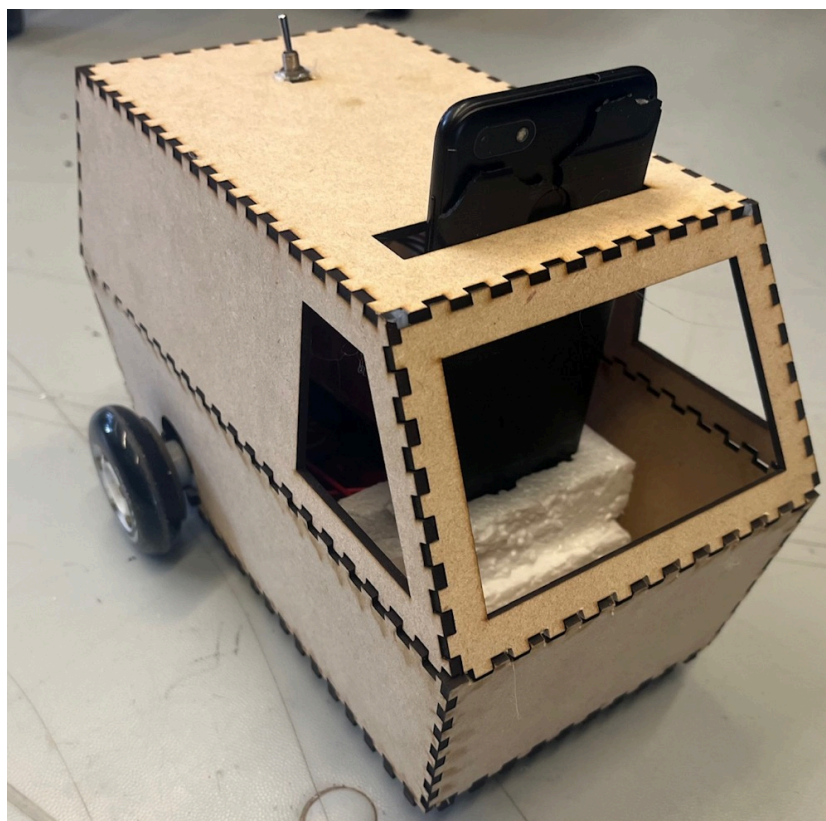


Projet de Physique P6
STPI/P6/2024 – #15

Robot mobile piloté par Wifi



Etudiants :

Camille AUDOIN

Pierre LYNCH

Loic SEBA

Victorien MERAND--MAUREL

Rayen ZOUAGHI

Enseignant-responsable du projet :

Fabrice DELAMARE

Date de remise du rapport : 14/06/2024

Référence du projet : STPI/P6/2024 – 15

Intitulé du projet : Robot mobile piloté par Wifi

Type de projet : Expérimental

Objectifs du projet :

L'objectif du projet est de réaliser un robot pouvant être contrôlé par wifi à l'aide d'un téléphone portable, d'une tablette ou d'un ordinateur. Notre groupe a également décidé d'intégrer les possibilités d'avoir un retour vidéo de l'environnement du robot ainsi que d'émettre des sons et de capter les sons environnants du robot.

Il sera nécessaire, pour mener à bien ce projet, de réaliser la conception du robot sur *Solidworks*, mais aussi de réaliser un programme *Arduino* pour le robot ainsi qu'une application *Android* via *MIT App Inventor*, et enfin de créer une carte électronique pour souder les composants.

Mots-clefs du projet : **Robotique, Électronique, Mécanique, Informatique**

TABLE DES MATIÈRES

1. INTRODUCTION.....	6
2. MÉTHODOLOGIE / ORGANISATION DU TRAVAIL.....	7
3. TRAVAIL RÉALISÉ ET RÉSULTATS.....	8
3.1. PARTIE MÉCANIQUE.....	8
3.1.1. Design.....	8
3.1.2. Modélisation 3D et fabrication.....	8
3.2. PARTIE ÉLECTRONIQUE.....	9
3.2.1. Description de la carte électronique.....	9
3.2.2. Émission et réception de sons.....	10
3.2.3. Soudures.....	10
3.2.4. Problèmes rencontrés.....	10
3.3. PARTIE INFORMATIQUE.....	12
3.3.1. Fonctionnement du robot.....	12
3.3.2. Programmation MIT App Inventor.....	12
3.3.3. Code Arduino.....	15
3.4. RÉSULTATS.....	16
4. CONCLUSION ET PERSPECTIVES.....	17
4.1. Conclusions sur le travail réalisé.....	17
4.2. Conclusions sur l'apport personnel de cet EC projet.....	17
4.3. Perspectives pour la poursuite de ce projet.....	18
5. BIBLIOGRAPHIE.....	19
6. ANNEXES.....	20
6.1. Documents.....	20
6.1.1. Planning prévisionnel de l'avancé du projet.....	20
6.1.2. Résumé des séances.....	20
6.2. Code Arduino.....	22
6.3. Schémas de montages, plans de conception et soudures.....	25
6.3.1. CAO.....	25
6.3.2. Soudures.....	26
6.3.3. Illustration des tests de contrôle du robot.....	26
6.4. Propositions de sujets de projets.....	26

NOTATIONS, ACRONYMES

CAO : Conception Assistée par Ordinateur

URL : Uniform Resource Locator

IDE : Integrated Development Environment

APK : Android Package Kit

PWM : Pulse Width Modulation

1. INTRODUCTION

Nous tenons ici à tout d'abord souligner le fait que tous les membres du groupe n'ont pas choisi les mêmes filières de pré-spécialisation : certains ayant opté pour Physique/Informatique d'autres pour Mathématiques/Informatique ou bien encore Mécanique ; il fallut donc prendre en compte les capacités et connaissance de chacun afin de se répartir correctement le travail. Toutefois, ces différences nous ont permis de mettre en pratique et de développer les connaissances de chacun dans les domaines utiles à la réalisation de ce projet.

En effet, en tant que futur.e ingénieur.e, ce projet est un merveilleux moyen pour nous d'apprendre à travailler en équipe, en répartissant nos tâches et tirant avantage des compétences de chacun. Ce projet de construction d'un robot a la particularité de réunir plusieurs domaines de compétence, à savoir de l'informatique, de la mécanique et de l'électronique.

Les objectifs de ce projet peuvent être séparés en fonction de leur domaine de rattachement. En premier lieu, il était nécessaire de concevoir les pièces du robot sur *SolidWorks* puis de les réaliser en bois par découpe laser. Secondement, nous devons créer une carte électronique et y souder nos composants électroniques. Dernièrement, il nous était indispensable de créer une application mobile permettant de contrôler le robot, mais aussi donnant accès au flux vidéo venant d'un téléphone placé dans le robot.

Enfin, ce projet nous a permis d'étendre notre éventail de compétences par la recherche de solutions aux divers problèmes et interrogations qui ont pu être soulevés par ce projet.

2. MÉTHODOLOGIE / ORGANISATION DU TRAVAIL

Il a été décidé lors de la première séance que le groupe fonctionnerait de façon horizontale, sans chef d'équipe désigné, et ce afin de pouvoir impliquer au mieux chaque membre dans ce projet. Par ailleurs, un planning prévisionnel de l'avancée du projet a été créé à la suite de cette première séance. (cf. Annexes)

Les tâches à réaliser ont été réparties suivants leur domaine respectif : mécanique, électronique, et enfin informatique.

Les différentes sous-parties de ces domaines ont été réparties au sein du groupe comme suit :

- Design et CAO : Camille AUDOIN et Pierre LYNCH
- Électronique et montage : Camille AUDOIN, Victorien MERAND--MAUREL et Loic SEBA
- Informatique : Loic SEBA et Rayen ZOUAGHI

La partie *Design* était proche du fait de "donner la forme et les dimensions" du robot, Camille s'est occupé de cette partie et fût ensuite intégré au groupe *Électronique*. De son côté, Loic dû renforcer l'équipe en charge du domaine *Informatique* après le départ de Mattéo au cours du projet.

Nous pourrions ici préciser que le changement de carte *ESP* - initialement *ESP8266* puis *ESP32* - nous a mis en retard à cause des changements d'attribution des broches de la carte, de ses fonctionnalités mais aussi de sa programmation. L'équipe *Informatique* a également eu du mal à pouvoir confirmer les possibilités de la carte *ESP* par manque de connaissances techniques entraînant un certain retard pour l'équipe *Électronique*.

3. TRAVAIL RÉALISÉ ET RÉSULTATS

3.1. PARTIE MÉCANIQUE

Le but de cette partie était de concevoir le robot en fonction des matériaux à notre disposition. La forme du robot fut choisie par Camille et Pierre, avec accord du reste du groupe, puis le design et les dimensions furent réalisés par Camille. Enfin Pierre réalisa la modélisation 3D ainsi que les impressions.

3.1.1. Design

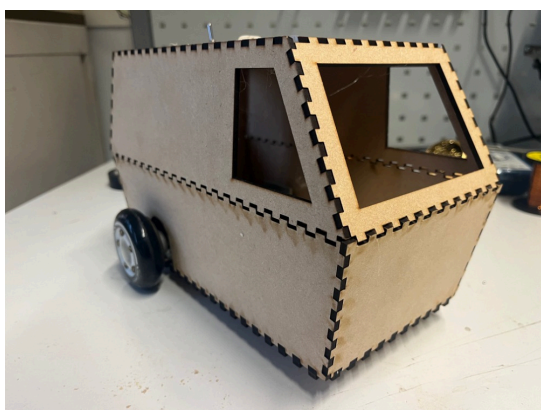
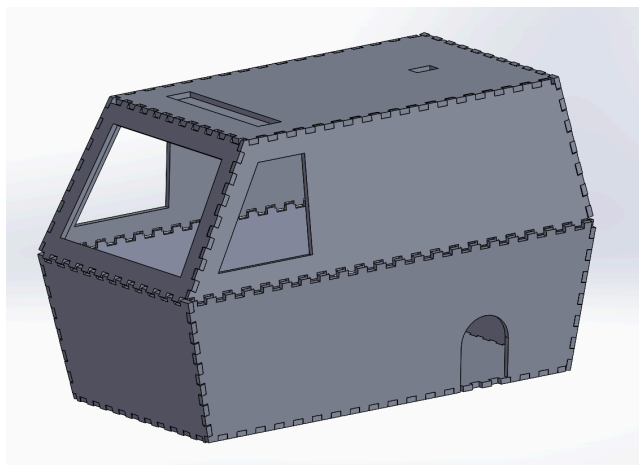
Pour la réalisation de la coque, nous nous sommes inspirés du van du dessin animé *Scooby-doo*. En effet, nous avons besoin de quelque chose de suffisamment grand pour contenir tous les composants, plutôt ressemblant à un véhicule puisque notre robot devait avancer avec des roues, et cela ajoutait notre touche personnelle à ce projet.

3.1.2. Modélisation 3D et fabrication

Après avoir choisi la forme de notre robot, nous avons commencé sa réalisation sur *Solidworks* dans le but de l'imprimer à l'aide de l'imprimante 3D de l'INSA.

Cependant, une fois notre coque terminée, c'est-à-dire la semaine du 9 avril environ, nous avons découvert que l'imprimante ne pouvait imprimer que des pièces de taille inférieure à 240 mm, alors que notre robot mesurait 260 mm de long. Cela nous empêchait de l'imprimer en une seule fois.

Nous avons la possibilité de l'imprimer en deux parties, mais nous craignons que cela ne donne un rendu peu esthétique et peu solide.



Nous avons donc décidé de fabriquer la coque du robot en utilisant des plaques de médium, découpées au laser. Pour cela, nous avons dû refaire notre conception en découpant le modèle en plaques distinctes, car *Solidworks* ne possède pas, à notre connaissance, d'outils permettant de convertir un modèle complet en un assemblage de plaques. Ensuite, nous avons assemblé les plaques en les collant. Ce processus nous a appris à adapter nos conceptions aux contraintes techniques et à trouver des solutions alternatives, enrichissant ainsi notre expérience en ingénierie et en fabrication.

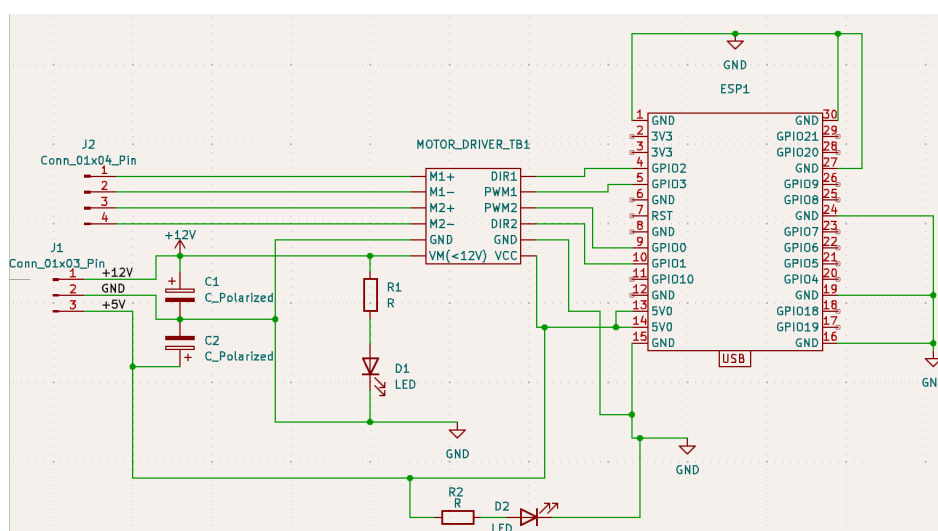
3.2. PARTIE ÉLECTRONIQUE

Les objectifs du groupe chargé de cette partie furent de définir les composants nécessaires à la réalisation du projet, vérifier leurs capacités techniques, réaliser un circuit imprimé et y souder tous les composants.

Victorien et Loic furent en charge du choix des composants et de la vérification de leurs capacités. Victorien réalisa en majeure partie le système d'émission et de réception des sons, avec l'aide de Loic ; Camille modélisa une carte électronique avec l'appui de Loic ; puis Victorien souda les composants sur la carte et tous trois assemblèrent le robot avec Pierre.

3.2.1. Description de la carte électronique

Comme nous avons plusieurs composants à placer dans notre circuit électrique, notre enseignant nous a conseillé de réaliser une carte électronique afin d'assurer la stabilité de ceux-ci. Le schéma de notre carte électronique est le suivant :



L'entrée 12V est directement connectée à notre batterie au lithium - en courant continu donc - l'entrée 5V est également liée à la batterie mais par l'intermédiaire d'un module abaisseur de tension (non présent sur le schéma), pour ne pas endommager la carte *ESP*. Afin de contrôler les moteurs, nous utilisons un *Dual Motor Driver* (double contrôleur de moteur) qui est connecté d'une part aux moteurs et à la batterie et d'autre part à l'*ESP*.

Des composants auxquels nous n'avions pas pensé, comme des condensateurs, des résistances et des diodes, nous ont été conseillés par notre enseignant pour garantir l'absence de sursaut de tension et un courant électrique stable.

Les broches de l'*ESP* ont été choisies par les équipes électronique et informatique pour leurs fonctionnalités et leur facilité de mise en œuvre pour les soudures.

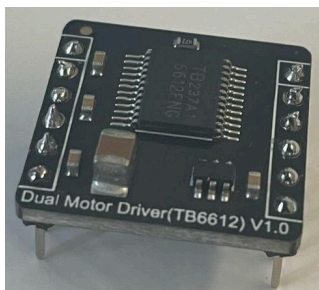
3.2.2. Émission et réception de sons

La partie liée à l'émission et la réception de sons est un système électronique récupéré sur un casque bluetooth qui n'était plus en état de marche. Il s'agit d'un apport personnel de notre part en composants électroniques (*cf. Annexes*).

Pour l'intégrer à notre robot, il fallut le remettre en état de fonctionner, c'est-à-dire refaire les soudures de la carte bluetooth aux enceintes ainsi que recréer une liaison électrique entre deux sous-parties du système. Sa réparation demanda du temps en raison de la difficulté à ressouder les fils électriques très fins utilisés à l'origine - fils que nous n'avions pas souhaité changer pour limiter l'utilisation de ressources de l'école pour des composants nous appartenant.

3.2.3. Soudures

Les soudures de tous les composants furent particulièrement délicates à réaliser puisqu'il s'agissait pour nous d'une découverte, à petite échelle, et que nous n'avions en conséquences que peu d'expériences. Par ailleurs, le circuit imprimé obtenu était réalisé à l'envers, compliquant davantage les soudures sur le circuit. Néanmoins, cela s'est globalement bien passé et l'accès à un atelier électrique - disposant de tout le matériel nécessaire à cette pratique - a grandement contribué à notre réussite. Nous souhaitons par ailleurs remercier notre enseignant ainsi que les techniciens qui nous ont prodigué des conseils éclairés pour améliorer nos soudures.



Exemples de soudures avec le module de contrôle des moteurs et ses broches soudées.

3.2.4. Problèmes rencontrés

Le changement de carte *ESP* a conduit à d'importantes modifications du circuit imprimé à réaliser. Cela a ralenti son développement et sa création.

De même, initialement, une caméra fournie par l'enseignant, la *Pixy2*, devait être utilisée mais des problèmes de réception de l'image liés à la carte *ESP* fournie ont été rencontrés par l'équipe en charge de la partie *Informatique*. Nous avons donc dû nous résoudre à utiliser la caméra d'un téléphone portable, rendant notre robot moins esthétique et demandant encore des modifications de la carte électronique à développer. Ce

changement entraîna un retard non négligeable puisqu'il avait fallu un peu plus d'une séance et demie pour comprendre le fonctionnement de la caméra et l'assembler.

Les soudures devant être précises au regard de l'étroitesse des espaces entre les chemins du circuit imprimé, nous devons être particulièrement vigilants et concentrés pour réaliser les liaisons électriques. Certaines soudures nous ont ainsi demandé plusieurs essais et retouches avant d'être finalisées.

Comme évoqué, l'inversion de l'impression des circuits sur la carte a complexifié la mise en œuvre des soudures, mais cette difficulté supplémentaire fut surmontable.

3.3. PARTIE INFORMATIQUE

3.3.1. Fonctionnement du robot

Le robot fonctionne grâce à un *ESP32-C3-DevKitM* et un téléphone portable. Comme dit précédemment, le choix du téléphone a été fait car le système original fonctionnant avec une caméra n'était pas compatible avec notre modèle d'*ESP*. Le téléphone portable nous sert de caméra.

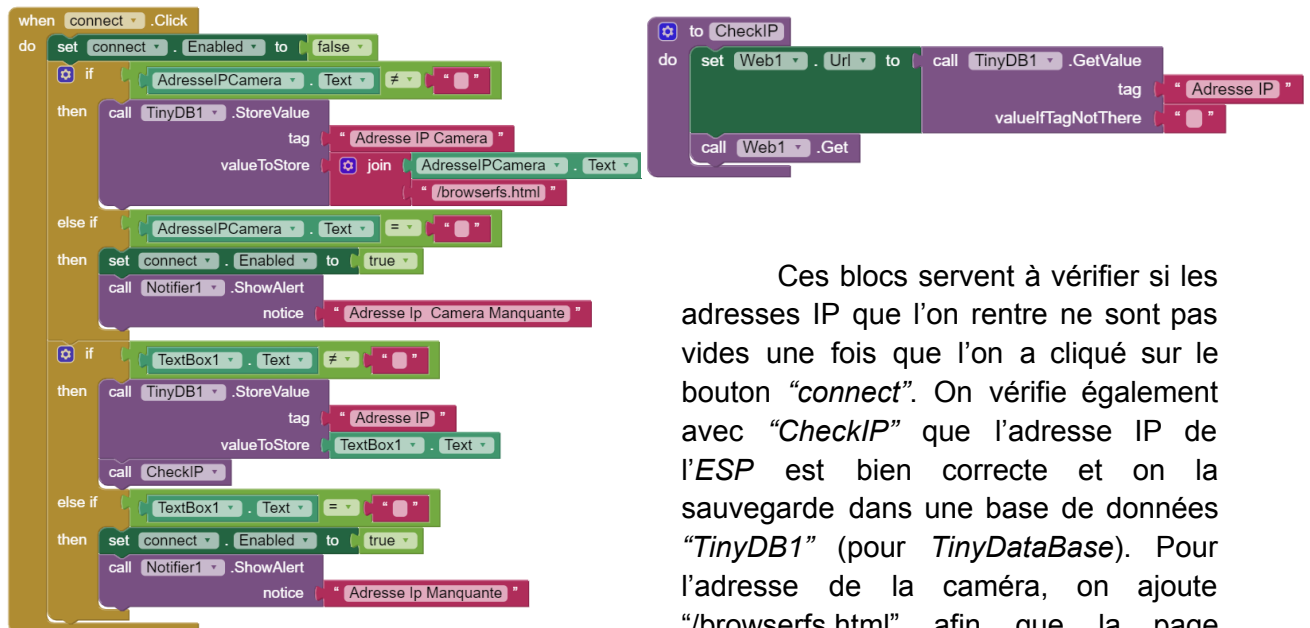
Grâce à une application se nommant *IP Webcam*, le téléphone crée un serveur sur lequel on peut se connecter pour accéder à la caméra. L'*ESP* quant à lui sert à contrôler les moteurs : il indique à la carte moteur à quelle vitesse et dans quelle direction nous allons.

Pour se faire, on utilise une application se nommant *MIT App Inventor* sur un téléphone *Android* ou un ordinateur compatible. Cette dernière sert à indiquer quelles procédures appliquer lorsque tel bouton est appuyé. Les procédures, quant-à-elles, sont codées en langage *Arduino* (variante de *C++*) et indiquent quelle broche activer et à combien les PWM doivent être fixés pour chaque procédure (Tourner à gauche, à droite..).

3.3.2. Programmation MIT App Inventor

3.3.2.1. Ecran 1 :

Le premier écran sert à se connecter au robot. On y voit deux champs de texte : un pour rentrer l'adresse IP de la caméra et un autre pour rentrer l'adresse IP de l'*ESP*.



Ces blocs servent à vérifier si les adresses IP que l'on rentre ne sont pas vides une fois que l'on a cliqué sur le bouton "connect". On vérifie également avec "CheckIP" que l'adresse IP de l'*ESP* est bien correcte et on la sauvegarde dans une base de données "TinyDB1" (pour *TinyDataBase*). Pour l'adresse de la caméra, on ajoute "/browserfs.html" afin que la page s'ouvre en plein écran.

```

when Web1 .GotText
  url responseCode responseType responseContent
  do
    if
      get responseCode = 200
    then
      call Notifier1 .ShowAlert
        notice "Connexion en cours"
      call TextBox1 .HideKeyboard
      open another screen screenName Screen2
    else if
      get responseCode = 404
    then
      call Notifier1 .ShowAlert
        notice "La page que vous cherchez est introuvable."
      set connect .Enabled to true
    else if
      get responseCode = 403
    then
      call Notifier1 .ShowAlert
        notice "Accès refusé. Vérifiez l'adresse IP"
      set connect .Enabled to true
    else
      call Notifier1 .ShowAlert
        notice "Problème de connexion à l'adresse IP"
      set connect .Enabled to true

when Screen1 .ErrorOccurred
  component functionName errorNumber message
  do
    if
      get errorNumber = 1101
    then
      call Notifier1 .ShowAlert
        notice "Vérifiez que votre adresse IP est correcte."
      set connect .Enabled to true
    else if
      get errorNumber = 1109
    then
      call Notifier1 .ShowAlert
        notice "L'adresse IP n'est pas entrée."
      set connect .Enabled to true
    else
      call Notifier1 .ShowAlert
        notice "Problème de connexion à l'adresse IP"
      set connect .Enabled to true
  
```

Lorsqu'une adresse IP est rentrée, on interprète le code "responseCode" qui nous est renvoyé par la fonction "call Web1.get". Ainsi, chaque code correspond à une certaine erreur (ou réussite) lorsque l'on veut se connecter à une adresse IP (problème de protocole *https* par exemple). De plus, le bloc de droite gère les cas où l'adresse IP rentrée n'est pas valide (problème de faute de syntaxe par exemple). S'il n'y a pas d'erreur, alors l'écran 2 s'affiche.

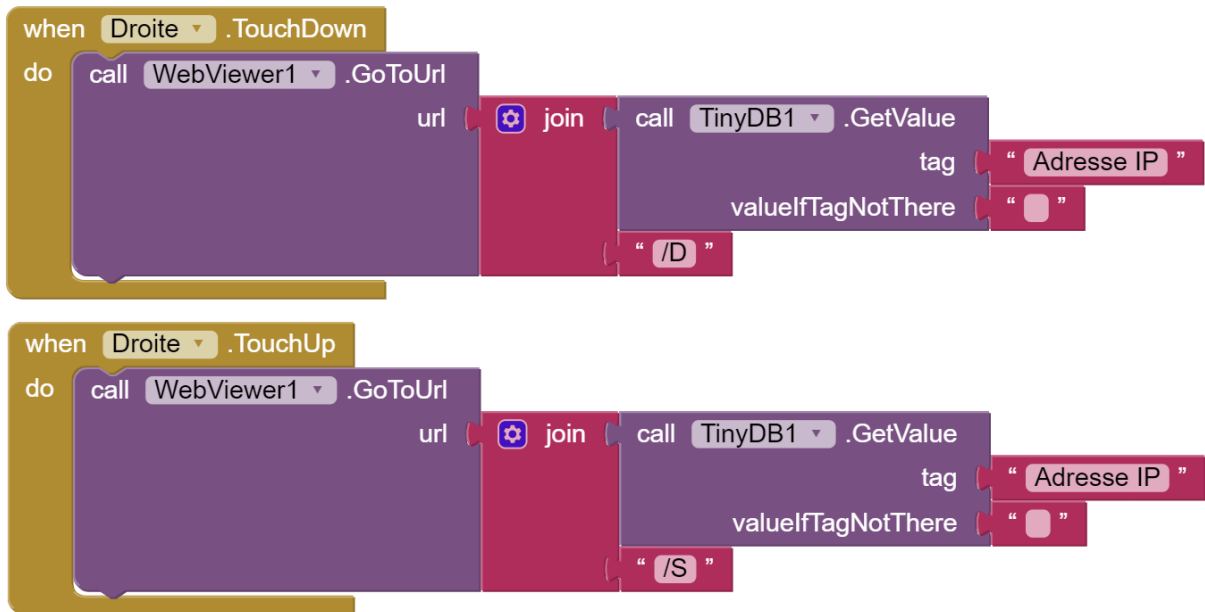
3.3.2.2. Ecran 2 :

Le deuxième écran est divisé en deux parties. Sur la partie du haut, il y a la caméra du robot et, sur la partie basse, il y a les commandes pour pouvoir commander le robot.

```

when Screen2 .Initialize
  do
    call WebView2 .GoToUrl
      url call TinyDB1 .GetValue
        tag "Adresse IP Camera"
        valueIfTagNotThere " "
  
```

Ce bloc sert à ce que le navigateur intégré (*WebView2*) se connecte à la page créée par le téléphone grâce à l'adresse IP sauvegardée précédemment.

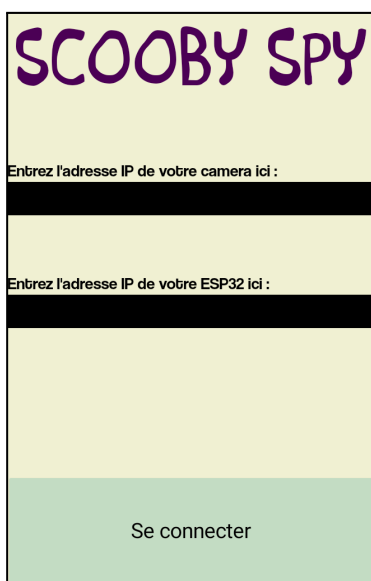


Pour contrôler le robot, deux blocs sont associés pour chaque direction. Le fonctionnement est simple, lorsque l'on appuie sur le bouton, on envoie à l'ESP, grâce à *WebView1*, une instruction codée par une lettre (G pour Gauche, D pour Droite...) et lorsque l'on relâche le bouton, un signal est envoyée pour qu'il s'arrête (S pour Stop). Ainsi, chaque instruction donnée par le code *MIT App Inventor* correspond à une fonction Arduino qui va régler les bonnes broches afin d'ajuster les différents PWM et les directions (Avancer/Reculer), à l'aide de la carte moteur.

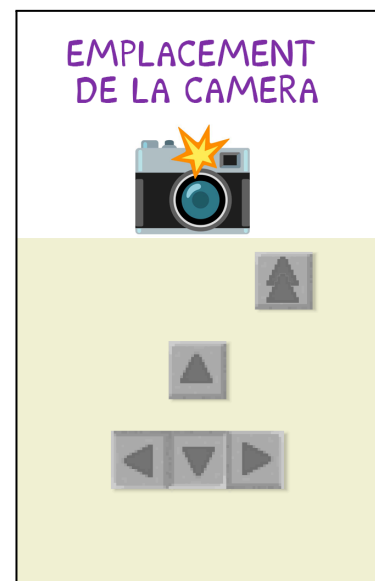
3.3.2.3. Aperçu de l'écran de contrôle :

Voici un aperçu de l'application utilisée pour contrôler le robot :

Écran 1 :



Écran 2 :



3.3.3. Code Arduino

La carte *ESP32-C3-DevKitM-1* fonctionne de la même manière qu'un Arduino mais avec la possibilité de se connecter à un réseau *WiFi* (ou bien d'en créer un).

Ainsi, pour programmer notre carte, nous avons utilisé l'IDE Arduino 2.3.2. C'est le même IDE utilisé afin de programmer sur des cartes arduino. Il fallait juste configurer correctement l'IDE utilisé afin qu'il fonctionne avec l'*ESP* : il faut appeler les bonnes bibliothèques conçues pour la manipulation de l'*ESP*.

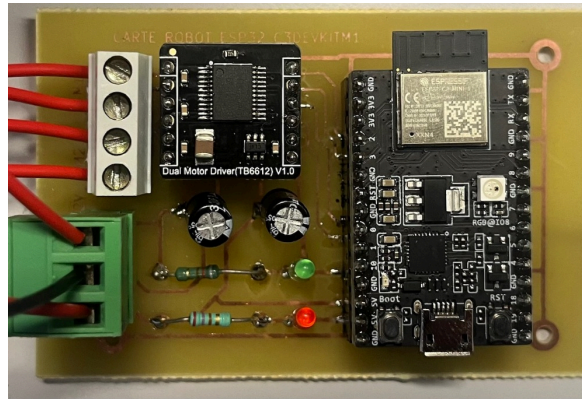
Puisque nous n'avons quasiment aucune connaissance concernant le C++ (langage de programmation de l'IDE), il était nécessaire de chercher des exemples de code sur des forums Internet pour nous en inspirer. Petit à petit, nous avons commencé à appréhender les bases de C++.

Le robot est contrôlé via des requêtes envoyées sur le serveur créé par l'*ESP*. Une requête est envoyée par l'application à l'*ESP* qui l'interprète pour faire tourner les moteurs.

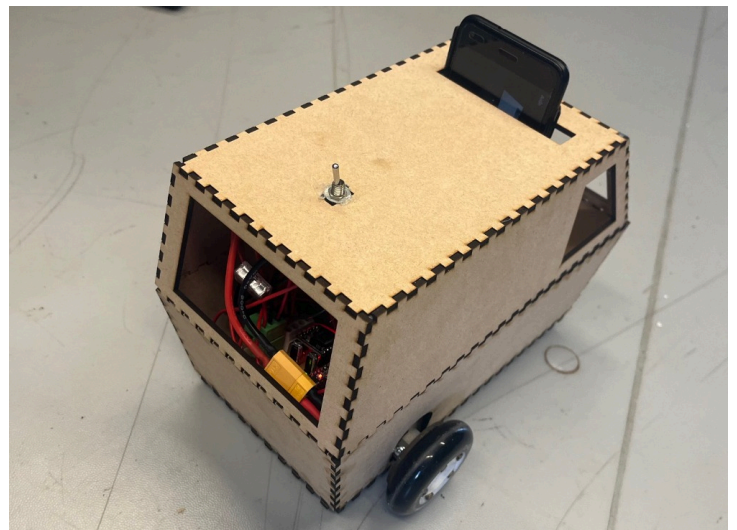
De par la longueur du code Arduino, celui-ci est donné en Annexes.

3.4. RÉSULTATS

Les résultats de notre travail sont concluants et nous sommes fiers de pouvoir donner un aperçu de notre robot.



Circuit imprimé avec nos composants soudés



Vues extérieures de notre robot

4. CONCLUSION ET PERSPECTIVES

4.1. Conclusions sur le travail réalisé

Nous avons pu atteindre tous nos objectifs pour ce projet, dont le plus important était la réalisation du robot. Bien que nous soyons déçus de ne pas avoir pu utiliser une caméra individuelle au lieu d'un smartphone, nous sommes tout de même fiers du travail accompli. Tous les membres du groupe se sont impliqués et cela nous a permis de faire face aux divers problèmes que nous avons rencontrés, tant en électronique et en informatique qu'en ce qui concerne la gestion du temps.

4.2. Conclusions sur l'apport personnel de cet EC projet

Camille AUDOIN : Ce projet a été très enrichissant, que ce soit d'un point de vue personnel ou technique. En effet, je n'avais jamais participé à un projet d'une telle envergure, et cela m'a permis d'apprendre beaucoup. Tout d'abord, j'ai compris que bien se répartir le travail était primordial, et qu'il fallait faire confiance à chacun dans l'avancement de leurs tâches. De plus, ce projet m'a apporté de nouvelles connaissances dans des domaines que je ne connaissais pas, comme la mécanique et l'électronique, ce qui m'a permis de réfléchir à d'éventuelles possibilités d'orientation.

Pierre LYNCH : La conception d'un robot pour ce projet m'a apporté une expérience notable tant sur le plan technique que personnel. Sur le plan technique, j'ai pu appliquer concrètement mes connaissances en mécanique tout en acquérant de nouvelles compétences en résolution de problèmes. Par ailleurs, ce projet m'a permis de développer mes capacités de travail en équipe, de gestion de projet et de communication, en collaborant avec mes camarades pour surmonter les problèmes rencontrés.

Victorien MERAND--MAUREL : Ce projet fût particulièrement formateur concernant le domaine de l'électronique et des interactions entre divers composants que je n'avais jusqu'à présent peu l'habitude d'utiliser. De plus, notre organisation horizontale m'a montré qu'il ne suffisait pas de dates limites pour qu'un projet avance mais qu'il fallait aussi la volonté de chaque membre du groupe (moi y compris) pour mener à bien un projet, cet enseignement m'est particulièrement important pour mes futurs travaux de groupe, tant scolaires que professionnels.

Loïc SEBA : Participer à ce projet m'a permis de développer plusieurs compétences que je n'avais pas acquises avant, notamment en ce qui concerne la partie électronique. De plus, ce projet m'a enseigné l'importance de la gestion du temps et de l'organisation pour respecter les délais et les objectifs du projet. Mais le plus grand avantage de ce projet, pour

moi, est le fait que ce soit un projet avec quelque chose de concret à rendre à la fin. En effet, la satisfaction de voir notre robot fonctionner grâce à nos efforts a été extrêmement gratifiante.

Rayen ZOUAGHI : Ce projet m'a offert une expérience enrichissante tant sur le plan technique que personnel. En me chargeant de la partie informatique, j'ai pu approfondir mes connaissances en programmation. J'ai aussi acquis de nouvelles compétences en résolution de problèmes et en intégration de systèmes ce qui est l'essence du métier d'ingénieur. Par ailleurs, ce projet m'a également permis de développer des compétences en gestion de projet et en gestion de temps. Finalement, voir nos efforts aboutir à un résultat concret a été particulièrement récompensant.

4.3. Perspectives pour la poursuite de ce projet

Il pourrait être pratique d'utiliser une caméra individuelle à la place d'un téléphone portable pour pouvoir réduire la taille du robot, par exemple avec une *ESP Cam*.

De même, la création d'une version *IOS* de l'application serait un apport non négligeable pour la portabilité du projet sur les Iphone, Ipad et Mac. Toutefois, cela requiererait plus de travail car cela demanderait de coder une application en Java ou de trouver d'autres alternatives plus complexes. En effet, l'application que l'on utilise, *MIT App Inventor*, n'exporte que des *.apk* - qui est un format de fichiers pour le système d'exploitation *Android* uniquement. Toutefois, la politique commerciale d'Apple ne permet pas de porter cette application sur Iphone pour le moment, ni de trouver un équivalent sur ce type de téléphone.

5. BIBLIOGRAPHIE

[1] Introduction to ESP32 - Getting Started

https://www.youtube.com/watch?v=xPIN_Tk3VLQ&list=PLxoU7Q4jEfiTuv4YYTBUDlablKgr4RY4Y&index=7&t=15s&ab_channel=DroneBotWorkshop

(valide à la date du 26/05/2024).

[2] How To Build a WiFi Based Robot with Android Application Control | Android App with MIT App Inventor

https://www.youtube.com/watch?v=S-67_HUxrOg&ab_channel=MohammadAnsar

(valide à la date du 26/05/2024).

[3] B-06 NODEMCU (ESP8266) en Français: Recevoir des données de l'ESP8266 avec une application Android

https://www.youtube.com/watch?v=8plw9ETZMIA&ab_channel=ADELKASSAH

(valide à la date du 26/05/2024).

[4] Драйвер моторов. DFrobot.com

https://www.youtube.com/watch?v=50J3Odiwkj8&list=PLxoU7Q4jEfiTuv4YYTBUDlablKgr4RY4Y&index=20&ab_channel=HomeMade

(valide à la date du 26/05/2024).

[5] ESP32 Cam Video Streaming App Inventor 2

https://www.youtube.com/watch?v=zpEhZ41608g&list=PLxoU7Q4jEfiTuv4YYTBUDlablKgr4RY4Y&index=20&ab_channel=appyProjekt

(valide à la date du 26/05/2024).

[6] Pixy2 documentation

https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:pixy_regular_quick_start

(valide à la date du 26/05/2024).

6. ANNEXES

6.1. Documents

6.1.1. Planning prévisionnel de l'avancé du projet

Planing (date ~ au plus tard, des délais supplémentaires sont anticipés, se décalant sur les mois de mai et juin)		Agenda INSA
13/02	Choix du robot, définitions des spécifications	
20/02	Premières esquisses du robot, tests de l'électronique, début de programmation	
		24/02 -> 10/03 : Vacances
19/03	CAO et électronique théorique terminée, code avancé permettant un premier test de l'interface et des connexions électroniques.	
		02/04 : IS
09/04	Début de rédaction du rapport, début d'usinage des pièces et mise en oeuvre de l'électronique	
16/04	Code terminé	
		20/04 -> 12/05 : Vacances
14/05	Premiers tests complets	
04/06	Correction de tous les problèmes, fin de rédaction du rapport, début de préparation de l'oral	
13/06	Rendu du rapport	
		17/06 : DS
21/06	Rendu poster et support présentation orale	
24/06	Oral du projet	

Les différents retards accumulés n'ont pas permis de respecter l'intégralité du planning, cependant la date finale de rendu a été respectée.

6.1.2. Résumé des séances

6 Février	Découverte du projet, prise de connaissances des attendus et réflexion sur les fonctionnalités possibles du robot.
13 Février	Choix du robot et spécifications des attendus, recherches des composants utiles (électronique et mécanique) et apprentissage de la programmation d' <i>ESP</i> .
20 Février	Premières esquisses (physiques et numériques) du robot, tests du matériel électronique et début de programmation de l' <i>ESP</i> (partie <i>arduino</i>).

12 Mars	CAO en cours, montage de la caméra Pixy2 et difficultés de codage de l'ESP (partie <i>arduino</i>).
19 Mars	CAO en cours, difficultés à faire fonctionner la caméra Pixy2, programmation de l'ESP en cours et début de codage de l'application.
26 Mars	Fin de la CAO, réussite du fonctionnement de la caméra Pixy2 en la connectant à un ordinateur avec le logiciel PixyMon, difficultés de codage de l'ESP (partie <i>arduino</i>) et application en cours.
9 Avril	Information de l'enseignant des dimensions maximales de l'imprimante 3D, difficultés de contrôle de la caméra Pixy2 via l'ESP 8266, changement de logiciel de création de l'application pour des raisons techniques.
16 Avril	Adaptation de la CAO pour l'utilisation d'une découpeuse laser, changement de carte de contrôle des moteurs et d'ESP sur avis de l'enseignant, début de modélisation du circuit imprimé, changement du code de l'ESP (partie <i>arduino</i>) et création de l'application en cours.
14 Mai	CAO en cours, découverte de l'impossibilité de contrôler la caméra Pixy2 avec notre modèle d'ESP et changement du codage de l'application.
21 Mai	Fin de la CAO, envoi des fichiers utiles à la création des pièces à l'enseignant, fin de modélisation du circuit imprimé, fin des parties informatique et électronique théorique.
28 Mai	Impossibilité de montage des pièces par leur absence, dernier test de contrôle des parties informatique et électronique réussi et début des soudures finales.
4 Juin	Obtention de toutes les pièces mécaniques, montage provisoire du robot, obtention de tous les composants électroniques (hors circuit imprimé en cours de fabrication), test de fonctionnement des composants, test du robot (sur un circuit électronique provisoire), début de création du poster.
11 Juin	Réception du circuit imprimé, soudures de tous les composants, montage finale du robot, test de fonctionnement du robot Fin du projet, résultat : réussite du projet.

6.2. Code Arduino

```
#include <WiFi.h> //Inclue la bibliothèque WIFI.H

const char* ssid = "WIFI-WPA"; // Nom du hotspot créé par le
téléphone/ou bien du wifi utilisé pour se connecter à internet

const char* password = "password"; // Mot de passe du réseau

WiFiServer server(80); // Port réseau utilisé pour la configuration
d'un serveur HTTP

#define AIN 2 // GPIO2
#define BIN 1 // GPIO1
#define PWM_A 0 // GPIO0
#define PWM_B 3 // GPIO3

void setup() {

    Serial.begin(115200); // Vitesse de communication entre la carte et
le moniteur série (sur le PC)

    delay(10);

    // Setup PWM channels
    ledcSetup(0, 20000, 8); // 20 kHz PWM, 8-bit resolution (PWM_A)
    ledcSetup(1, 20000, 8); // 20 kHz PWM, 8-bit resolution (PWM_B)
    ledcAttachPin(PWM_A, 0);
    ledcAttachPin(PWM_B, 1);
    pinMode(AIN, OUTPUT);
    pinMode(BIN, OUTPUT);
    toutArreter(); // Vérifie si les moteurs sont au repos initialement
    Serial.println();
    Serial.println();
    Serial.print("Connexion borne wifi : ");
    Serial.println(ssid);

    WiFi.mode(WIFI_STA); // Assurez-vous que l'ESP32 est en mode
station
    WiFi.begin(ssid, password);
    unsigned long startAttemptTime = millis();
    // Réessayer jusqu'à ce que la connexion soit établie ou timeout
(10 secondes)
    while (WiFi.status() != WL_CONNECTED && millis() - startAttemptTime
< 10000) {
        delay(500);
        Serial.print(".");
    }
    if (WiFi.status() == WL_CONNECTED) {
```

```
Serial.println("");
Serial.println("Connexion WiFi etablie");
Serial.print("Adresse IP de l'ESP32 : ");
Serial.println(WiFi.localIP());
server.begin();
Serial.println("Serveur en marche");
} else {
    Serial.println("");
    Serial.println("Échec de la connexion WiFi");
    // Redémarrer l'ESP32 après un échec de connexion
    Serial.println("Redémarrage de l'ESP32...");
    ESP.restart();
}
}
void loop() {
    WiFiClient client = server.available();
    if (!client) {
        return;
    }
    Serial.println("Nouveau Client");
    while (!client.available()) {
        delay(1);
    }
    String request = client.readStringUntil('\r');
    Serial.println(request);
    client.flush();
    if (request.indexOf("/S") > 0) {
        toutArreter();
        Serial.println("arreter");
    } else if (request.indexOf("/Av+") > 0) {
        avancerRapidement();
        Serial.println('Avancer rapidement');
    } else if (request.indexOf("/Av-") > 0) {
        avancerLentement();
        Serial.println('Avancer');
    } else if (request.indexOf("/D") > 0) {
        tournerDroite();
        Serial.println('Droite');
```

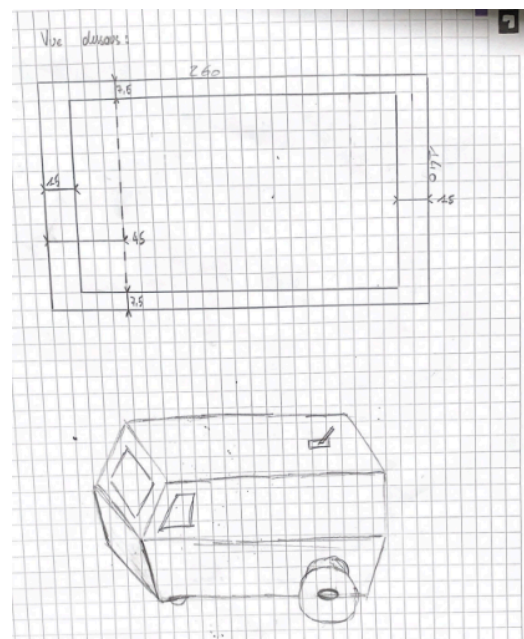
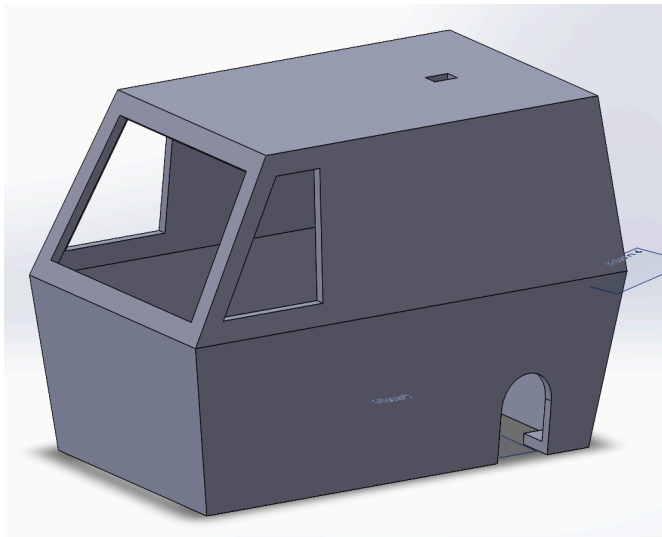
```
    } else if (request.indexOf("/G") > 0) {
        tournerGauche();
        Serial.println('Gauche');
    } else if (request.indexOf("/R") > 0) {
        marcheArriere();
    }
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("");
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    client.println("<head>");
        client.println("<meta    name='apple-mobile-web-app-capable'
content='yes' />");
        client.println("<meta name='apple-mobile-web-app-status-bar-style'
content='black-translucent' />");
        client.println("<meta name='viewport' content='width=device-width,
minimum-scale=0.1' />");
    client.println("</head>");
    client.println("<body bgcolor='#0000FF' style='margin: 0;'>");
    client.println("</body>");
    client.println("</html>");
}
void toutArreter() {
    digitalWrite(AIN, LOW);
    digitalWrite(BIN, LOW);
    ledcWrite(0, 0);
    ledcWrite(1, 0);
}
void avancerRapidement() {
    digitalWrite(AIN, LOW);
    digitalWrite(BIN, HIGH);
    ledcWrite(0, 255);
    ledcWrite(1, 255);
}
void avancerLentement() {
    digitalWrite(AIN, LOW);
    digitalWrite(BIN, HIGH);
```



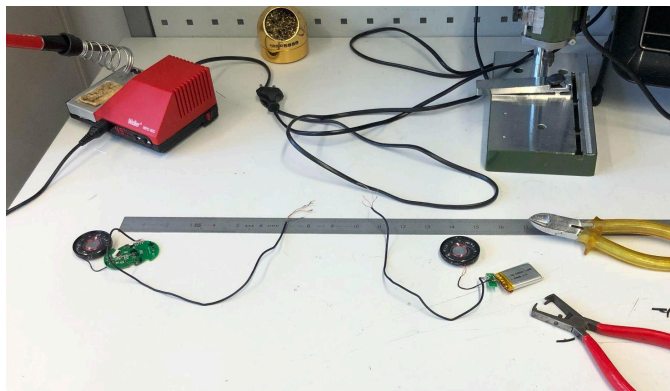
```
    ledcWrite(0, 150); // Ajuste la vitesse de la roue (255 est la
vitesse maximale qu'on peut avoir)
    ledcWrite(1, 150); // Ajuste la vitesse de la roue (255 est la
vitesse maximale qu'on peut avoir)
}
void tournerGauche() {
    digitalWrite(AIN, LOW);
    digitalWrite(BIN, HIGH);
    ledcWrite(0, 50); // Ajusté pour tourner
    ledcWrite(1, 255); // Ajusté pour tourner
}
void tournerDroite() {
    digitalWrite(AIN, HIGH);
    digitalWrite(BIN, LOW);
    ledcWrite(0, 255); // Ajusté pour tourner
    ledcWrite(1, 50); // Ajusté pour tourner
}
void marcheArriere() {
    digitalWrite(AIN, HIGH);
    digitalWrite(BIN, LOW);
    ledcWrite(0, 255);
    ledcWrite(1, 255);
}
```

6.3. Schémas de montages, plans de conception et soudures

6.3.1. CAO

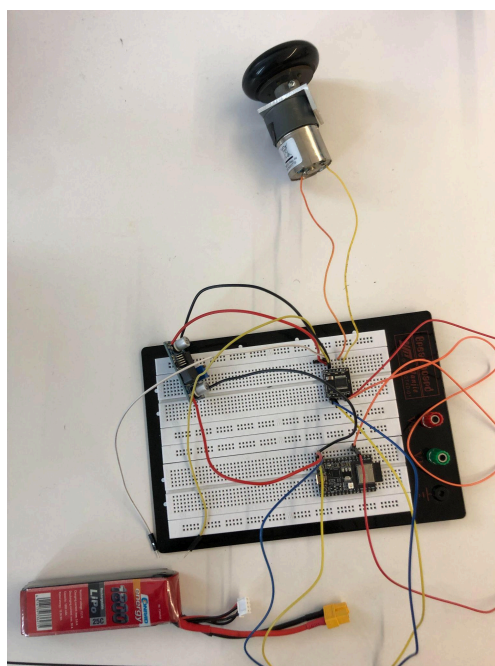


6.3.2. Soudures



Cette illustration montre le dispositif de réception et d'émission de sons lors de sa remise en état.

6.3.3. Illustration des tests de contrôle du robot



Au cours de notre projet, nous avons réalisé plusieurs essais des composants afin de nous assurer à chaque instant de leur fonctionnement et de leur bonne interaction entre eux. Cette illustration correspond au dernier test réalisé avant l'utilisation du circuit imprimé.

6.4. Propositions de sujets de projets

- Étude de bactéries bioluminescentes : quelles perspectives d'usages à l'avenir ?
- Comparaison du rendement énergétique d'un train/tramway et d'une voiture thermique/électrique/hydrogène