

Génie industriel

PROJET DE FIN D'ANNEE

REALISATION D'UN ROBOT DESINFECTEUR DU SOL

Encadré par :

Mr. HAMED Oussama

Réalisé par :

MAZI Assia

HANNI Nawal

ELBOUHAILI Hajar

Remerciements :

« La reconnaissance est la mémoire du cœur »

Hans Christian Andersen

Nous tenons à remercier tout particulièrement Mr. HAMED Oussama, notre tuteur, pour son soutien, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion.

Nous remercions également le Club-MECATRONICS, qui nous a fourni le matériel nécessaire pour ce projet, et nous a également beaucoup appris sur les défis à relever dans le monde de robotique.

Nous apportons notre plus profonde reconnaissance à Mr. FAROUK Marouane, président du Club-MECATRONICS, d'avoir partagé ses connaissances et ses expériences dans ce milieu, tout en nous accordant sa confiance et sa disponibilité.

Enfin, nous n'oserions oublier de remercier tout le corps professoral de l'ENSAM-CASA, pour le travail énorme qu'il effectue pour nous créer les conditions les plus favorables pour le déroulement de nos études.

Table des matières

Introduction :	5
Chapitre I : Etude préalable du projet	7
1- Description générale du projet :	7
1.1-Le produit :	8
1.2-Utilisation :	8
1.3-Identification du service :	8
1.4-Le milieu environnant :	8
2- Expression fonctionnelle du besoin :	9
2.1 - Expression du besoin :	9
2.2- Diagramme bête à corne :	9
2.3- Diagramme des interacteurs :	11
2.4- Le diagramme FAST :	12
2.5- Cahier de charges fonctionnel :	13
2.6- Diagramme SADT :	14
Chapitre II : Conception du système	16
1- Structure du système :	16
2- Fonctionnement du robot :	17
3- Interfaces utilisateur :	18
4- Description de la partie matérielle :	18
4.1.1- La carte Arduino :	18
4.1.2- La motorisation du Robot :	19
4.1.3.- Le Module Bluetooth :	20
4.1.4- La pompe :	21
5- Description de la partie logicielle :	22
5.1- L'environnement de programmation Arduino :	22
5.2- Programmation du robot :	22

5.3 - L'application App Inventor :	22
Chapitre III : Phase de réalisation	23
1- Contrôle global de robot :	23
a- Déplacement :	23
b- L'injection :	23
c- Désinfection :	23
2- Organigramme général du système :	24
3- Application mobile :	25
4- La forme finale du robot :	27
CONCLUSION :	29
WEBOGRAPHIE :	30
Annexe :	31

Liste de figures :

Figure 1 : Diagramme bête à corne

Figure 2 : Diagramme des interacteurs

Figure3 : Diagramme FAST

Figure4 : Diagramme SADT niveau 0

Figure 5 : Diagramme SADT niveau 1

Figure 6 : Structure générale du système

Figure7 : Vue d'ensemble de la carte Arduino

Figure 8 : Moteur à courant continu DC

Figure 9 : Moteur driver

Figure 10: Module Bluetooth

Figure 11: Pompe

Figure 12 : Organigramme général du système

Figure 13 : Menu principal de l'application

Figure 14 : A propos de l'application

Figure 15 : Menu de contrôle manuel

Figure 16 : Notification permanente de connexion

Figure 17 : Reconnaissance vocale

Figure 18 : L'application en écoute de la commande

Figure 19 : Vues globales du robot

Figure 20 : Vue de dessus de robot

Figure 21 : Vue de dessous du châssis

Introduction :

Depuis bien longtemps, l'humain rêve de créer des machines intelligentes capables d'effectuer des tâches à sa place. Ainsi, les humains auraient plus de temps à consacrer pour leurs loisirs, ou prendraient moins de risques pour effectuer des tâches dangereuses dans des endroits inaccessibles et hostiles. La robotique est l'ensemble des techniques permettant la conception et la réalisation de ces machines-là intitulé « robot ». Un robot est un appareil effectuant, grâce à un système de commande automatique à microprocesseur, une tâche précise pour laquelle il a été conçu dans le domaine industriel, domestique ou scientifique.

La robotique est un ensemble de disciplines (mécanique, électronique, automatique, informatique), elle se subdivise en deux types : les robots industriels et les robots mobiles. Les robots industriels sont généralement fixes, ils sont utilisés dans des nombreuses applications industrielles : l'assemblage mécanique, la soudure, la peinture... Les robots mobiles ne sont pas fixes, ils sont classifiés selon la locomotion en robots marcheurs, à roues, à chenilles... comme ils peuvent être classifié selon le domaine d'application en robots militaires, de laboratoire, industriels et de services. Les robots mobiles sont largement utilisés dans les environnements industriels, le plus souvent pour des tâches répétitives en suivant un chemin bien défini matérialisé parfois par des lignes sur le sol ou par l'utilisation d'amers artificiels. Cependant actuellement il y'a une forte tendance à élargir les milieux où évoluent les robots à des environnements domestiques. Les types d'applications possibles sont innombrables. Cela peut aller des tâches de nettoyage et d'entretien, à une assistance à une personne handicapée dans des tâches d'exploration et de préhension. On parle alors, de façon générale, de robotique d'intérieur.

L'aspect particulier de la mobilité impose une complexité technologique et méthodologique qui s'ajoute en général aux problèmes rencontrés par les robots manipulateurs. La résolution de ces problèmes passe par l'emploi de toutes les ressources disponibles tant au niveau technologique (capteurs, motricité, énergie) qu'à celui du traitement des informations par l'utilisation des techniques de l'intelligence artificielle ou de processeurs particuliers (vectoriel, cellulaires). L'autonomie du robot mobile est une faculté qui lui permet de s'adapter ou de prendre une décision dans le but de réaliser une tâche malgré un manque d'informations préliminaires ou éventuellement erronées. Dans d'autres cas d'utilisation, comme celui des véhicules

d'exploration de planètes, l'autonomie est un point fondamental puisque la télécommande est alors impossible par le fait de la durée du temps de transmission des informations.

Le but de ce travail est la conception et réalisation d'un robot désinfecteur mobile télécommandé à distance par Bluetooth. Ce projet est organisé de la façon suivante :

- Le premier chapitre est consacré à une étude générale du projet.
- Le deuxième chapitre présente la conception de notre propre robot.
- Le troisième chapitre est dédié à la réalisation du robot mobile.

Les objectifs :

- **Objectif technique** : concevoir et réaliser, sur la base du Arduino, un robot nettoyant et désinfectant des sols.
- **Objectifs pédagogiques** : A la fin de ce projet, nous aurons acquis les notions de travail en équipe, exposition et valorisation du travail, acquisition du vocabulaires techniques, utilisation d'une démarche expérimentale, lecture et conception d'un algorithme, reconnaissance de composants électroniques et connaissance de leur utilité, utilisation d'un plan de montage, codage d'un programme informatique ...
- En plus de diverses notions mathématiques, logiques, grammaticales, manuelles, sociales...

Chapitre I : Etude préalable du projet

1- Description générale du projet :

Suite à l'épisode du Coronavirus sont apparues, pour la sécurité de chacun, de nouvelles tâches obligatoires, entraînant souvent de nouvelles charges de travail, et parfois des risques accrus pour certains métiers.

Devoir nettoyer et désinfecter en permanence, provoque une augmentation conséquente de travail pour le personnel de service des établissements déjà très sollicité, et de plus l'expose à un risque de contamination. L'idée est donc de trouver un moyen sans risque d'exécuter ces tâches, comme un complément au travail de nettoyage, sans pour autant remplacer le personnel de service.

Nous les élèves ingénieurs de l'ENSAM_CASA avons mis au point un robot désinfecteur des sols qui limite les gestes humains afin d'aider le personnel de nettoyage pour lutter contre le coronavirus.

Le robot désinfecteur s'inscrit dans une stratégie de sécurisation des mesures barrières face à la Covid-19. La désinfection des surfaces et des endroits difficiles se fera à travers un robot mobile commandé et sera basée sur la diffusion d'une solution désinfectante. Cette diffusion permettra d'empêcher l'accumulation de la saleté.

Le dispositif est un petit appareil prenant la forme munis d'une brosse pour un nettoyage optimal, efficace et sécurisé et notamment dans les coins. En même temps qu'il se déplace, il nettoie et désinfecte les saletés présentes sur le sol. Il peut atteindre avec une grande dextérité des endroits difficiles à nettoyer, comme l'espace sous les lits ou les tables.

Le projet de robot consiste à construire et développer un système robot complet télécommandé intégrant une reconnaissance vocale. L'élément central du projet est un robot (au départ en kit...) possédant un système autonome de contrôle pour les déplacements. Nous devons développer un système permettant de télécommander ce robot. Pour cela nous devons maîtriser les étapes de développements mécaniques électriques et de programmations logicielles de tous les éléments qui vont intervenir dans ce système. Cette télécommande se fera si possible par reconnaissance vocale. Suivant notre aisance nous ajouterons un certain degré d'intelligence pour les déplacements du robot en autonome.

1.1- Le produit :

Le projet c'est un robot désinfecteur de sol inscrit dans une stratégie de sécurisation des mesures barrières face à la Covid-19. En ce projet on a 2 modes de fonctionnements, commandé manuellement par des touches et commandé par la reconnaissance vocale. On peut choisir une de ces 3 modes à partir d'une application mobile, qui est programmé spécialement pour ce projet, compatible avec tous les appareils électroniques mobile qu'ils supportent les fichiers Apk, facile à utiliser et exploitable par tous les types d'utilisateurs.

Après choisir le mode de fonctionnement on peut contrôler le robot qu'avec l'application mobile sans difficulté. Le mode de commande par touches dont l'utilisation est comme les jeux vidéo, il suffit seulement d'appuyer sur le bouton voulu, et pour la commande en reconnaissance vocale, elle rassemble à des applications célèbres comme « Cortana, Siri ... », il suffit seulement de dire oralement la consigne voulue et le robot va l'exécuter.

1.2- Utilisation :

Ce robot a une large gamme d'utilisation, et surtout dans les endroits où est impossible pour les être humaines de les accéder,

L'utilisation donc n'est pas limité en quelque tâches pour ce robot.

1.3- Identification du service :

Diagramme bête a corne

1.4- Le milieu environnant :

Ce robot désinfecteur pourrait apporter une solution pour les structures dotées de grandes surfaces accueillant du public telles que les hôpitaux, les réfectoires, les salles de cours ayant besoin d'une désinfection des pièces de manière rapide, efficace et sécurisée, grâce à sa conception, sa forme et son autonomie en énergie électrique.

2- Expression fonctionnelle du besoin :

2.1 - Expression du besoin :

Cahier de charges :

Nous avons fixé notre propre cahier des charges en proposant ce sujet. L'idée était donc de créer notre propre robot mobile idéalement commandé à distance soit par télécommande mobile soit par reconnaissance vocale. Nous avons donc pu distinguer les différentes fonctions de notre cahier des charges comme suit :

- Fonction principale : Permettre une intervention de l'homme sur une zone géographique difficilement accessible.
- Fonction secondaire 1 : Être contrôlable et pilotable à distance.
- Fonction secondaire 3 : Disposer d'une autonomie énergétique suffisante.
- Fonction secondaire 4 : Pouvoir circuler dans un environnement hostile.

Comment désinfecter régulièrement un sol avec le même effectif d'agents d'entretien ?

Un robot pour nettoyer les sols et les agents d'entretien utilisent leur temps pour le nettoyage des poignées de porte et le matériel utilisé.

Comment entretenir la propreté du sol en période de la crise sanitaire ?

Avec la crise sanitaire, les sols des locaux des établissements recevant du public doivent être régulièrement nettoyés. Or, le nombre des personnes pour l'entretien des locaux ne peut être augmenté. Or les agents d'entretien déjà présents doivent aussi nettoyer régulièrement les poignées de porte et le matériel utilisé. Ces deux dernières tâches sont précises et réalisables que par l'homme. Ils ne peuvent donc pas assumer correctement ces deux missions.

2.2- Diagramme bête à corne :

L'analyse des besoins faite auparavant se concentrant surtout sur les besoins majeurs du produit, il est ensuite nécessaire d'approfondir pour pouvoir rédiger un document contractuel sur lequel la maîtrise d'ouvrage et la maîtrise d'œuvre s'entendent. Les besoins

devront être exprimés sous forme de fonctions (d'où le nom d'analyse fonctionnelle) et non de solutions pour permettre un choix lors de l'étude technique.

Ces fonctions doivent donc être listées, classées et explicitées. L'outil But-Besoin-Fonction-Moyen permet de lister de manière exhaustive l'ensemble des Besoins et puis y associer les Fonctions puis les Moyens afférents. L'expression du besoin permet d'isoler l'objet étudié afin d'identifier le principal destinataire, sa matière d'œuvre, et d'exprimer sa fonction globale.

Pour établir la bête à cornes, il est essentiel de se poser les trois questions suivantes :

- À qui, à quoi le produit rend-il service ?
- Sur qui, sur quoi agit-il ?
- Dans quel but ? (Pour quoi ?)

L'expression du besoin est un outil de représentation de ces questions fondamentales : Sur quoi agit-il ? Dans quel but ? A qui rend-il service ? L'utilisateur L'environnement Robot de Supervision Permettre à l'utilisateur de superviser un endroit inaccessible à l'aide d'une plateforme Androïde Figure 2 : Diagramme Bête à Corne exprimant le besoin du Robot de Supervision.

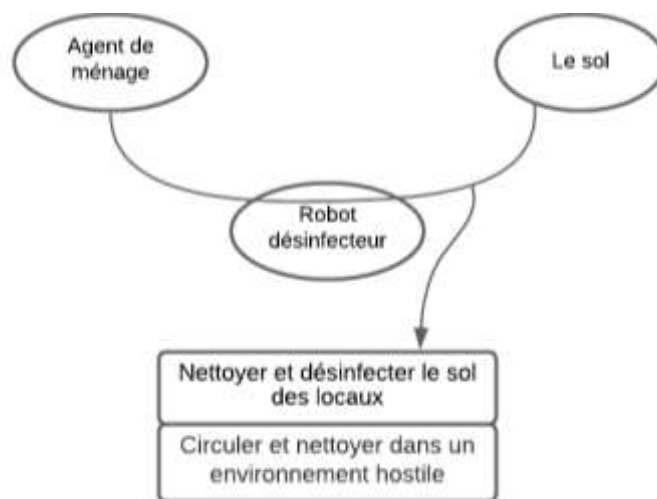


Figure 1 : Diagramme bête à corne

2.3- Diagramme des interacteurs :

En analysant le produit, on peut mettre en évidence les relations entre les différents éléments de l'environnement du produit et identifier les fonctions de service de produit qui conduisent à la satisfaction du besoin.

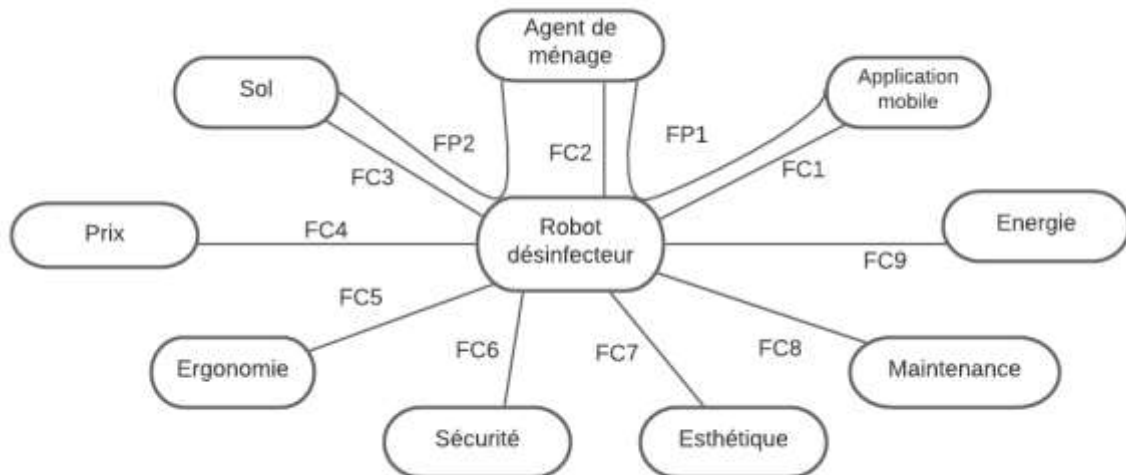


Figure 2 : Diagramme des interacteurs

FP1 : Permettre à l'utilisateur de nettoyer et désinfecter le sol des locaux

FP2 : Permettre à l'utilisateur de le commander avec l'application mobile

FC1 : Être pilotable et contrôlable par reconnaissance vocale ou à distance par une application mobile

FC2 : Permettre à l'agent de ménage le démontage et le montage rapide du flacon du détersif

FC3 : Se déplacer sur un sol (plan, dur, régulier)

FC4 : Être commercialisé avec un prix raisonnable

FC5 : Être maniable

FC6 : Être conforme aux normes de sécurité

FC7 : Avoir un beau design

FC8 : Être maintenable

FC9 : Disposer d'une autonomie énergétique suffisante / être autonome en énergie

2.4- Le diagramme FAST :

Le diagramme FAST constitue alors un ensemble de données essentielles permettant d'avoir une bonne connaissance d'un produit complexe et ainsi de pouvoir améliorer la solution proposée.

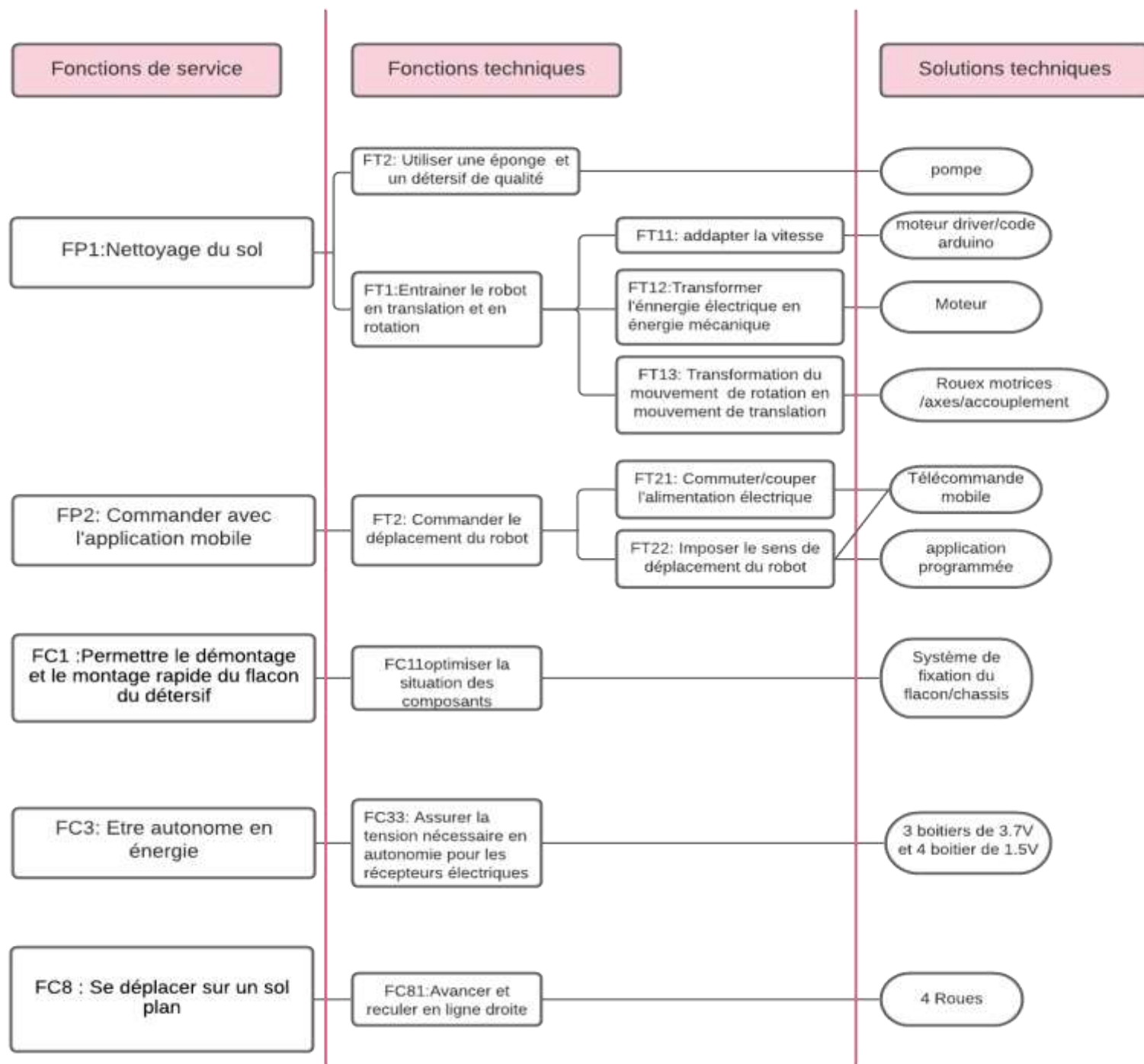


Figure 3 : Diagramme FAST

2.5- Cahier de charges fonctionnel :

Repères	Fonctions	Critères	Niveaux-Flexibilité
FP1	Permettre à l'agent de ménage de nettoyer et désinfecter le sol des locaux	Ejection du détersif et rotation de l'éponge	Débit d'éjection 0.02l/s Vitesse de rotation 12rad/s
FP2	Permettre à l'utilisateur de le commander avec l'application mobile	Télécommande mobile maniable	Déplacement selon l'ordre donné
FC1	Être commandé par une application mobile programmée	<ul style="list-style-type: none"> • Appareil • Système d'exploitation du mobile • Communication 	<ul style="list-style-type: none"> • Smartphone • Android • Bluetooth
FC2	Permettre le démontage et le montage rapide du flacon du détersif	Ne pas endommager le robot ou les autres composantes	Système de fixation permettant le montage et le démontage rapide.
FC3	Se déplacer sur un sol(plan, dur, régulier)	Ne pas glisser Matière du sol	Tous types de sol horizontale
FC4	Être commercialisé avec un prix raisonnable	Prix de fabrication	Ne pas dépasser 3000 DH
FC5	Doit s'utiliser facilement	Maniable et ergonomique	Maniable
FC6	Être conforme aux normes de sécurité	Nettoyer sans blesser personne ou endommager les biens	Normes de sécurité Décret n° 89-662 du 12/09/89 relatif à la prévention des risques résultant de l'usage des jouets (JORF du 15/09/89) Modifié par le décret n° 96-796 du 6/09/96 (JORF du 13/09/96).
FC7	Avoir un beau design	<ul style="list-style-type: none"> • Taille • Masse • Aspect 	<ul style="list-style-type: none"> • Dimensions : 40cm*30cm*20cm • Poids : <= 600g • Doit être attractif
FC9	Être autonome en énergie	<ul style="list-style-type: none"> • Type de générateur • Type de technologie • Autonomie 	<ul style="list-style-type: none"> • Pile ou batterie rechargeable • Toute technologie permettant d'assurer une autonomie suffisante >= 30 min

Tableau 1: Critères d'appréciation proposées.

2.6- Diagramme SADT :

Afin d'étudier l'aspect automatisé nous utilisons le diagramme SADT ci-dessous. La méthode SADT (Structured Analysis and Design Technique) est un outil graphique associé à une méthode d'analyse descendante modulaire et hiérarchisée (Design se traduit ici par conception). Il permet de représenter un modèle (image de la réalité) du système réel. Elle s'appuie sur la mise en relation de ces différents flux avec les fonctions que remplit le système.

Représentation graphique (Actigramme) :

La fonction globale assurée par le robot est «Nettoyer et désinfecter le sol des locaux» ce robot est prolongé dans un milieu dans lequel certains flux s'établissent :

- Flux entrant : le liquide qui est placé dans le réservoir.
- Flux sortant :
 - Le liquide qui est pompé par la pompe vers l'extérieur.
 - Le mouvement de la roue.
 - La rotation d'éponge.
- Flux d'information : ce sont l'ensemble d'informations échangées entre la carte Arduino et les différents moteurs drivers.
- Flux d'énergie : c'est l'énergie électrique interne assurée par les piles, mécanique donnée par le moteur...

Contraintes de pilotage :

C : indique la configuration (programmation) dans notre cas le programme interne qui est installé sur la carte Arduino.

R : indique le réglage (vitesse, course, paramètre...) dans notre cas ces réglages tels que le sens de rotation des roues et la vitesse de rotation sont intégrés dans le programme Arduino.

E : indique l'exploitation (démarrage et arrêt) dans notre cas cette exploitation se fait soit par reconnaissance vocale ou par télécommande mobile.

Contraintes de commande :

W : indique l'énergie (mécanique, électrique, pneumatique...) dans notre robot on utilise deux énergies mécanique et électrique.

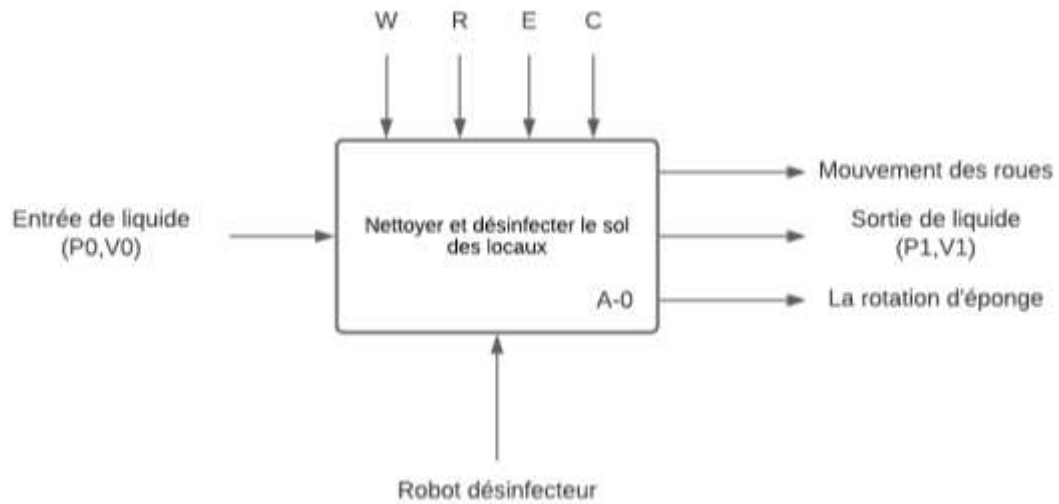


Figure4 : Diagramme SADT niveau0

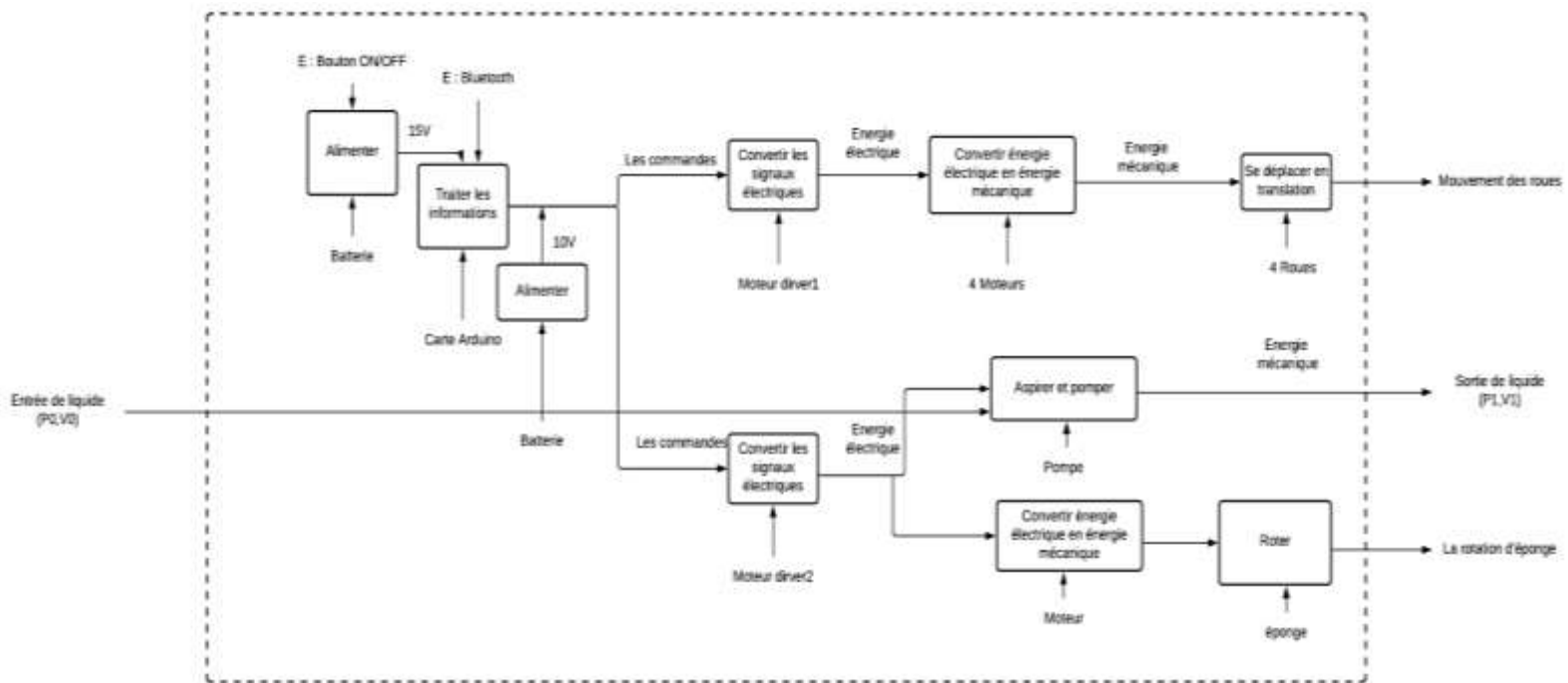


Figure 5 : Diagramme SADT niveau1

Chapitre II : Conception du système

1- Structure du système :

Le système conçu comporte un bloc principal constitué d'une carte Arduino pour l'acquisition d'informations à partir d'un module Bluetooth connecter à l'application mobile et faire commander les actionneurs (moteurs des roues, moteur de l'éponge et la pompe).

Le schéma de ci-dessous illustre la structure globale de notre système :

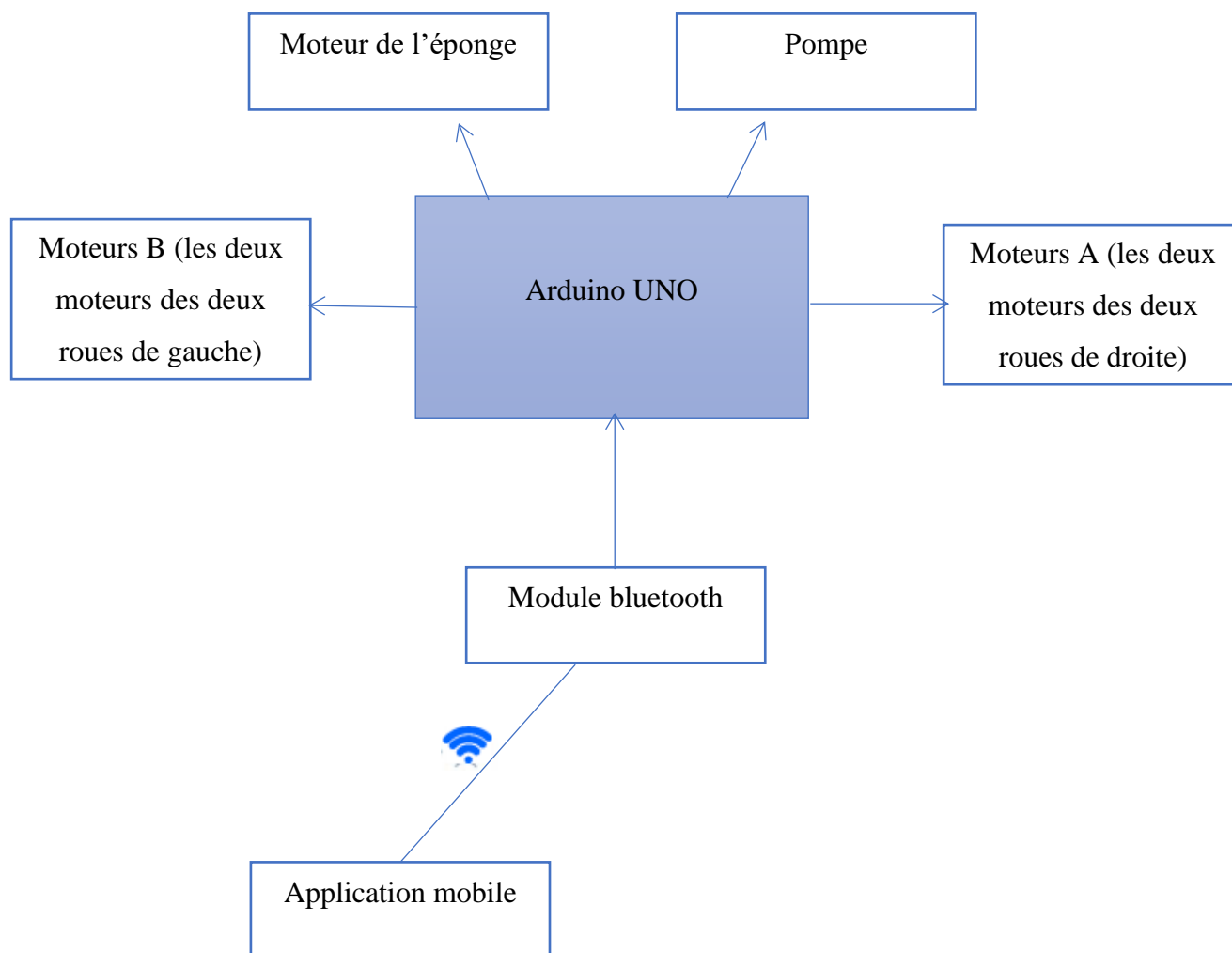


Figure 6 : Structure générale du système

2- Fonctionnement du robot :

Le projet du robot désinfecteur a pour but de construire et concevoir un robot télécommandé qui peut se déplacer en fonction des ordres données par son utilisateur (déplacements vers l'avant, vers l'arrière, vers la gauche et vers la droite), via une application mobile installée sur un dispositif support des applications de type APK et possédant la technologie Bluetooth pour assurer la communication entre le robot et l'application, cette dernière va garantir le contrôle du déplacement du robot et la gestion de l'opération de la désinfection par l'utilisateur.

Le boîtier de commande du robot qui va analyser les consignes d'utilisateur et gérer tous les composants du robot, est la carte Arduino et c'est elle qui va commander le déplacement du robot à partir du contrôle de 4 moteurs à courant continu à l'intermédiaire d'un moteur driver, et c'est elle aussi qui va commander l'injection de la solution désinfectrice et la rotation de l'éponge pour repartir le liquide sur le sol à partir d'une pompe et un moteur à courant continu à l'intermédiaire d'un deuxième moteur driver

L'alimentation de notre robot est faite par les piles partie 10V pour alimenter les machines électriques (partie opérative) + 6V pour alimenter la carte Arduino (partie commande).

Sa source d'alimentation sera une pile. Le boîtier de commande permettra de contrôler les différents mouvements du robot. Il sera muni d'une carte d'acquisition de la voix qui permettra d'enregistrer puis d'envoyer les commandes vers le robot. Il contiendra également une antenne émettrice, une prise mini jack pour brancher le micro récepteur (pour la reconnaissance vocale) et un port série pour la connexion avec un ordinateur. Enfin, en fonction de l'aisance de l'équipe de conception et du temps qu'il restera en fin de projet, nous pourrions intégrer un certain degré d'intelligence pour les déplacements du robot en autonome. Cette dernière fonctionnalité permettra également d'obtenir des informations sur les obstacles rencontrés par le robot sur son trajet, et peut être réalisé à l'aide d'un détecteur d'obstacles.

Le robot comprendra les commandes suivantes :

- Avancer
- Stopper
- Reculer
- Tourner à droite
- Tourner à gauche

L'utilisateur doit prononcer des mots enregistrés auparavant qui correspondent aux commandes du robot. La commande demandée arrive dans le boîtier de commande grâce au micro intégré, qui envoie des informations au logiciel grâce au port série. Le logiciel renvoie ensuite l'information au boîtier de commande, qui la renvoie au robot grâce à son antenne émettrice. Le robot exécute alors la commande demandée. On pourra également contrôler le robot grâce aux touches directionnelles de l'interface graphique de l'application mobile qui y est installé. Dans ce cas l'information sera directement transmise de l'application vers le boîtier et du boîtier vers le robot. Ce processus constituera une solution de repli au cas où le système serait compromis (en cas d'erreur dans la reconnaissance de la commande par la carte d'acquisition de la voix).

3- Interfaces utilisateur :

Une application programmée par l'équipe, sera installée dans un dispositif Android et sera capable de passer du mode « reconnaissance vocale » au mode « télécommande » et inversement. L'utilisateur aura le choix entre ces deux modes de contrôle du robot, qui devront permettre de répondre aux mêmes exigences et qui devront intégrer les mêmes fonctionnalités :

- Commandes à travers l'application : nous devons pouvoir envoyer des ordres au robot afin de faire déplacer le véhicule en avant, en arrière, vers la gauche et vers la droite. Une commande sera également implantée pour permettre de stopper le véhicule.
- Télécommande par reconnaissance vocale : tous les déplacements du PC devront pouvoir être effectués par reconnaissance vocale, à travers des mots enregistrés par l'utilisateur (« Avancer », « Reculer », etc.).

4- Description de la partie matérielle :

4.1.1- La carte Arduino :

Une carte Arduino est une petite (5,33 x 6,85 cm) carte électronique équipée d'un microcontrôleur. Le microcontrôleur permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs ; la carte Arduino est donc une interface programmable.

Une carte Arduino est un cerveau qui permet de rendre intelligent des systèmes électroniques et d'animer des dispositifs mécaniques.

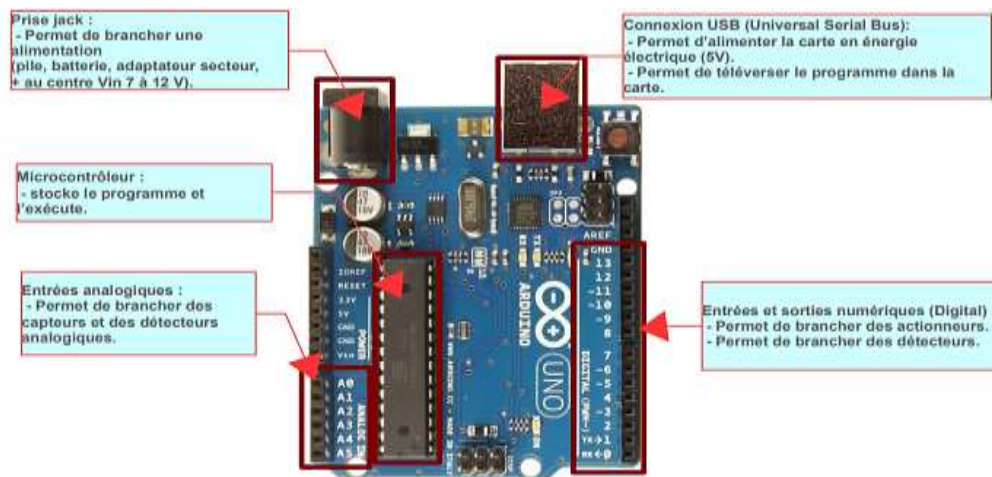


Figure 7 : Vue d'ensemble de la carte Arduino

4.1.2- La motorisation du Robot :

4.1.2.1-Moteur DC :

Un servomoteur permet de réaliser des mouvements de rotation jusqu'à 180°. Mais s'il s'agit de réaliser un robot qui roule, un servomoteur ne peut pas être utilisé pour la propulsion (ou la traction). Il nous faut donc utiliser un moteur qui permet de tourner infiniment dans un sens ou dans l'autre. C'est pour cela on a choisi le moteur à courant continu pour motoriser notre robot.

Caractéristiques :

Taille du moteur : 70mmx22mmx18mm

Moteur poid : environ 50g

Tension : entre 3V et 12V

Vitesse : 15tr/min

Bruit : environ <65db>



Figure 8 : Moteur à courant continu DC

4.1.2.2- Moteur Driver :

Il peut contrôler deux moteurs courant continu. Il est conçu pour supporter des tensions plus élevées, des courants importants tout en proposant une commande logique TTL.

Il peut piloter des charges inductives

- Il peut délivrer jusqu'à 2A en pointe et 20W en continu.
- Il possède son propre circuit d'alimentation logique (permettant d'alimenter l'Arduino)

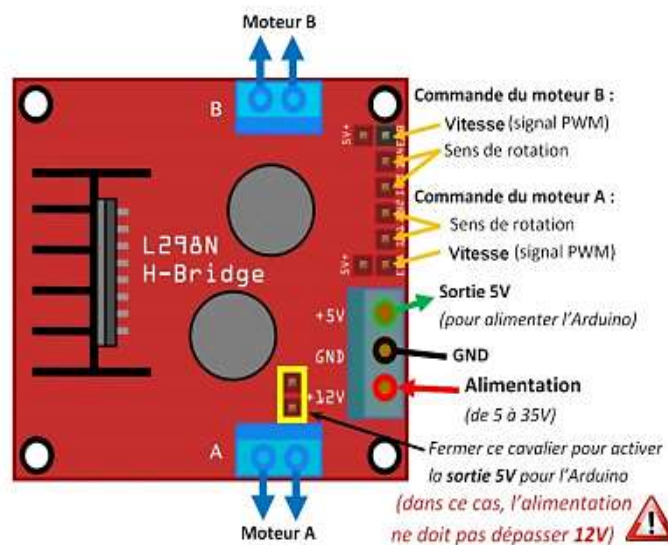


Figure 9 : Moteur driver

4.1.3.- Le Module Bluetooth :

Votre carte Arduino peut communiquer avec d'autres appareils (Smartphone, ordinateur ou un autre microcontrôleur) par liaison Bluetooth en utilisant un module HC-05 (maître/esclave). Cela vous permet de connecter la carte Arduino à un autre système afin d'envoyer et de recevoir des données. La communication par Bluetooth permet, par exemple, de piloter votre robot via une application Smartphone, récupérer les mesures d'un capteur sur votre ordinateur ou encore permettre à deux robots de communiquer entre eux.



Figure 10 : Module Bluetooth

Le module Bluetooth HC-05 présente 6 broches pour permettre d'établir la connexion.

- VCC broche d'alimentation. Typiquement connectée à la broche 5V de l'Arduino.
- GND masse. Typiquement connectée à la broche GND de l'Arduino
- RX broche de réception. Typiquement connecté à la broche de transmission (TX) de l'Arduino
- TX broche de transmission. Typiquement connecté à la broche de réception (RX) de l'Arduino
- State retourne 1 lorsque le module est connecté
- Key ou EN doit être alimentée pour entrer dans le mode de configuration et ne doit pas être connecté pour être en mode communication.

4.1.4- La pompe :

Il s'agit d'un moteur de pompe Arduino submersible a faible cout et de petite taille qui peut être alimente par une alimentation de 2,5 – 6 V. Cela peut prendre jusqu'à 120 litres par heure avec une très faible consommation de courant de 220 mA. En connectant simplement le tube a la sortie du moteur.

On le plonge dans l'eau et on l'alimente. En assurant que le niveau d'eau est toujours plus élevé que le moteur pour ne le pas endommager.



Figure 11 : Pompe

5- Description de la partie logicielle :

5.1- L'environnement de programmation Arduino :

L'environnement de programmation Arduino appelé IDE est distribué sur le site d'Arduino (compatible avec Windows, Linux et mac), A noté qu'on peut associer au compilateur Arduino d'autre programme saisi avec Arduino (extensions pour code blocks, visual studio, Eclipse, etc.). Et la carte Arduino utilise le langage de programmation C/C++ parce qu'il est facile à utiliser et très efficace. L'interface de l'IDE Arduino est plutôt simple (voir la figure 26), il offre une interface minimale et épurée pour développer un programme pour les cartes Arduino.

5.2- Programmation du robot :

En utilisant l'IDE de l'Arduino, on peut programmer notre robot, il suffit seulement d'avoir quelque compétence basique du langage de programmation nommée précédemment au rapport, et savoir aussi utiliser les instructions spéciales des composants qu'ils sont liés à l'Arduino. Avant de commencer la programmation il faut connaître précisément la logique de fonctionnement de robot afin d'obtenir la fonction voulue. Le code commence par l'ajout des bibliothèques des composants et la déclaration de toutes les variables, qu'ils sont utilisés dans le programme.

La configuration et l'initialisation des pins et la détermination de la vitesse du port série, on les fait au fonction setup. Pour faciliter et obtenir un code, on a utilisé plusieurs fonctions que chacune a un rôle précis, les fonctions utilisés sont : « forward » : avancer, « reverse » : reculer, « Left » : tourner à gauche, « right » : tourner à droite, « stop » : s'arrêter, « ultrason » : Activer le capteur ultrason» : La fonction Loop ici, a un rôle de gérant de toutes les autres fonctions ajoutées, l'information venue de l'application mobile se traite à cette fonction puis elle appelle d'autres fonctions selon la commande reçoit. Vous trouverez le code en annexe.

5.3 - L'application App Inventor :

Comme nous avons déjà mentionné, le robot sera équipé par un modèle Bluetooth pour le fonctionnement du pilotage à travers la liaison sans fil Bluetooth. Pour cela on a réalisé une interface graphique à l'aide de l'application App Inventor pour créer une application android de commande à distance.

Chapitre III : Phase de réalisation

1- Contrôle global de robot :

a- Déplacement :

Pour faire déplacer notre robot, il faut faire actionner les moteurs à courant continu, on peut avoir les sens de directions suivantes :

i. En Avant :

- Déplacement en avant : il faut que tous les moteurs soient alimentés dans un seul sens ;
- Avant-droite : il faut actionner les moteurs du côté gauche en avant, et les moteurs du côté droit en arrière ;
- Avant-gauche : il faut actionner les moteurs du côté droit en avant, et les moteurs du côté gauche en arrière.

ii. En Arrière

- Pour le recul du robot, il faut que tous les moteurs actionner dans des sens opposés.
- Arrière-droit, il faut actionner les moteurs de côté droits en avant, et les moteurs de côté gauche en arrière.
- Arrière-gauche : il faut actionner les moteurs du côté droit en avant, et les moteurs du côté gauche en arrière.

b- L'injection :

Pour faire injecter la solution, il faut faire actionner la pompe.

c- Désinfection :

Pour faire désinfecter le sol, il faut faire actionner le moteur a courant continu, pour avoir la rotation de l'éponge.

2- Organigramme général du système :

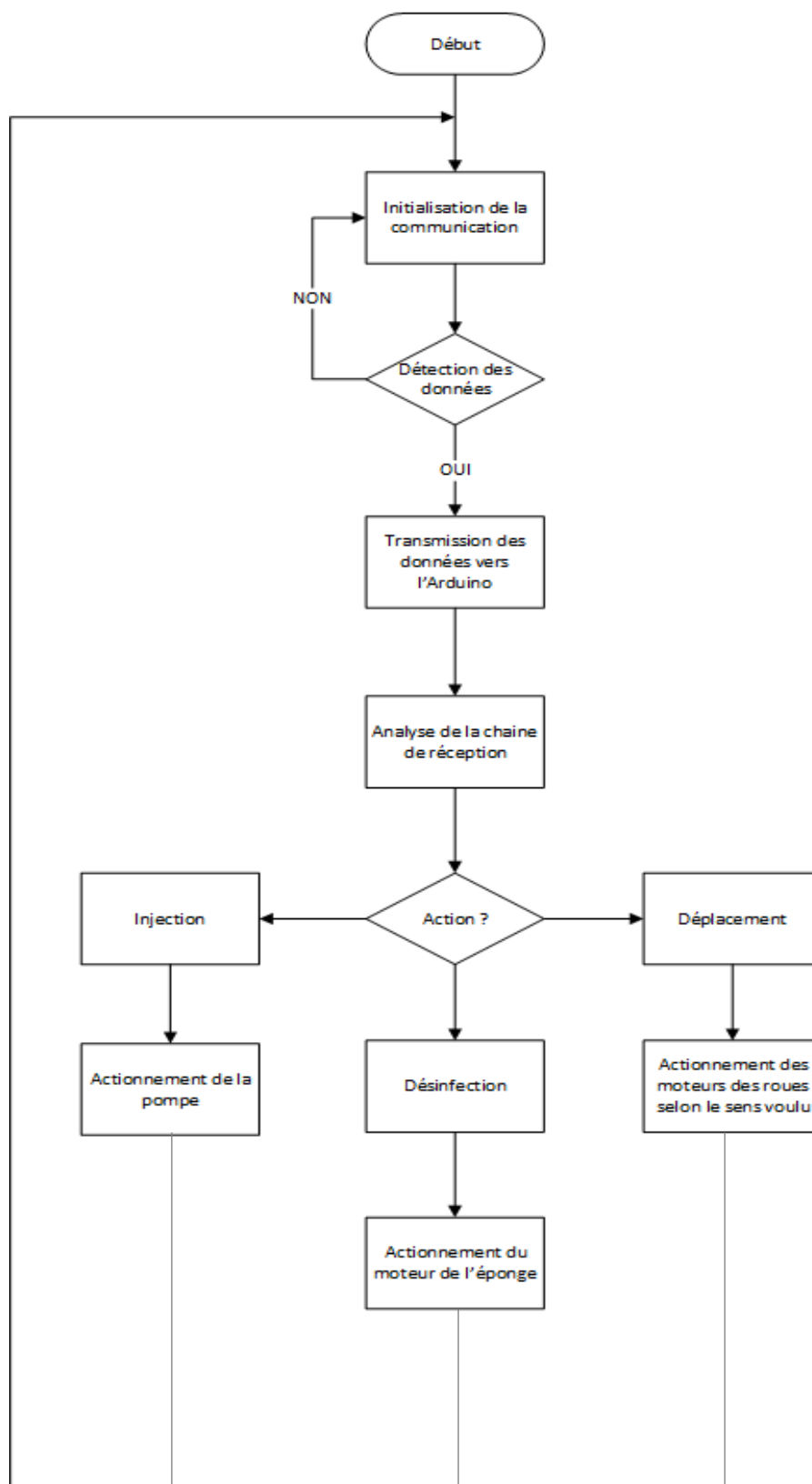


Figure 44 : Organigramme général du Système

3- Application mobile :

L'application permet la commande des moteurs du robot. On désire commander le robot à partir de l'application via un smartphone. L'aspect de l'application est le suivant :

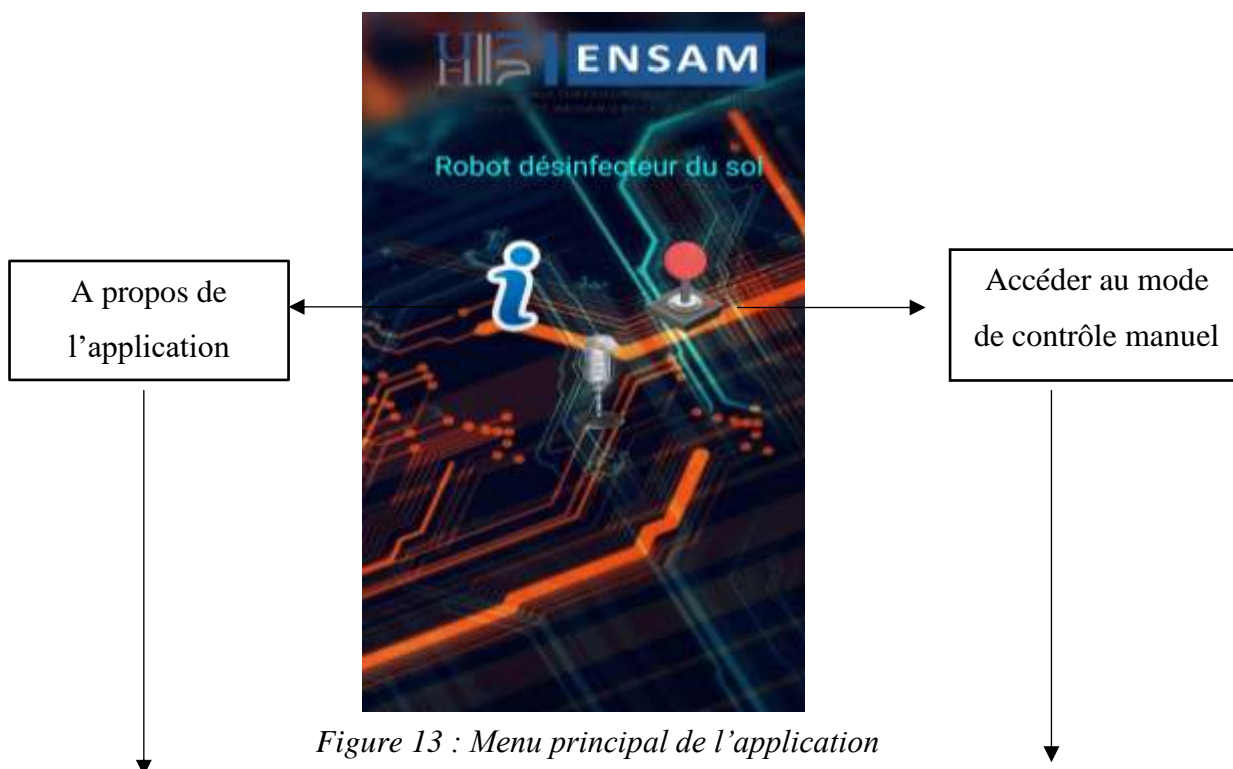


Figure 13 : Menu principal de l'application



Figure14 : A propos de l'application



Figure 15 : Menu de contrôle manuel

Pour se connecter au robot il suffit d'activer Bluetooth sur votre smartphone et choisir le module Bluetooth HC-05



Figure 16 : Notification permanente de connexion



Figure 17 : reconnaissance vocale

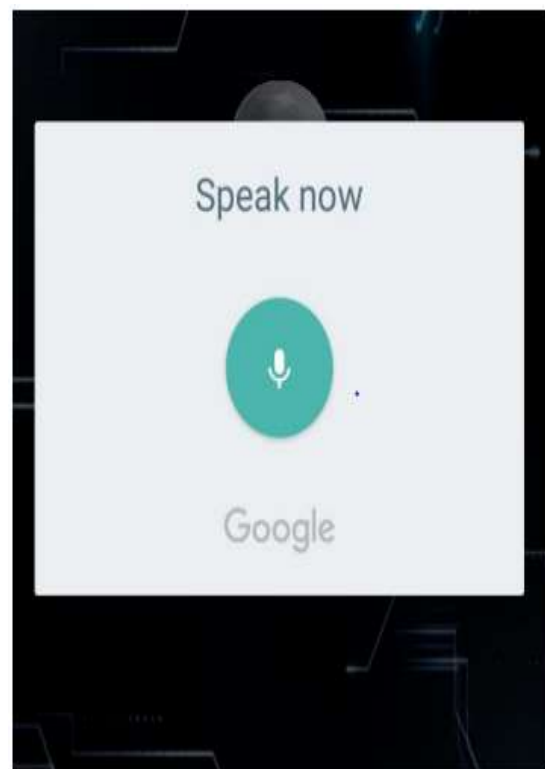


Figure 18 : L'application en écoute de commande

4- La forme finale du robot :



Figure 19 : Vues globales du robot

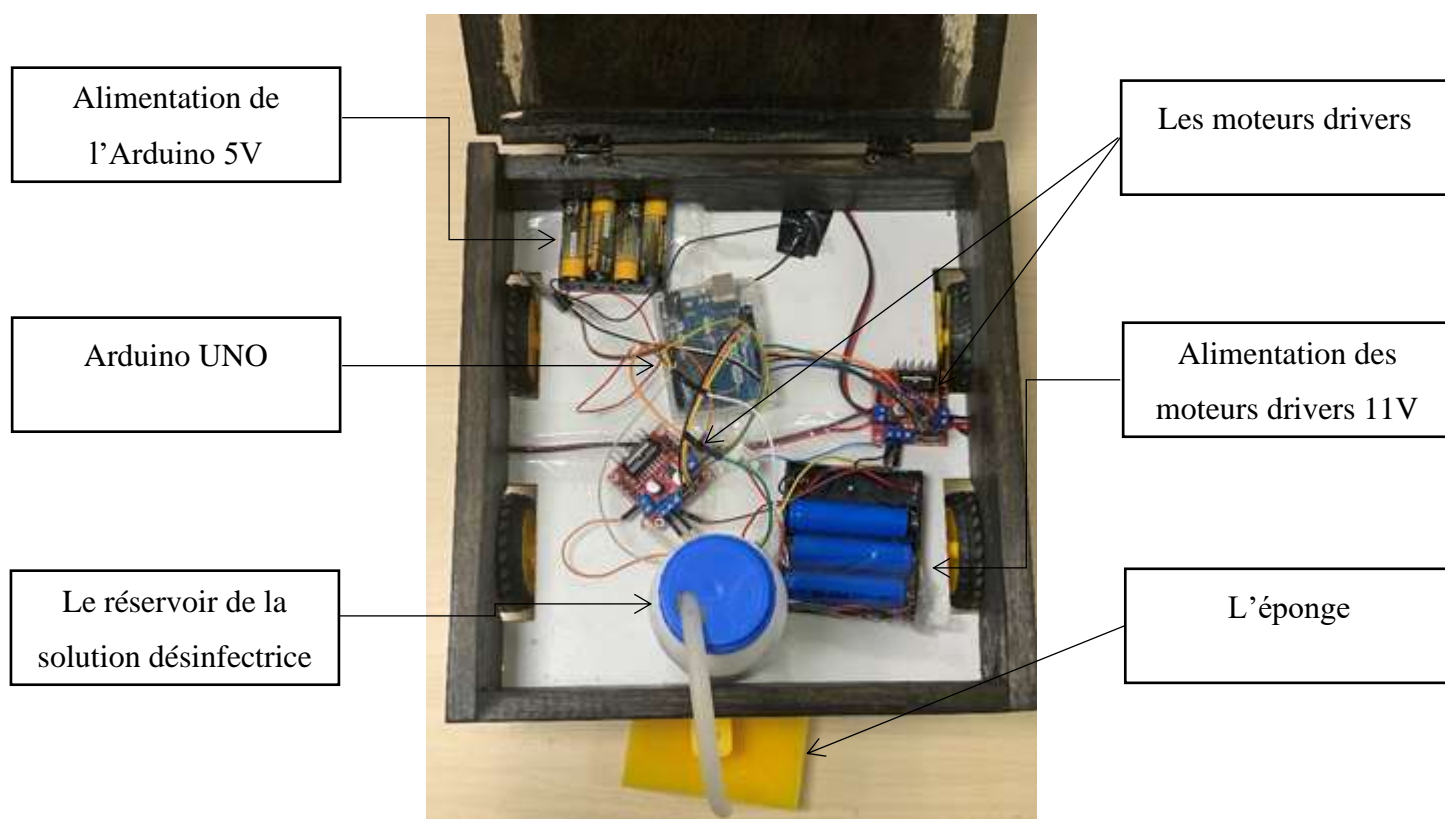


Figure 20 : Vue de dessus du robot

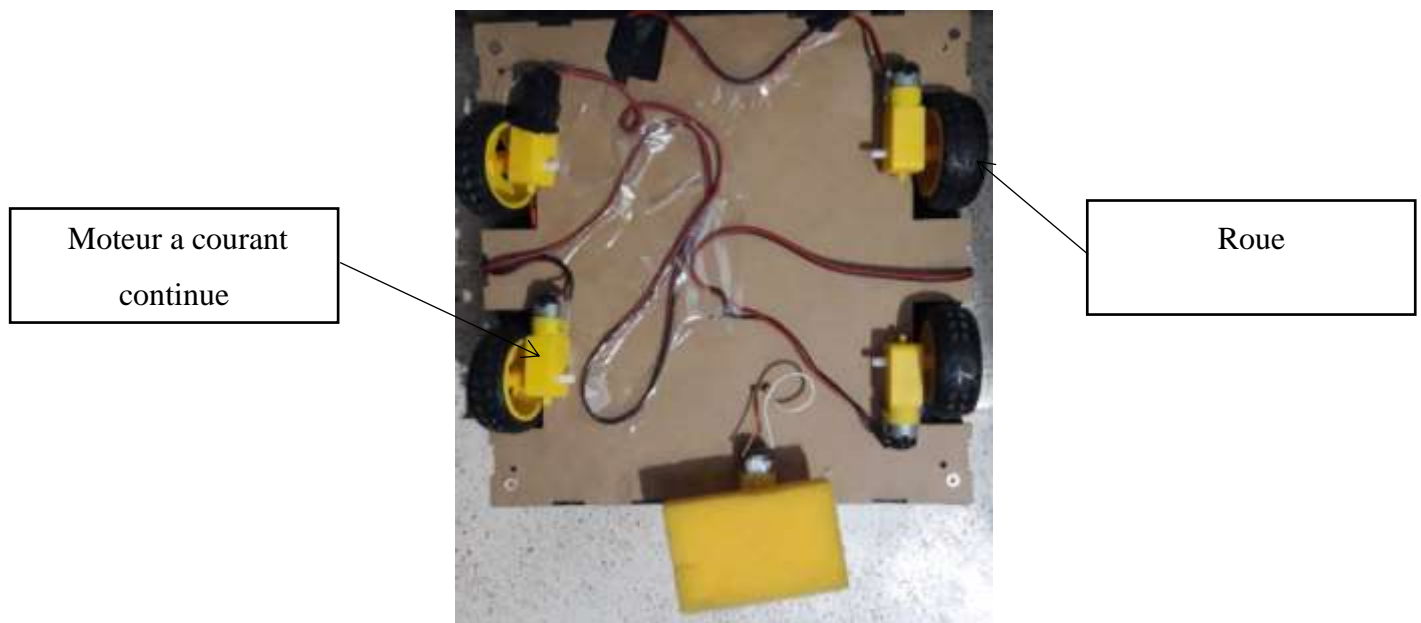


Figure 21 : Vue de dessous du châssis du robot

CONCLUSION :

La conception et la réalisation d'un robot désinfecteur du sol novateur, luttant contre l'accumulation des saletés, et permettant de limiter le risque de contamination par le Covid-19 en réduisant les gestes humains a fait l'aboutissement du présent projet.

Le premier chapitre de ce rapport a été consacré à la description générale de projet et à l'étude fonctionnelle du besoin.

Dans le second chapitre, nous avons détaillé le fonctionnement du robot et implémenté la partie matérielle.

La dernière partie de ce rapport est consacrée pour décrire les différentes étapes de l'investigation pratique 'prototypage' et de développement de l'application mobile, ainsi que les résultats des tests effectués avec notre smart-robot.

A travers la mise en œuvre de notre projet, nous avons pris conscience de ses imperfections et avons ainsi réfléchi à des perspectives d'améliorations et à d'autres idées pour le compléter. Nous suggérons d'analyser d'autres paramètres, tels que la température des personnes présentes dans l'endroit pour pouvoir les déclarer et d'intégrer d'autres capteurs tels que le détecteur de la présence, le détecteur d'obstacles et d'implémenter une caméra pour effectuer un traitement d'images, ainsi qu'étudier l'effet de la diffusion de rayons ultra-violets sur la modification de l'ARN du virus.

WEBOGRAPHIE :

[1] <https://dl.ummto.dz/bitstream/handle/ummto/6326/IcheroufeneOmar.pdf?sequence=1&isAllowed=y>

[2] <https://en.wikipedia.org/wiki/Robot>

[3] <https://fr.slideshare.net/mobile/baddoumohamed7/conception-et-ralisation-dun-robot-de-supervision-acquisition-et-change-de-donnes-distance-via-une-liaison-sans-fil>

[4] <https://www.techniques-ingenieur.fr/base-documentaire/automatique-robotique-th16/applications-en-robotique-42623210/>

[5] <https://openclassrooms.com/fr/courses/2778161-programmez-vos-premiers-montages-avec-arduino/3285333-le-moteur-a-courant-continu-partie-1-transistors-et-sorties-pwm>

[6] <https://arduino103.blogspot.com/2018/07/mit-app-inventor-creer-facilement-des.html>

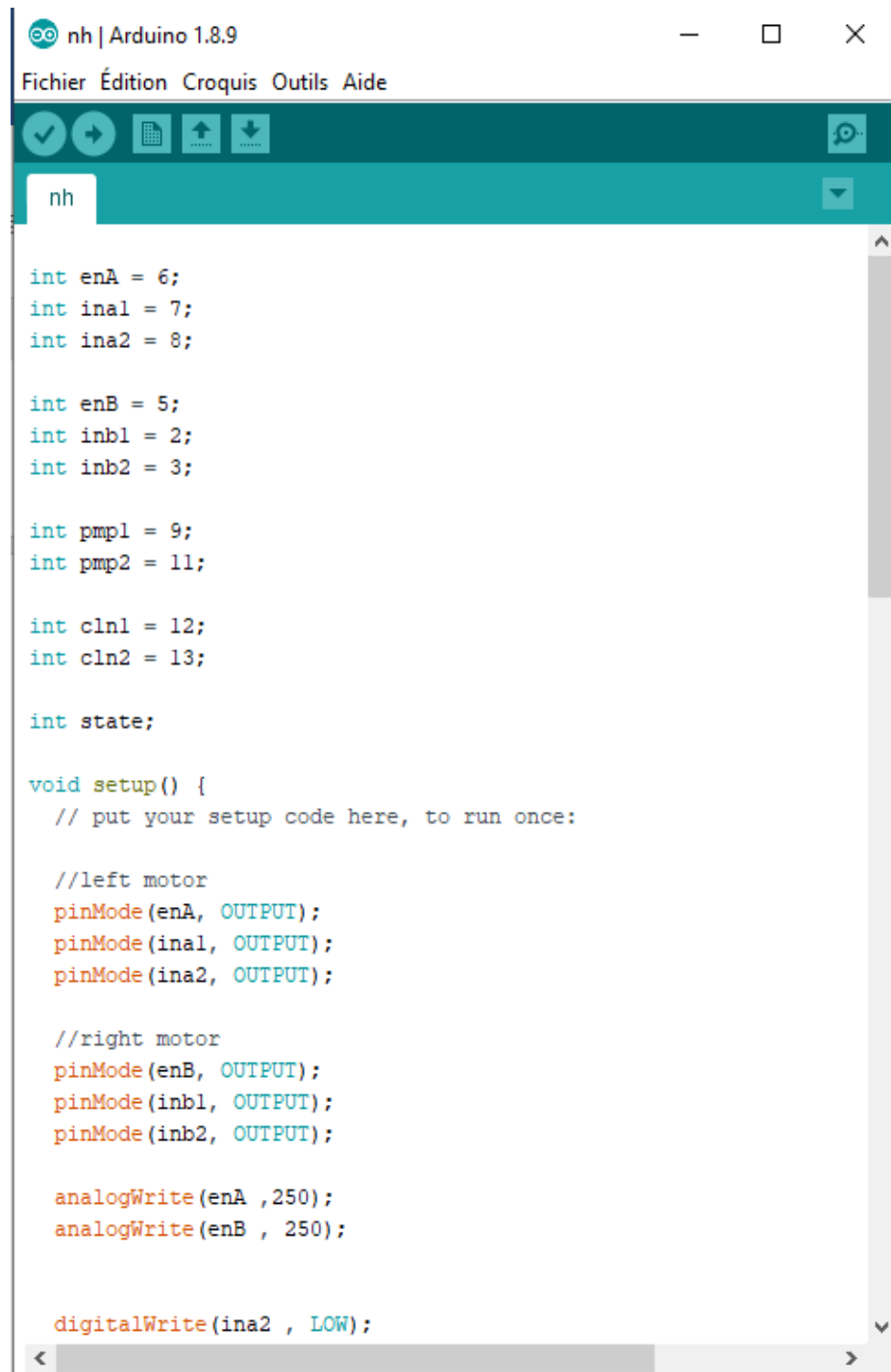
[7] <http://ww2.ac-poitiers.fr/math/spip.php?article807>

[8] <https://www.ionos.fr/digitalguide/serveur/know-how/projets-arduino/>

[9] <https://fr.slideshare.net/mobile/baddoumohamed7/conception-et-ralisation-dun-robot-de-supervision-acquisition-et-change-de-donnes-distance-via-une-liaison-sans-fil>

[10] <https://www.microsonic.de/fr/support/capteurs-%C3%A0-ultrasons.htm>

Annexe :



The image shows a screenshot of the Arduino IDE interface. The title bar indicates the file is named 'nh' and the IDE version is 1.8.9. The menu bar includes 'Fichier', 'Édition', 'Croquis', 'Outils', and 'Aide'. The toolbar contains icons for saving, running, uploading, and other IDE functions. The code editor displays the following C++ code:

```

int enA = 6;
int ina1 = 7;
int ina2 = 8;

int enB = 5;
int inb1 = 2;
int inb2 = 3;

int pmp1 = 9;
int pmp2 = 11;

int cln1 = 12;
int cln2 = 13;

int state;

void setup() {
  // put your setup code here, to run once:

  //left motor
  pinMode(enA, OUTPUT);
  pinMode(ina1, OUTPUT);
  pinMode(ina2, OUTPUT);

  //right motor
  pinMode(enB, OUTPUT);
  pinMode(inb1, OUTPUT);
  pinMode(inb2, OUTPUT);

  analogWrite(enA ,250);
  analogWrite(enB , 250);

  digitalWrite(ina2 , LOW);
  
```


nh | Arduino 1.8.9

Fichier Édition Croquis Outils Aide








nh

```

digitalWrite(ina2 , LOW);
digitalWrite(inal , LOW);
digitalWrite(inb1 , LOW);
digitalWrite(inb2 , LOW);
digitalWrite(cln1 , LOW);
digitalWrite(cln2 , LOW);
digitalWrite(pmp1 , LOW);
digitalWrite(pmp2 , LOW );

Serial.begin(9600);

}

void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available() > 0) // Checks whether data is coming f
    state = Serial.read();
  if (state == '2') {
    digitalWrite(inal , HIGH);
    digitalWrite(ina2 , LOW);
    digitalWrite(inb2 , HIGH);
    digitalWrite(inb1 , LOW);

  }
  else if (state == '3') {
    digitalWrite(ina2 , LOW);
    digitalWrite(inal , HIGH);
    digitalWrite(inb1 , HIGH);
    digitalWrite(inb2 , LOW);

  }
  else if (state == '4') {
    digitalWrite(inal , LOW);

```

 nh | Arduino 1.8.9

Fichier Édition Croquis Outils Aide








nh \$

```

    }
    else if (state == '4') {
      digitalWrite(ina1 , LOW);
      digitalWrite(ina2 , HIGH);
      digitalWrite(inb2 , HIGH);
      digitalWrite(inb1 , LOW);
    }
    else if (state == '1') {
      digitalWrite(ina2 , HIGH);
      digitalWrite(ina1 , LOW);
      digitalWrite(inb1 , HIGH);
      digitalWrite(inb2 , LOW);
    }

    else if (state == '5') {
      digitalWrite(ina2 , LOW);
      digitalWrite(ina1 , LOW);
      digitalWrite(inb1 , LOW);
      digitalWrite(inb2 , LOW);
      digitalWrite(c1n1 , LOW);
      digitalWrite(c1n2 , LOW);
      digitalWrite(pmp1 , LOW);
      digitalWrite(pmp2 , LOW );
    }
    else if (state == '6') {
      digitalWrite(pmp1 , HIGH);
      digitalWrite(pmp2 , LOW );
    }

    else if (state == '7') {
      digitalWrite(c1n1 , HIGH);
      digitalWrite(c1n2 , LOW);
    }
  }
}

```