**Une image contenant texte, Police, Graphique, logo

Le contenu généré par l’IA peut être incorrect.Rapport de projet**

**L1 – CMI**

**Session 2025**

Robot support LiDAR InfraRouge (IR)

**Une image contenant texte, Police, Graphique, logo

Le contenu généré par l’IA peut être incorrect.**

**Centre d’Enseignement et de Recherche en Informatique**

**Campus Jean-Henri Fabre**

|  |  |
| --- | --- |
| **Auteurs :**  VILLARD Logan  GOMEZ-BARTHEYE Esteban  RANDRIAMANANTENA Samy | **Encadrants :**  M. GOZLAN  M. SILANUS |

**339 Chemin des Meinajaries**

**84911 Avignon Cedex 9**

Table des matières

[**1. Introduction** 3](#_Toc197385406)

[**1.1 Contexte** 3](#_Toc197385407)

[**1.2 Problématique** 3](#_Toc197385408)

[**2. État de l’art** 4](#_Toc197385409)

[**2.1 Solution 1 : robots aspirateurs** 4](#_Toc197385410)

[**2.2 Solution 2 : robots tondeurs** 5](#_Toc197385411)

[**3. Solution développée** 5](#_Toc197385412)

[**3.1 Partie matérielle** 5](#_Toc197385413)

[**3.2 Partie logiciel** 8](#_Toc197385414)

[**4 Organisation** 9](#_Toc197385415)

[**4.1 L’équipe de projet** 9](#_Toc197385416)

[**4.2 Répartition des tâches** 9](#_Toc197385417)

[**4.3 Environnement matériel/logiciel** 10](#_Toc197385418)

[**4.4 Planification** 10](#_Toc197385419)

# **1. Introduction**

## **1.1 Contexte**

Dans le cadre de l’Unité d’Enseignement (UE) « CMI 3 : Interfaces du réel au numérique », nous avons eu l’occasion de développer certaines compétences que nous devons avoir à la fin de notre Cursus Master Ingénierie informatique, et ce grâce au projet que nous avons choisi : la réalisation d’un robot, sur lequel viendra s’ajouter le radar LiDAR, lui-même produit par un second groupe.

## **1.2 Problématique**

Lors de ce projet, le robot que nous avons créé doit répondre à plusieurs critères.  
Premièrement, le robot doit pouvoir se déplacer au sol, de manière manuelle ou en étant autonome :

* Déplacements manuels : l’utilisateur devra être capable de faire se déplacer le robot au sol en lui donnant les directions où se diriger. Le robot exécutera alors l’ordre de l’utilisateur et prendra la direction souhaitée.
* Déplacements autonomes : le robot devra être capable de se déplacer dans l’espace dans lequel il se trouve, sans recevoir d’ordre, que ce soit par l’utilisateur ou par un logiciel qui aurait enregistré un parcours au préalable. Il devra donc s’adapter à chaque situation.

Afin de réaliser ses déplacements en étant autonome, plusieurs critères vont se rajouter à la réalisation du robot :

* Afin de pouvoir se déplacer en toute liberté dans son espace, le robot ne doit pas être relié à un appareil externe fixe, tel qu’un ordinateur ou une prise. Autrement, le robot serait contraint à rester dans une zone limitée.
* L’espace dans lequel va se trouver le robot ne sera pas infiniment grand, ni vide : il comportera des murs, des obstacles comme des chaises, des bureaux, ou même des éléments mobiles comme des personnes ou encore d’autres robots ! Il devra donc les détecter afin de les éviter, et d’éviter toute collisions.
* L’utilisateur ne va pas laisser le robot se déplacer indéfiniment. Ce dernier doit donc comporter des solutions pour s’arrêter quand l’utilisateur le décide.

Enfin, le projet exige une application de gestion des déplacements du robot, sur téléphone, pour pouvoir répondre à certaines contraintes citées ci-dessus.

En dehors de ces critères/contraintes, la réalisation du robot est totalement libre.

# **2. État de l’art**

La robotique autonome connait un grand succès, que ce soit dans la robotique autonome domestique ou industrielle. D’un côté, ils permettent d’exécuter des tâches ménagères ou d’entretien sans intervention humaine constante, de l’autre ils permettent d’accomplir des tâches de précision, de transport de marchandises dans des entrepôts, et bien d’autres encore.

C’est pourquoi ce domaine attire de nombreuses et grandes entreprises qui innovent de plus en plus pour satisfaire la demande grandissante des particuliers et des responsables d’entreprises.

|  |  |
| --- | --- |
| Domestique | Secteur industriel |
| Robots aspirateurs  Robots tondeurs  Robots laveurs de vitres  Robots de surveillance | Logistique  Automobile (assemblage, soudure)  Aéronautique (assemblage, soudure)  Agroalimentaire (emballage, tri automatisé) |

Voici un tableau représentant quelques domaines d’utilisation, au niveau domestique ou industriel :

Les technologies créées pour les robots autonomes sont nombreuses.

Parmi les plus récentes et remarquables :

* Le développement de cartographies 3D et de cartes dynamique rend la navigation autonome plus efficace et sécurisée.
* Retour à une station de recharge en étant autonome, et ce même après avoir été déplacés ou rencontrés des obstacles.
* Utilisation combinée de LiDAR, caméras, ultrasons et capteurs infrarouges pour améliorer la détection d’obstacles, même dans des environnements mal éclairés.

Comparé à notre projet, certaines utilisations se rapprochent grandement de ce qu’on souhaite réaliser : les robots aspirateurs ou encore les robots tondeurs.

## **2.1 Solution 1 : robots aspirateurs**

Les robots aspirateurs sont utilisés pour nettoyer des appartements, des maisons ou des bureaux. Mais comme pour notre projet, une des contraintes les plus importantes est l’évitement d’obstacles.

En omettant cette contrainte, le robot rentrera en collision avec chaque objet présent au sol, avec le mobilier, ou les murs. Pour traiter cette nécessité d’évitement, le robot doit tout d’abord savoir où sont les obstacles, et c’est ainsi qu’intervient le LiDAR : LiDAR pour Light Detection And Ranging, Détection de la lumière et télémétrie en français. C’est un capteur qui va permettre au robot de savoir à quelle distance et quelle direction se trouve un obstacle donné.

Une fois l’obstacle repéré, le robot va pouvoir agir en conséquence, soit de manière arbitraire défini par les développeurs en amont, soit par une intelligence artificielle entrainée sur de très nombreuses situations déjà connues. Elle va donc prendre la décision la plus adaptée pour éviter l’obstacle en nettoyant le plus de surface possible.

Une fois l’obstacle évité, le robot reprend le trajet qu’il avait défini au préalable pour n’oublie aucune partie de la pièce dans lequel il se trouve.

Tous les calculs nécessaires à ses déplacements et aux traitements d’obstacles présents sont effectués par un microcontrôleur inséré à l’intérieur du robot et envoie différentes vitesses aux moteurs des roues afin de prendre telle ou telle direction, ou bien de ralentir ou accélérer.

Généralement, ces robots sont accompagnés d’une application dans l’optique de suivre ce que le robot a prévu de nettoyer, de créer des pièces et changer la priorité de nettoyage pour changer l’ordre de passage du robot par exemple.

## **2.2 Solution 2 : robots tondeurs**

Les robots tondeurs, tout comme les robots nettoyeurs de piscine, partagent un principe de fonctionnement similaire à celui des robots aspirateurs. Cependant, l'environnement dans lequel ils évoluent impose des contraintes très spécifiques, ce qui différencie leur conception.

En effet, le robot tondeur doit être conçu pour résister aux éléments extérieurs tels que l'humidité, les intempéries, et parfois même la pluie. L'un des aspects les plus importants à considérer est l’étanchéité du robot. Contrairement à un robot aspirateur, qui évolue principalement dans un environnement sec et intérieur, le robot tondeur fonctionne à l'extérieur et doit donc être protégé contre l'eau et l'humidité.

Outre l’étanchéité, le robot présente les mêmes contraintes qu’un robot aspirateur, il doit éviter les obstacles et donc utilise des LiDAR de la même manière que les robots aspirateurs.

# **3. Solution développée**

## **3.1 Partie matérielle**

Pour réaliser notre projet, nous nous sommes grandement inspiré des robots aspirateurs, et ce pour de nombreuses raisons : les déplacements libres, la détection d’obstacles avec LiDAR (et donc le fait qu’ils soient autonomes), leur taille plutôt réduite, et les composants nécessaires déjà disponibles au CERI.

Ce qui va suivre explique simplement quelle solution nous allons apporter pour notre problématique. L’assemblage et les explications plus techniques viendront dans les sous-sections 5.x.

Nous avons donc décidé de réaliser un robot rond, avec 2 roues, placés de sorte que le centre de rotation soit au centre du robot (Nous verrons après pourquoi ce choix.).

Comme décris dans le nom du projet, nous allons utiliser le radar LiDAR pour la détection d’obstacle.

Passons en revue tout le matériel dont nous avons eu besoin et pourquoi :

**Moteurs** : nous avons utilisé les motoréducteurs avec encodeur EMG30 car ils étaient disponibles au CERI et c’est exactement ce qu’il nous fallait : moteurs à courant continu, incluant des encodeurs permettant d’émettre des impulsions dès que la roue en question tourne dans un sens ou dans l’autre. Cela va nous permettre de connaître la distance parcourue par la roue.

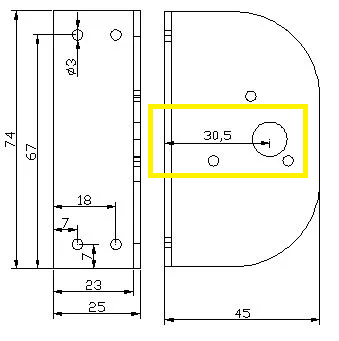
Rapport : 1 :30 | 360 impulsions/révolution | Prix : environ 40€

**Roues** : nous avons utilisé les roues fournies directement avec les motoréducteurs EMG30, c’est-à-dire des roues de 100 mm de diamètres.

Diamètre : 100 mm | Prix : environ 15€

**Une image contenant texte, Dessin d’enfant, dessin, croquis

Le contenu généré par l’IA peut être incorrect.Support pour motoréducteur**: encore une fois nous avons utilisé les supports directement fournis avec les moteurs. Ils permettent de tenir les moteurs à une certaine hauteur pour éviter un contact avec la structure du robot, mais surtout de réduire l’écart entre le sol et le dessous du robot car les roues sont plutôt grandes (ils réduisent de 30.5mm cet écart (voir schéma)).

Prix : environ 5€/unité

(Ce schéma ainsi que tout ceux qui suivront ont été fait par mes soins à la main. Ils ne sont donc pas à l’échelle.)

**Module de commande moteur** : pour finir avec l’ensemble moteur, nous avons utilisé la carte MD25 directement fournie avec les moteurs. Elle permettra de piloter les 2 moteurs que nous avons mis sur notre robot et de lire les informations de leurs encodeurs.

Prix : 60€ | Communication : Série ou I2C

Maintenant que nous avons tout ce qu’il nous faut concernant la motorisation, passons maintenant au **microcontrôleur** : ici nous avons choisi une carte Arduino Uno. Elle a une bonne puissance de calcul pour ce que nous comptons en faire, c’est-à-dire des calculs pour interpréter les données que vont nous renvoyer les capteurs du radar LiDAR (dont on parlera juste après), les calculs de dérivé du robot (nous en parlerons plus tard également), ou encore traiter les données que nous recevons de l’utilisateur, que ce soit par liaison filaire ou Bluetooth.

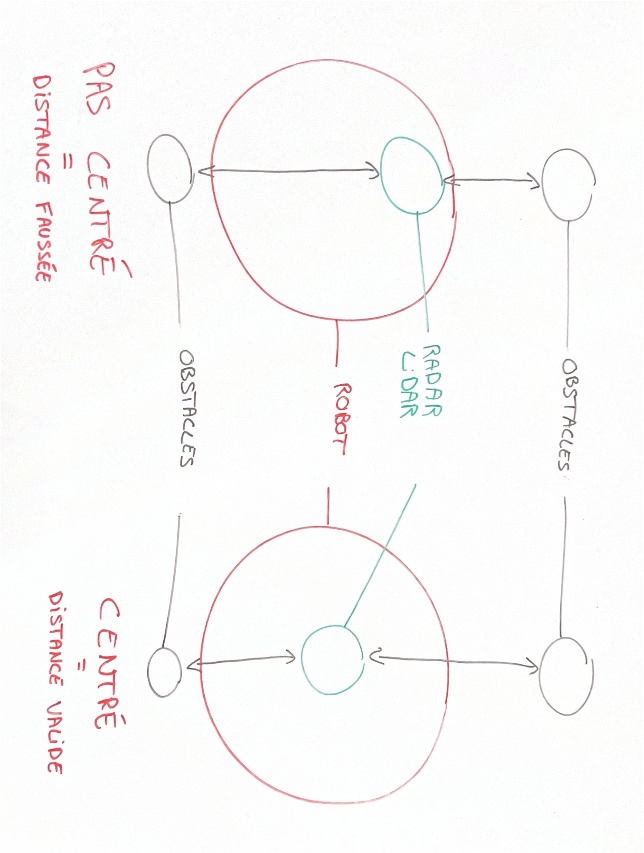
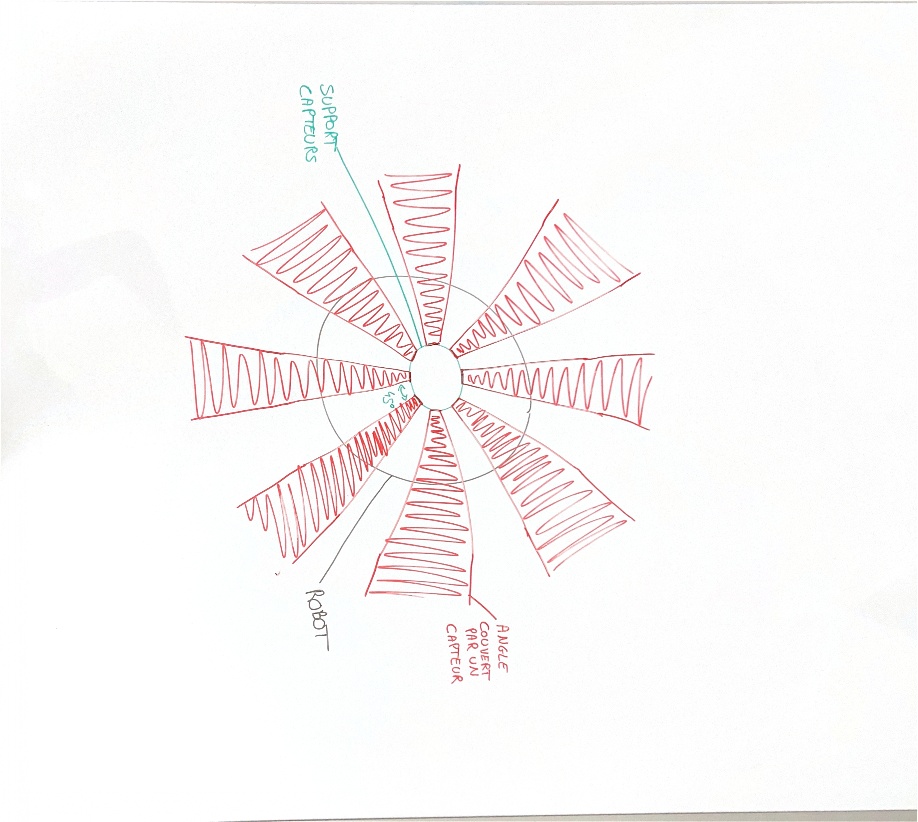
Concernant le nombre de ports disponibles, ce n’est pas un problème : nous avons intégré directement sur la carte Arduino un **Shield** de chez SeeedStudio. Cela permet des branchements plus faciles et surtout un nombre conséquent de port pour avoir une possibilité d’évolution importante.

Prix de l’Arduino Uno : environ 30€ | Prix du Shield : 3.5$ sur le site officiel

**Connectiques** : concernant les câbles, nous avons utilisés ceux déjà présents au CERI.

**Module Bluetooth** : afin de pouvoir communiquer sans fil, nous avons besoin d’un module Bluetooth. Nous enverrons donc les commandes et les ordres donnés par l’utilisateur via une application qui sera connecté en Bluetooth au module, lui-même connecté au robot. Ce dernier les interprétera et agira en fonction. Le nom du module est « Module Bluetooth HC05 ».

Prix : environ 15€

Et enfin l’un des composants les plus importants du robot : le **radar LiDAR Infrarouge (IR)**. Ce dernier a été conçu par le groupe cité plus bas. Il comporte 8 capteurs IR, disposé circulairement à intervalle régulier (45 degrés, voir schéma). Ils couvrent chacun un cône de 25 degrés, sur une distance de 10 à 80 cm. Étant positionnés de manière circulaire, nous pouvons détecter les obstacles tout autour du robot, ses déplacements sont donc sécurisés.

Le radar est disposé au centre du robot, ainsi il jugera les distances de la bonne manière (voir schéma).

Pour terminer, il faut alimenter le tout. Pour cela, nous avons une **batterie** présente sur le robot. Ainsi, le robot n’est pas lié à une prise ou un ordinateur et est libre d’aller où il le souhaite. La batterie actuelle est une batterie LiPo pour Lithium Polymère. **Ces batteries représentent un risque si elles ne sont pas manipulées avec précaution !** Il faut donc en prendre soin et les manipuler calmement et les conserver dans une protection ignifuge afin d’éviter/réduire les risques d’incendie à la suite d’une chute par exemple.

Prix : 30€

Résumé dans un tableau du matériel utilisé avec des liens vers leur site :

|  |  |  |
| --- | --- | --- |
| Moteurs | EMG30 | [Lien](https://www.lextronic.fr/moteur-reducteur-avec-encodeur-emg30-1897.html) |
| Roues | Roues fournies avec moteurs (pack) | [Lien](https://www.lextronic.fr/roue-diametre-100-mm-1896.html) |
| Support moteur | Support fournis avec moteurs (pack) | [Lien](https://www.lextronic.fr/support-de-moteur-1898.html) |
| Module de commande moteur | Carte MD25 (fournie avec moteurs (pack)) | [Lien](https://www.lextronic.fr/module-de-commande-de-moteur-md25-1899.html) |
| Microcontrôleur | Arduino Uno | [Lien](https://store.arduino.cc/products/arduino-uno-rev3?srsltid=AfmBOoqabLJtDmHVTFAapTGaCP5-H30e8nrE0hgriQ3qQ7N-KVYKbIfE) |
| Shield | Base Shield de chez SeeedStudio | [Lien](https://www.seeedstudio.com/Base-Shield-V2.html?srsltid=AfmBOoqsABuFLRxyUPBiFRobZoV6U_JLE39ILdHti3o8yaSCsCUYAOb3) |
| Connectiques | Câbles classiques | Pas de lien |
| Module Bluetooth | Module Bluetooth HC05 | [Lien](https://www.gotronic.fr/art-module-bluetooth-hc05-26097.htm?srsltid=AfmBOooSgXiyi2SCvuKGdrLd3FMusdZ1pTMzQmNeqKwBRSuRhYnz6SXc) |
| Radar LiDAR | Créé par le groupe partenaire | Pas de lien |
| Batterie | Black Lithium 3S | [Lien](https://www.flashrc.com/fr/batteries-de-propulsion-avions-rc/25816-batterie-black-lithium-3s-111v-2200mah-45c-prise-xt60-a2pro-3701383601132.html) |

Abordons maintenant l’aspect logiciel.

## **3.2 Partie logiciel**

Au niveau logiciel, nous nous sommes concentrés sur trois aspects :

* Le programme que contient le microcontrôleur
* Un programme sur ordinateur pour envoyer les directions souhaitées en mode manuel
* Une application sur mobile pour envoyer les directions souhaitées sans fil.

Ces trois aspects seront abordés directement dans les sous-sections 5.x également.

# **4 Organisation**

## **4.1 L’équipe de projet**

Notre équipe était constituée GOMEZ-BARTHEYE Esteban, RANDRIAMANANTENA Samy, et moi-même VILLARD Logan.

Notre équipe a travaillé en étroite collaboration avec le groupe qui était chargé du radar LiDAR IR, qui était composé de HARO Ewan, PANTANI Timothé et FAIVRE D’ARCIER Maxence.

Nous avons pu demander des avis extérieurs à d’autres personnes du groupe CMI, et plus particulièrement HERNANDEZ Mathis avec qui nous avons fréquemment échangés nos avis sur le projet de l’un et l’autre.

Afin de faciliter le travail collaboratif, nous avons choisis la plateforme GitHub pour déposer ce que nous avons créé au fil des semaines.

[Lien GitHub](https://github.com/loganvlr/Robot-LiDAR-Infrarouge)

## **4.2 Répartition des tâches**

|  |  |  |
| --- | --- | --- |
| Responsable | Tâches | Besoins spécifiques pour la tâche |
| Esteban | Modélisation de divers prototypes | Fusion 360 |
| Réalisation de divers prototypes | Utilisation d’outils |
| Samy | Création application mobile | Kotlin, prise en main d’Android Studio |
| Logan | Assemblage des composants électroniques du robot | Apprentissage protocole de communication |
| Programmation (programme robot, programme saisie ordinateur) | C++, Python, bibliothèques spécifiques |
| Implémentation radar LiDAR du groupe partenaire | Communication importante, savoir s’adapter |

## **4.3 Environnement matériel/logiciel**

|  |  |  |
| --- | --- | --- |
| Responsable | Matériel/logiciel | Raisons |
| Groupe | Ordinateurs personnels (portables et fixes) | Support pour pouvoir développer les programmes et applications nécessaires |
| Esteban | Fusion 360 | Bon moteur graphique et simple d’utilisation car beaucoup de ressource disponibles |
| Samy | Android Studio | - |
| Logan | Arduino IDE | Je ne l’utilise que pour téléverser le code vers la carte Arduino Uno |
| GitHub Desktop | Versionnage de mon code en cas de problème + travail collaboratif |
| Suite JetBrains (CLion pour C++, PyCharm pour Python) | J’utilise la suite JetBrains pour coder sur n’importe quel projet, je connais très bien les IDE qu’ils proposent, ils sont très complets. |

Pour le reste du matériel, nous avons toute la liste et le pourquoi de chaque composant dans la section [ci-dessus](#_3.1_Partie_matérielle).

## **4.4 Planification**

VILLARD Logan :

Une image contenant ligne, capture d’écran, Tracé, Parallèle

Le contenu généré par l’IA peut être incorrect.

GOMEZ-BARTHEYE Esteban : ?

RANDRIAMANANTENA Samy : ?