, une requête HTTP est envoyée et bien reçue sur le port série, ce qui confirme la communication.
- L'ESP32 réagit à chaque commande reçue, sans blocage ni délai perceptible.



The screenshot shows a web-based control interface titled "Robot PID Wi-Fi". It consists of three main sections:

- Top Section:** A form field labeled "Distance (cm)" with a dropdown arrow, followed by a blue "Avancer" button.
- Middle Section:** A blue "Parcours Escalier" button.
- Bottom Section:** A form field labeled "Rayon du cercle (2-20 cm)" with a dropdown arrow, followed by a blue "Tracer Cercle" button, and a blue "Rose des vents" button below it.

Figure 14 : Interface web

En conclusion, le module 1 est validé.

L'interface web est simple, ergonomique, compatible avec tous les appareils testés, et assure une communication fiable entre l'utilisateur et le robot. Elle constitue une base solide pour piloter les autres modules du projet.

B. Module 2 - Le déplacement du robot

Ce module porte sur les capacités de déplacement du robot, rendues possibles grâce à deux motoréducteurs avec encodeurs intégrés. Chaque moteur peut être piloté indépendamment, permettant au robot d'avancer en ligne droite, de tourner sur place ou d'effectuer des trajectoires plus complexes. Les encodeurs mesurent précisément les rotations des roues, ce qui permet de suivre la distance parcourue et d'effectuer des ajustements en temps réel.

L'objectif des tests était de valider :

- La réactivité du système moteur (démarrage, arrêt, inversion),
- La régularité de l'avancement sur différentes distances,
- La capacité du robot à suivre une trajectoire droite,
- La précision des rotations angulaires,
- Et la stabilité globale du déplacement, même après plusieurs changements de consigne.

Pour améliorer la précision et compenser le jeu des moteurs ainsi que les erreurs de vitesse entre les deux moteurs, un correcteur PID a été ajouté. Ce correcteur ajuste dynamiquement la vitesse appliquée aux moteurs, en fonction des écarts mesurés par les encodeurs.

Les tests ont montré que :

- Le robot répond rapidement aux commandes,
- La distance parcourue correspond à la consigne avec une erreur inférieure à $\pm 1\text{cm}$ sur sol plat,
- Les rotations sont précises à $\pm 5^\circ$ près après ajustement du PID,
- Les trajectoires en ligne droite sont bien maintenues grâce à la correction continue des écarts de vitesse entre les roues.

Les tableaux (image 8 et image 9) de la partie Développement présentent les effets respectifs des coefficients du PID pendant les déplacements. Ils ont été utilisés comme guide pour affiner les réglages et éviter les comportements indésirables comme les tremblements, les dépassements ou les ralentissements excessifs. Nous obtenons donc des valeurs pour le correcteur PID en ligne droite (1.5, 0.0, 0.8) et en rotation (0.8, 0.0, 1.2).

Comme nous l'avons vu précédemment à la figure 10, les moteurs présentent des imperfections. Nous avons précédemment tracé sur Teleplot les vitesses des deux moteurs et constaté de nombreuses erreurs.

Nous avons écrit un nouveau code afin de vérifier les valeurs du PID avec Teleplot, et nous pouvons dire que nos résultats sont concluants.



Figure 15 : Courbe montrant la vitesse de chaque moteur avec un PID.

Comme nous pouvons le remarquer sur le graphe de la figure 15, les courbes de vitesse des moteurs gauche et droit se superposent, ce qui témoigne de l'efficacité du PID.

En conclusion, le module de déplacement est **validé**. Le robot se déplace de façon fluide, stable et suffisamment précise pour exécuter les parcours demandés. Le recours aux encodeurs et à la correction PID contribue fortement à cette fiabilité, sans surcharger le système.

C. Module 3 - Séquence 1 : L'escalier

La séquence de l'escalier a été conçue pour valider la capacité du robot à enchaîner plusieurs segments rectilignes avec des changements de direction précis. Le robot doit exécuter un parcours en trois étapes : avancer de 20 cm, tourner à gauche pour un segment de 10 cm, puis tourner à droite pour une ligne finale de 40 cm. Ces virages ne sont pas réalisés par rotation sur place, mais en ajustant la vitesse relative des roues pour produire des trajectoires perpendiculaires. Le PID de rotation est prévu pour stabiliser l'orientation pendant ces transitions.

Afin d'évaluer la stabilité directionnelle du robot en ligne droite, une acquisition temporelle du cap a été réalisée. Comme le montre la figure 16 ci-dessous, la direction reste comprise entre 87° et 93° durant le déplacement, ce qui démontre un bon maintien de cap. Les légères oscillations sont naturelles et restent bien contenues, signe que le système corrige efficacement les petits écarts sans dérive marquée.

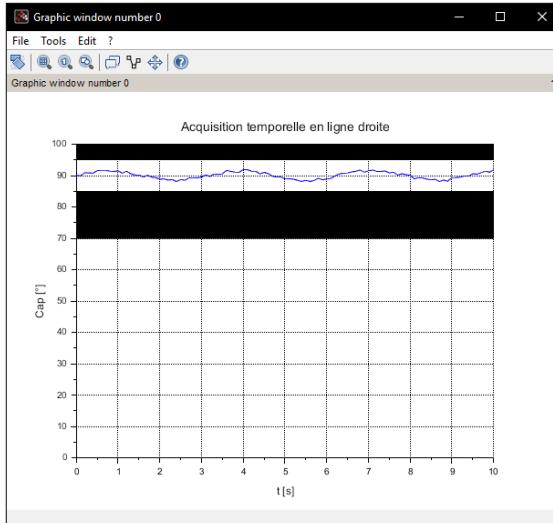


Figure 16 : Cap du robot lors d'une ligne droite

Pour vérifier la précision des changements de direction, une série de commandes angulaires a été envoyée au robot, et les angles effectivement réalisés ont été mesurés. Le graphe de l'image 17 compare l'angle demandé à l'angle mesuré. La forte linéarité des points, proches de la droite idéale (trait pointillé), indique une excellente corrélation entre la commande et la réalité. Cela confirme que les virages sont bien maîtrisés, avec une erreur angulaire inférieure à $\pm 5^\circ$, même sans rotation sur place.

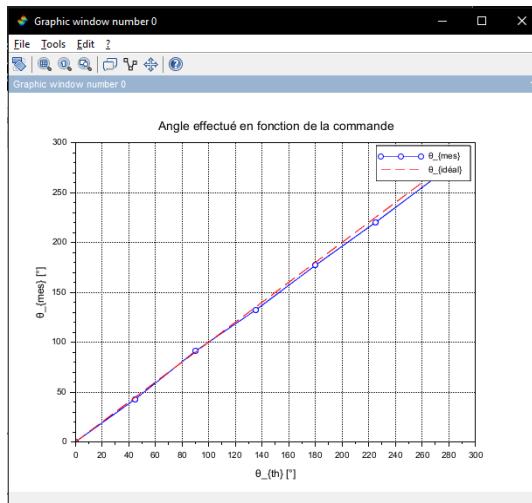


Figure 17 : Angle effectué en fonction de la commande

Enfin, la précision des longueurs parcourues a été évaluée en comparant la distance théorique demandée à celle mesurée par les encodeurs. L'image 18 montre une courbe parfaitement linéaire entre 0 et 70 cm, avec des erreurs négligeables (inférieures à ± 1 cm). Cela valide la fiabilité du comptage des tics pour mesurer des segments courts comme longs.

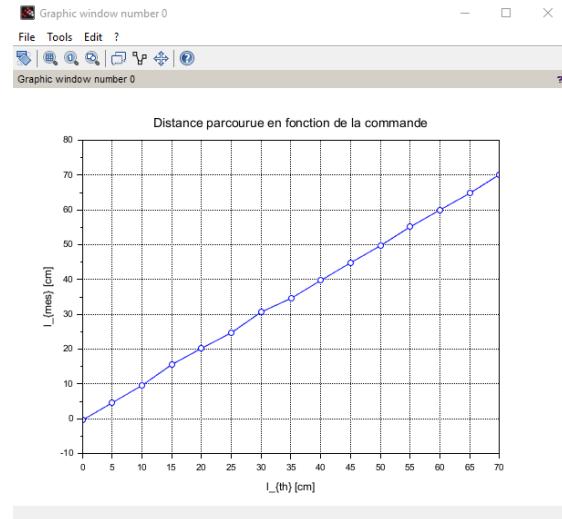


Figure 18 : Distance parcourue par le robot en fonction de la commande

Ci-dessous, nous pouvons voir en photo le tracé du robot avec les 3 différents segments et leurs mesures.

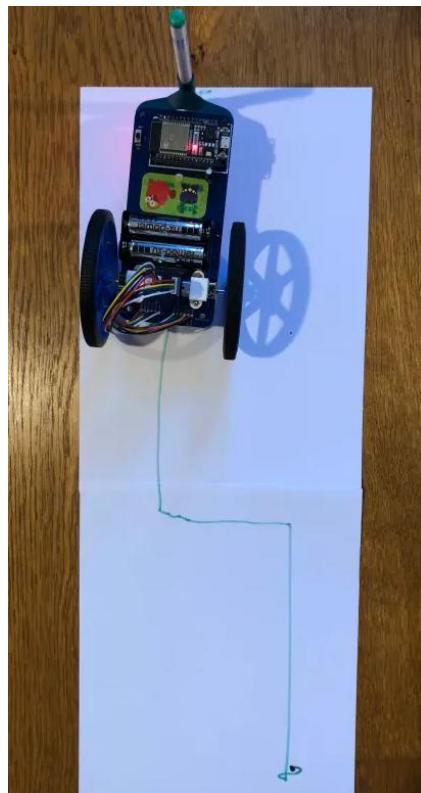


Figure 19 : Parcours escalier vue générale

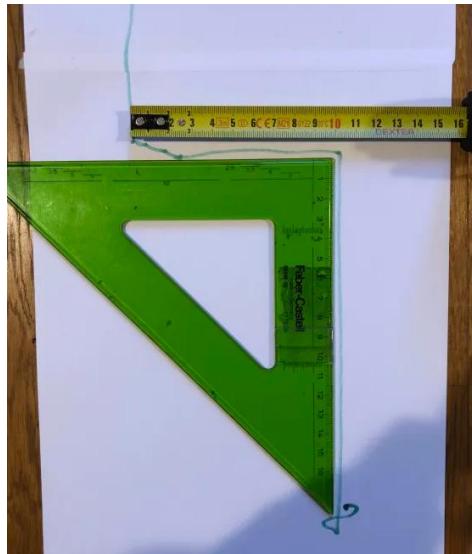


Figure 20 : Vérification de l'angle et de la distance du trait intermédiaire

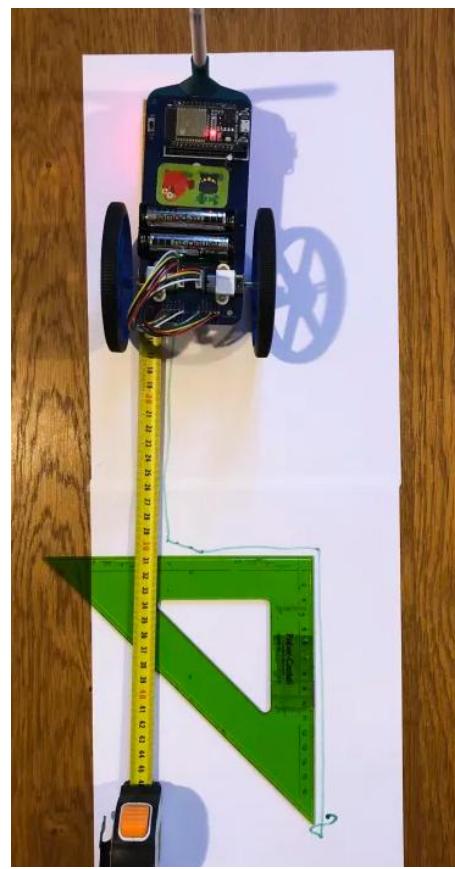


Figure 21 : Distance mesurée du troisième segment

En conclusion, le robot est capable d'enchaîner des segments perpendiculaires avec une bonne précision en distance comme en orientation. La séquence de l'escalier a été réalisée avec succès, ce qui permet de considérer le module comme **validé**.

D. Module 4 - Séquence 2 : Le cercle

Pour la séquence 2, nous avons développé un programme permettant au robot de tracer une trajectoire circulaire, avec un rayon paramétrable directement depuis l'interface web du site. L'utilisateur saisit le rayon souhaité, qui est ensuite transmis au robot via le Wi-Fi.

Le sous-programme dédié au tracé du cercle récupère la valeur du rayon entrée par l'utilisateur, puis adapte la commande envoyée aux moteurs en fonction de ce paramètre :

- Si le rayon demandé est inférieur à 13 cm, le robot exécute une séquence spécifique, illustrée par la figure 2A (voir annexe), permettant de tracer la plus petite courbe possible.
- Si le rayon demandé est exactement de 13 cm, le robot réalise un cercle parfait en pivotant sur lui-même : la roue intérieure tourne à la même vitesse que la roue extérieure, mais en sens inverse. Ce mouvement permet au robot d'effectuer un cercle de pile 13 cm de rayon.
- Si le rayon souhaité est supérieur à 13 cm, le programme calcule automatiquement la vitesse que doit avoir la roue extérieure par rapport à la roue intérieure afin de respecter précisément le rayon demandé.

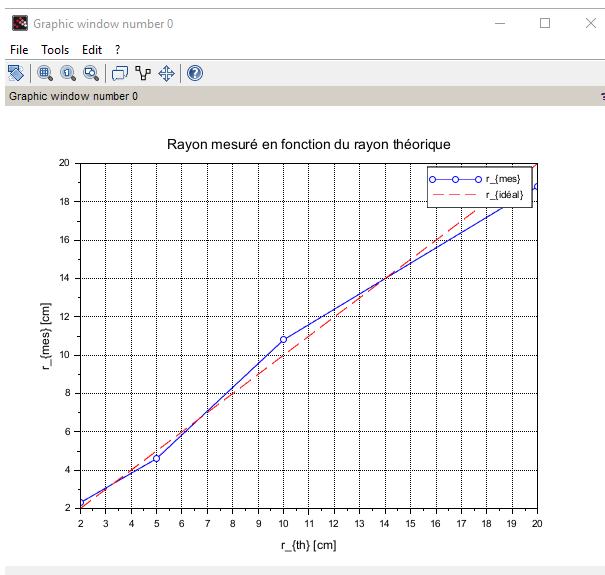


Figure 22 : Rayon d'un cercle mesuré en fonction de la commande

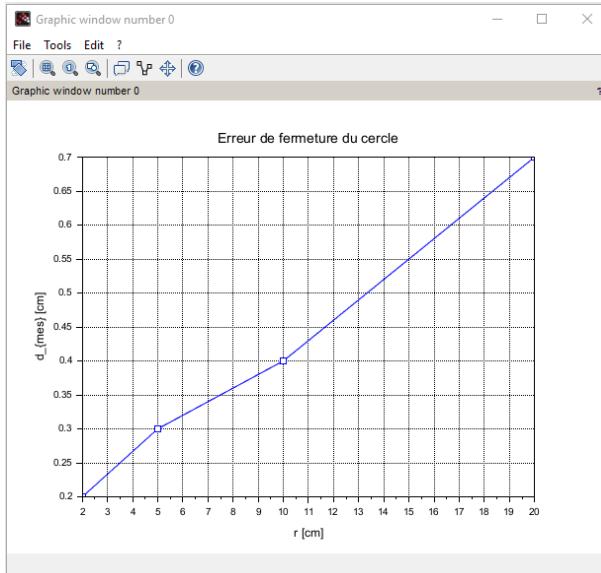


Figure 23 : Erreur entre le point de départ et le point d'arrivé dans un cercle

Les deux courbes ci-dessus nous permettent d'obtenir une estimation de la précision du tracé du cercle réalisé par le robot.

Le premier graphe montre que la valeur du rayon commandé est quasiment identique à celle du rayon mesuré. Cela nous permet de valider l'efficacité du contrôle en boucle fermée. Tout de même, il subsiste toujours de légères erreurs.

Le second graphe illustre l'erreur de fermeture parfaite du cercle, c'est-à-dire la distance entre le point de départ et le point d'arrivée. Sur l'ensemble des tests, cette erreur est restée inférieure à 1 cm.

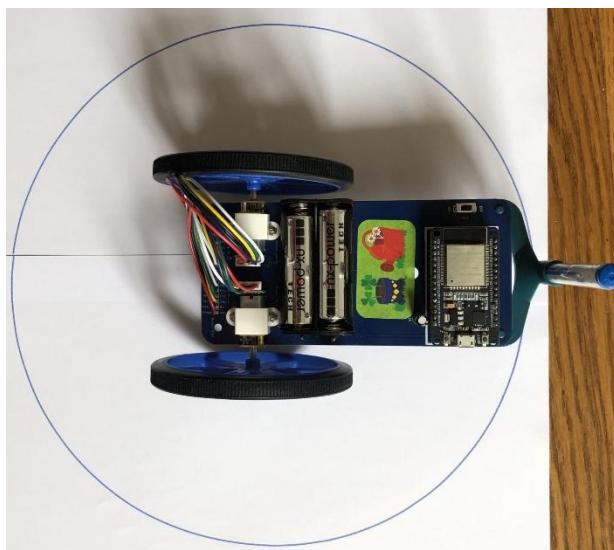


Figure 24 : Tracé du cercle par le robot.

Pour finir, ces résultats sont presque identiques à ceux de nos simulations avec le robot, et confirment le bon fonctionnement du système. Cette séquence est bien **validée**.

E. Module 5 - Le magnétomètre

Le magnétomètre nous a permis de déterminer l'orientation du robot par rapport au Nord magnétique. Cette fonctionnalité est indispensable pour les figures orientées, comme la rose des vents.

Après calibration, le capteur fournit un angle stable à partir des composantes X et Y du champ magnétique. Le robot peut ainsi détecter précisément sa direction et se réorienter automatiquement avec des rotations dans un sens et dans l'autre à vitesse lente pour être le plus précis possible.

Les objectifs atteints sont :

- Détection fiable du Nord avec une précision inférieure à $\pm 10^\circ$,
- Intégration complète du capteur avec l'ESP32 via I2C,
- Stabilité des mesures après filtrage.

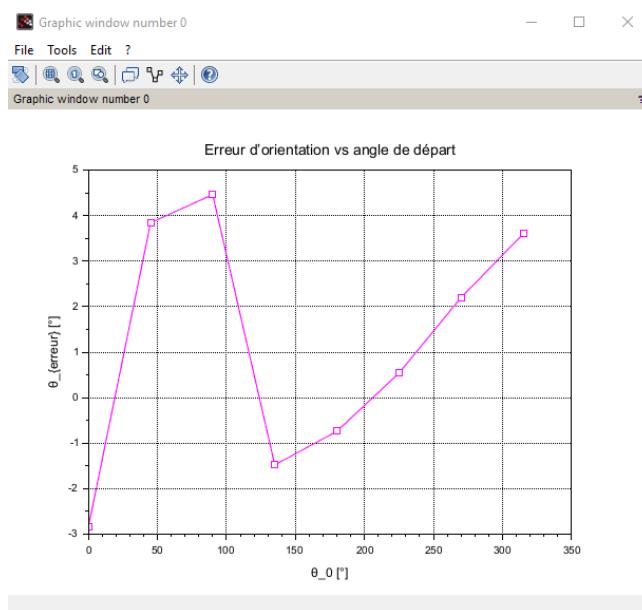


Figure 25 : Angle d'orientation du robot au début et après

Le graphique ci-dessus montre l'évolution de la correction de l'orientation du robot par rapport au Nord magnétique.

Et ci-dessous nous pouvons voir une image du robot, d'après l'image le robot a commencé en étant approximativement au Sud puis s'est corrigé une fois, est allé trop loin, est revenu dans l'autre sens, est encore allé un peu loin, puis s'est arrêté en pointant le Nord. On peut voir où est le Nord avec la boussole jaune en haut à gauche de l'image qui est dirigée vers le Nord avec l'aiguille rouge.

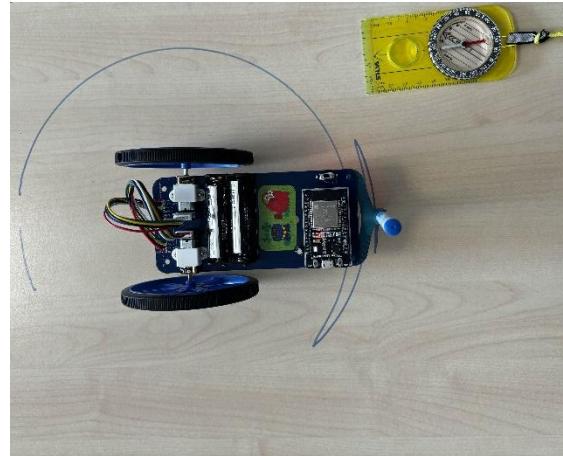


Figure 26 : Démonstration de l'utilisation du magnétomètre pour trouver le Nord

Les tests confirment que l'orientation est bien détectée, même dans un environnement perturbé. Le magnétomètre est donc opérationnel et **validé**.

F. Module 6 - Séquence 3 : La rose des vents

Pour cette dernière séquence, notre objectif était de faire tracer au robot une flèche pointant vers le Nord magnétique, puis une rose des vents stylisée. Cette figure combinait plusieurs compétences clés du projet : détection de l'orientation, rotation contrôlée, et tracé précis.

Grâce au magnétomètre LIS3MDL et à la validation du module 5, le robot détecte sa direction et effectue une rotation jusqu'à être aligné avec le Nord. Une fois orienté, il trace une flèche, puis complète la figure en tournant sur lui-même et faisant des marques représentant les autres points cardinaux.

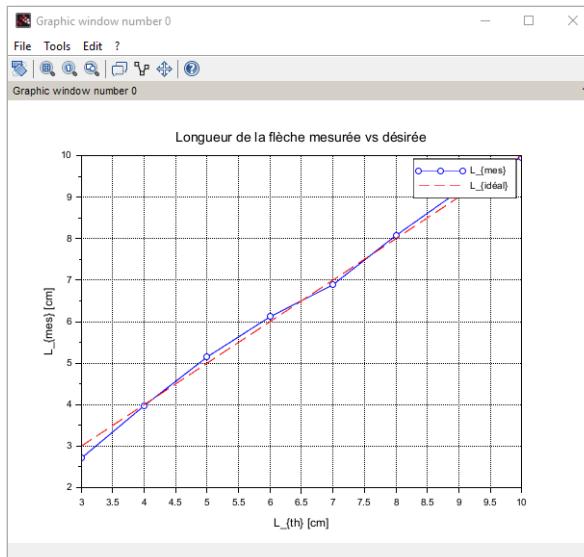


Figure 27 : Graphique Scilab montrant la précision du robot pour tracer une flèche

Objectif non atteint :

- Remplissage de la flèche

Malgré de nombreux essais pour remplir l'intérieur de la flèche de la rose des vents, le résultat le plus proche que nous avons atteint est l'image 28 ci-dessous. Nous avons retiré cette partie du code car elle ne répond pas au cahier des charges qui impose un remplissage d'au moins 80% de la surface de la flèche. Nous laissons donc une preuve dans ce rapport de l'essai le plus proche que nous ayons obtenu pour cette fonctionnalité.

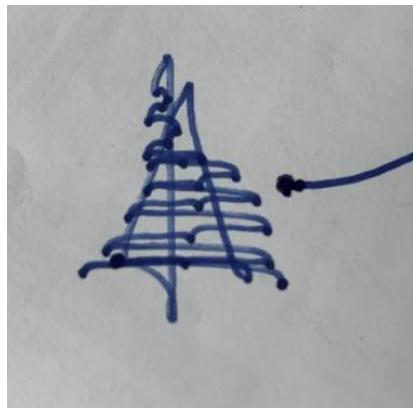


Figure 28 : Essai du remplissage de la flèche

Objectifs atteints :

- Alignement vers le Nord avec une précision inférieure à $\pm 10^\circ$,
- Tracé propre et stable de la flèche et des repères,
- Coordination réussie entre capteurs et moteurs.

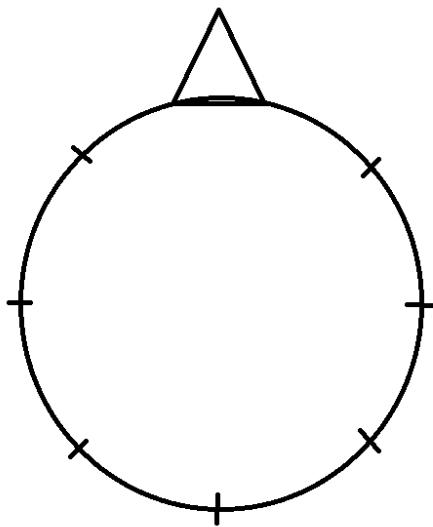


Figure 29 : Objectif de rose des vents

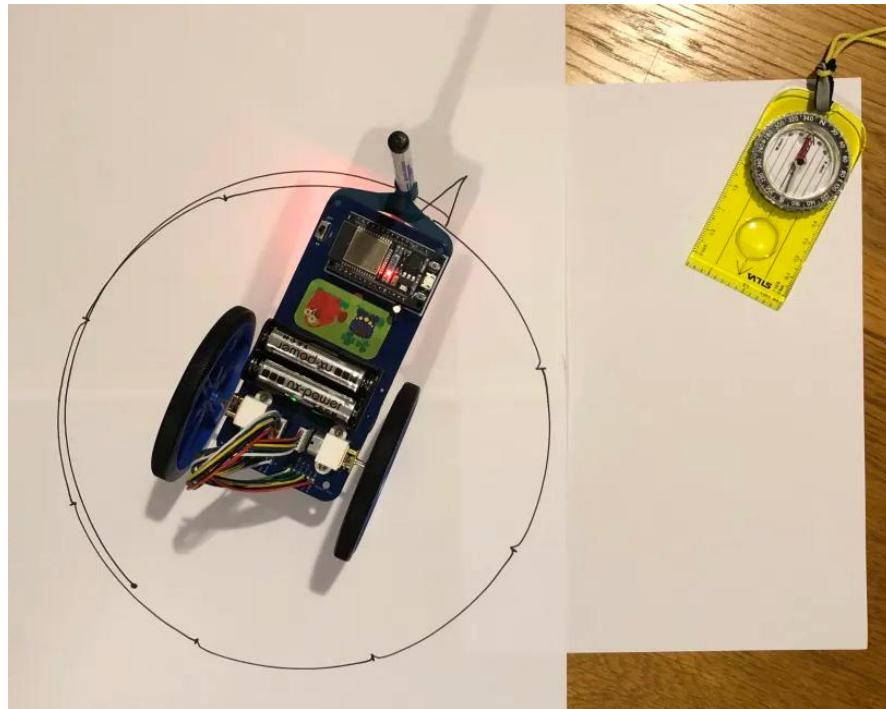


Figure 30 : Résultat obtenu pour la rose des vents

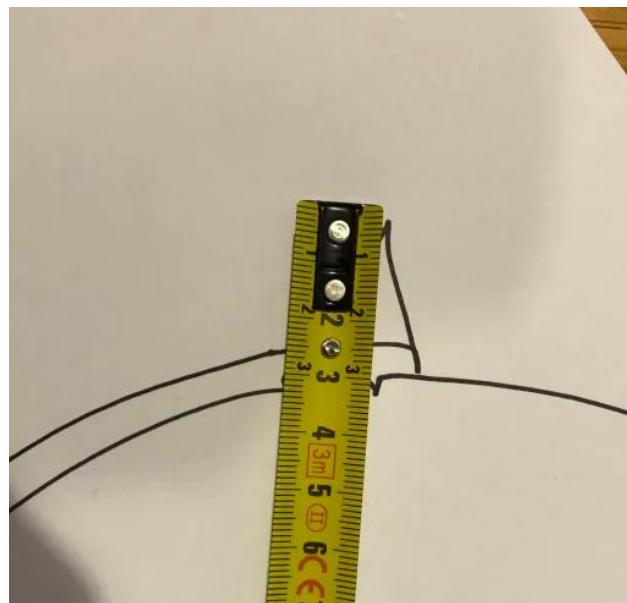


Figure 31 : Mesure montrant la précision du robot à tracer le cercle avec une différence maximum de 5mm entre le point de départ et d'arrivé

Cette séquence représente bien l'aboutissement du projet : elle montre que notre robot est capable de s'adapter à son environnement, de s'orienter et de dessiner avec autonomie. Nous considérons cette séquence comme **validée**.

G. Module 7 - La conception de la coque

Comme nous l'avons vu dans la première partie, nous avons utilisé fusion360.

Afin de concevoir au mieux une coque nous avons pris les mesures du robot et nous avons commencé à dessiner une esquisse sur le logiciel.

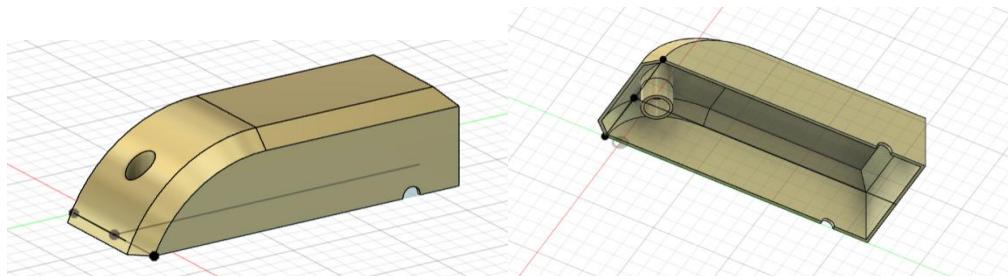


Figure 32 : Coque conçue sur mesure.

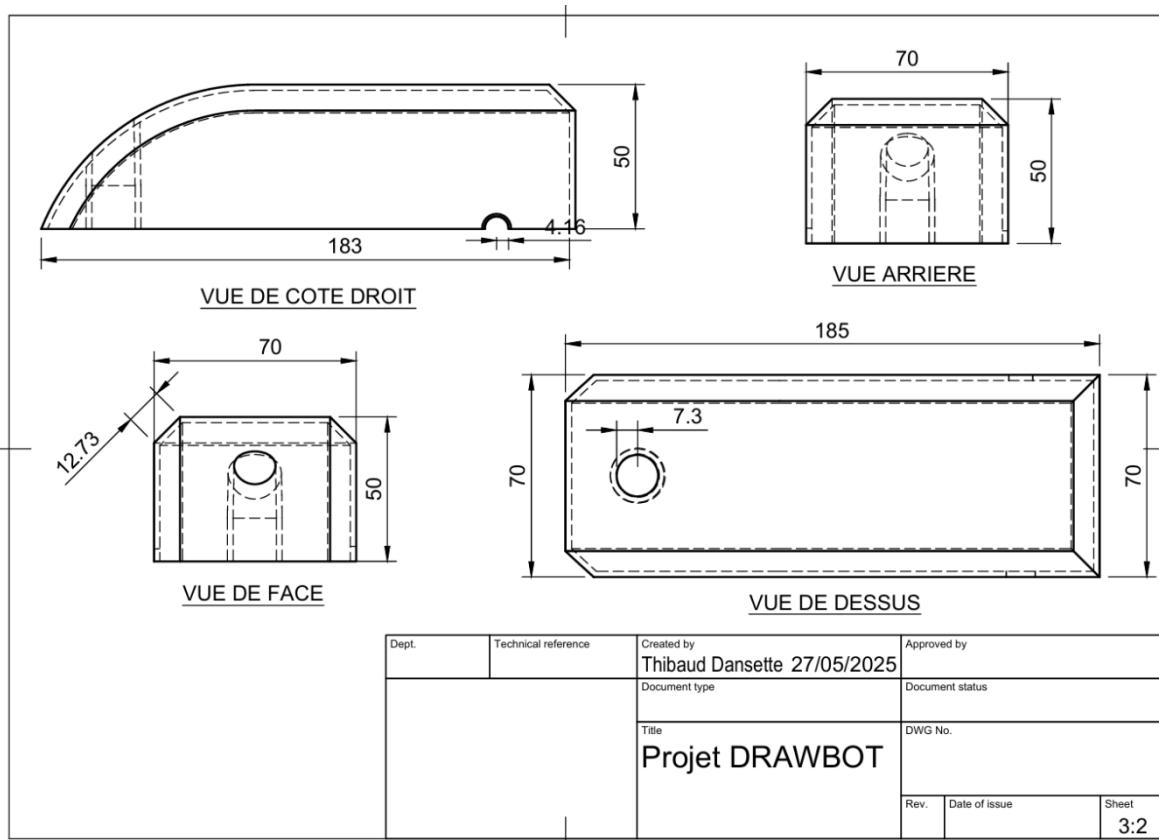


Figure 33 : Plan de la coque

Sur la figure ci-dessus nous pouvons voir le plan de la coque ainsi que ses mesures.

VIII. Bilan

A. État d'avancement

A l'heure actuelle le projet est à un stade très avancé, terminé si l'on se base sur les modules de base. Les objectifs qui étaient de réaliser les 3 parcours et une fonctionnalité avancée, dans notre cas la fonctionnalité "rose des vents" sont accomplis.

Si nous voulions être perfectionnistes, les 3 parcours seraient à retravailler pour être les plus parfaits et optimisés possible mais ce que nous avons actuellement est largement satisfaisant.

B. Pertinence de la solution technique

Lors de ce projet, nous avons rencontré quelques limites jusqu'à présent comme la faible puissance des moteurs qui ne peuvent pas tourner à une vitesse trop faible ce qui aurait pu être utile pour effectuer des déplacements très lents et précis. Le deuxième point est le fait que le feutre pour dessiner est excentré de l'axe des roues ce qui complique grandement la logique de déplacement du robot pour effectuer des dessins. Idéalement le feutre aurait dû être placé dans le même axe que les roues pour pouvoir effectuer des angles parfaitement droits ou avec l'angle que l'on souhaite en toute simplicité.

Quelles sont les possibilités d'évolution ou de poursuite ?

Sur le plan de notre robot, nous n'y voyons pas directement d'évolution à part changer certains composants ou la forme générale du Drawbot. Sur le plan du projet en équipe nous pourrions, à l'avenir, utiliser davantage d'outils pour l'organisation des tâches et mieux respecter les dates limites pour terminer les différents modules.

C. Bilan sur le travail d'équipe

Vous trouverez ci-dessous, le bilan personne de chaque personne de l'équipe ainsi qu'un plan d'action pour le prochain projet.

Membres de l'équipe	Bilan personnel
Erwan Planet	Malgré un sujet manquant de précisons, ce fut un projet intéressant sur le plan de la recherche d'informations et de la programmation avec des boucles fermées permettant de contrôler avec précision un robot. J'ai également pu renforcer mes compétences en programmation C++.
Hugo Faivre	Ce projet m'a permis de renforcer mes compétences en programmation embarquée, en régulation PID et en intégration capteurs-moteurs, des atouts majeurs pour un stage en robotique ou systèmes embarqués. Pour améliorer l'organisation, l'équipe pourrait mieux

	<p>documenter chaque étape du projet au fur et à mesure, afin de gagner du temps et éviter les incompréhensions.</p>
Thibaud Dansette	<p>J'ai beaucoup apprécié ce projet, que j'ai trouvé très instructif. Il nous a permis de mettre en œuvre toutes les connaissances que nous avons acquises durant ce semestre. Il m'a également permis de découvrir une nouvelle carte, l'ESP32, que je ne connaissais pas du tout et qui est très agréable à utiliser. Elle possède de meilleures fonctionnalités que la carte Arduino Nano.</p> <p>Ce projet m'a permis de m'améliorer en programmation ainsi que dans la réalisation de schémas avec KiCad.</p> <p>Je trouve que tous les projets que j'ai eus en électronique sont très intéressants et m'ont conforté dans mon choix de majeure, qui pourrait être les systèmes embarqués.</p>
Léo-Pol Bernard-Chaussée	<p>Ce projet m'a permis d'appliquer concrètement mes connaissances en automatique, en particulier sur le contrôle PID et les systèmes bouclés. J'ai beaucoup appris sur l'interaction entre la théorie et la pratique, notamment les écarts entre simulation et comportement réel du robot. Le travail en équipe m'a aidé à mieux structurer mes tâches et à coopérer efficacement dans un contexte technique. Ce projet m'a également donné envie de creuser davantage la robotique embarquée.</p>

Notre plan d'action pour le prochain projet :

1. Répartition claire des rôles dès le départ : chaque membre devra jouer un rôle précis (programmation, documentation...) défini lors de la première réunion. Cela évitera les zones floues et les doublons, et permettra à chacun de se concentrer sur ses responsabilités.
2. Mise en place d'un calendrier de travail partagé : utiliser un outil comme Google Agenda ou Trello pour fixer les deadlines intermédiaires, suivre les tâches en cours, afin de mieux répartir le travail sur la semaine.
3. Courtes réunions hebdomadaires obligatoires : se réunir 1 à 2 fois par semaine (20-30 minutes) pour faire le point sur l'avancement, les blocages éventuels, et redistribuer les tâches si besoin.
4. Archivage rigoureux de tous les fichiers : utiliser un Google Drive ou un GitHub commun pour centraliser tous les documents, versions de code, captures de tests, afin d'éviter les pertes de données ou les conflits de version. Cette solution serait préférable au dépôt de nos documents sur le fil de discussion directement.

IX. Bibliographie

Arduino Forum, « Reasoning for robot that draws a circle », Arduino Forum, 18 mai 2024. [En ligne]. Available: <https://forum.arduino.cc/t/reasoning-for-robot-that-draws-a-circle/1381212>. [Accès le 18 mai 2025].

UPESY, « Install ESP32 on Arduino IDE – Complete Guide », UPESY Tutorials, 2023. [En ligne]. Available: <https://www.upesy.fr/blogs/tutorials/install-esp32-on-arduino-ide-complete-guide>. [Accès le 30 avril 2025].

R. Santos, « ESP32 Bluetooth Classic with Arduino IDE », Random Nerd Tutorials, 2023. [En ligne]. Available: <https://randomnerdtutorials.com/esp32-bluetooth-classic-arduino-ide/>. [Accès le 02 mai 2025].

Adafruit, « Adafruit 9-DOF IMU Breakout - LSM6DSOX + LIS3MDL », Adafruit, 2024. [En ligne]. Available: <https://www.adafruit.com/product/4640#technical-details>. [Accès le 02 mai 2025].

Adafruit Forum, « How to use LSM6DSOX and LIS3MDL together », Adafruit Forums, 2022. [En ligne]. Available: <https://forums.adafruit.com/viewtopic.php?t=188417#top>. [Accès le 08 mai 2025].

M. Schwager, « Introduction to Nonholonomic Motion Planning », California Institute of Technology, 2007. [En ligne]. Available: <https://robotics.caltech.edu/wiki/images/4/4b/IntroNonholonomicMotionPlanning.pdf>. [Accès le 20 mai 2025].

X. Annexes

Documents annexes, éventuels codes (**pas de code dans le rapport**).

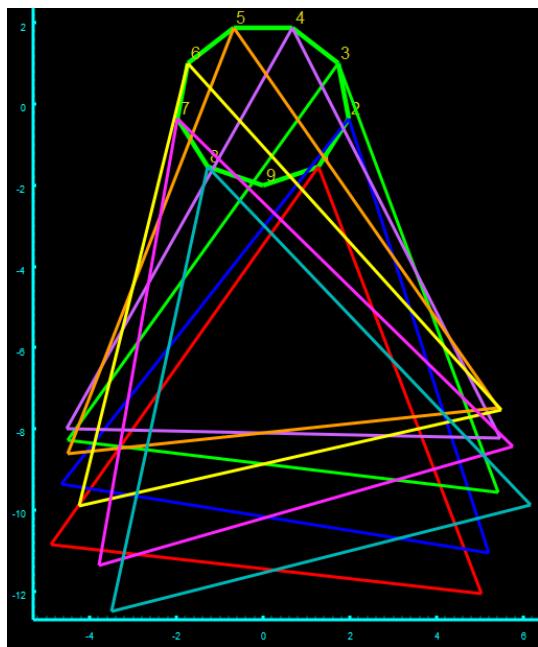


FIGURE 1A

$$(x - h)^2 + (y - k)^2 = r^2$$

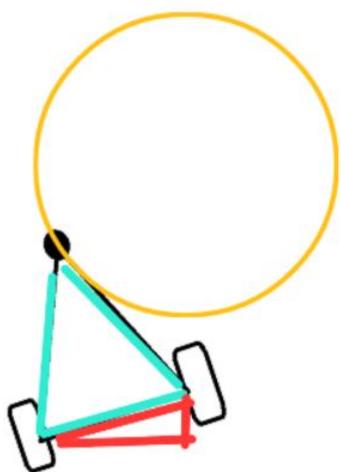


FIGURE 2A