

# Ecole Centrale Casablanca



---

## Rapport du Projet de Robotique :

*Cahier des charges et spécifications*

---

### [Equipe projet]

Membre de l'équipe	Tuteurs
BEN HASSI Rida	M. DARMOUL Saber
Ben ATTOU El Idrissi Hibat Allah	
CLABESSI Mardoché	
DANHO Jean-Brice Joseph	
EL BARICHI Mohamed	

## Sommaire

### Contents

Sommaire .....	2
Liste de figures .....	4
Liste de tableaux.....	4
Résumé .....	5
1. INTRODUCTION.....	6
2. GESTION DE PROJET .....	6
3. MOTIVATIONS DU PROJET .....	6
3.1. Raison d'être du projet .....	7
3.2. Expression du besoin.....	7
3.3. Parties prenantes.....	7
3.3.1 Clients.....	7
3.3.2 Utilisateurs.....	7
3.3.3 Techniciens de maintenance et de service.....	8
3.3.4 Autres parties prenantes .....	8
3.4. Objectifs du produit/système à développer .....	8
3.5. Diagramme de motivation .....	8
4. CONTRAINTES DU PROJET .....	9
4.1. Environnement de travail prévu .....	9
4.2. Environnement physique de mise en œuvre .....	9
4.3. Environnement applicatif, informatique et informationnel .....	10
4.4. Logiciels prêts à l'emploi.....	10
4.5. Contraintes technologiques .....	10
4.6. Contraintes de calendrier.....	11
4.7 Diagrammes pieuvre.....	12
5. EXIGENCES FONCTIONNELLES.....	13
5.1. Cas d'usage.....	13
5.2. Diagramme de capacités .....	14
5.3. Class Diagramme .....	14
5.4. State Machine Diagramme.....	15
5.5. Activity Diagramme .....	16
6. EXIGENCES NON FONCTIONNELLES .....	16
6.1. Lois, normes et standards influençant le produit.....	16
7. ANALYSE FONCTIONNELLE.....	18

7.1.	Caractérisation des exigences.....	18
7.2.	Etude de l'existant.....	19
7.3.	Analyse de compromis .....	19
7.4.	Functional Analysis System Technique (FAST).....	21
7.5.	Structured Analysis and Design Technique (SADT) .....	21
7.6.	Analyse des modes de défaillance et de leurs effets.....	22
8.	ARCHITECTURE LOGIQUE ET PHYSIQUE .....	22
9.	BIBLIOGRAPHIE – WEBOGRAPHIE .....	27
10.	Annexes.....	<b>Erreur ! Signet non défini.</b>

## Liste de figures

Figure 1. Exemple de diagramme Bête à Corne.....	
Figure 2. Exemple de diagramme de motivation.....	
Figure 3. Exemple de diagramme pieuvre. ....	
Figure 4. Cas d'usage du Robot .....	
Figure 5. Diagramme de capacités.. ....	
Figure 6. Diagramme de class.....	
Figure 7. State Machine Diagramm.....	

## Liste de tableaux

Tableau 1. Répartition des rôles entre les membres de l'équipe projet. ....	Erreur ! Signet non défini.
Tableau 2. Exemple 1 de contrainte technologique.....	10
Tableau 3. Exemple 2 de contrainte technologique.....	11

## Résumé

Ce projet s'inscrit dans le cadre du module d'introduction au génie industriel. Notre objectif était de concevoir un robot en utilisant un kit Lego Mindstorms, capable de ramasser un objet, éviter les obstacles et le déplacer le long d'une ligne noire vers une base située dans la grille, en utilisant un programme spécifique.

Le but de ce projet était de développer nos compétences en manipulation des composants Lego et en robotique, tout en accordant une attention particulière à la conception, à la modélisation et au fonctionnement des systèmes industriels. Nous avons utilisé Python pour programmer le robot en utilisant le logiciel Lego Mindstorms EV3, et avons également créé plusieurs diagrammes pour décrire clairement le fonctionnement et les tâches du robot.

Nous avons suivi des méthodes de codage avec Python, en combinant les fonctionnalités du kit Lego Mindstorms EV3. Nous avons également élaboré plusieurs diagrammes qui définissent principalement les exigences fonctionnelles et non fonctionnelles du robot.

Les résultats obtenus ont été satisfaisants, le robot que nous avons réalisé accomplit efficacement les tâches qui lui sont assignées, tout en tenant compte des critères de qualité, d'hygiène, de sécurité et d'environnement (QHSE).

## 1. INTRODUCTION

Ce projet a pour objectif de familiariser les jeunes élèves de première année du cycle ingénieur de l'école Centrale Casablanca avec un projet industriel en équipe, tout en les initiant aux concepts de conception, de production et d'analyse fonctionnelle. Les étudiants doivent appliquer les théories apprises lors du cours d'introduction au génie industriel.

Au cours des 4 séances de travaux dirigés qui se sont déroulés dans la salle F201 les équipes du projet doivent tout d'abord assembler le robot en utilisant les kits LEGO, puis commencer à programmer le robot pour accomplir les différentes tâches requises qui nous ont été fournies sous forme d'étapes itératives afin de l'accomplissement complète du projet

Les élèves doivent ensuite fournir un rapport de projet sous forme de cahier des charges fonctionnelles, à faire valider par l'école. Ce rapport détaille les caractéristiques du robot et les exigences à respecter.

## 2. GESTION DE PROJET

Membre de l'équipe	Rôles et responsabilités	Sections rédigées
Danho Jean-Brice et El Barichi Mohammed	Conception du programme	3.1-4.7-6.2-6.6-7.6-8.3
Ben Hassi Rida et Ben Attou Hibat Allah	Assemblage du robot	Résumé, abstract, introduction, 3.3- 4.4- 4.6- 5.2- 6.1- 6.8- 7.2, partie QHSE
Clabessi Mardoché	Responsable Design, montage et IT	3.5-4.1-6.5-7.4

*Tableau 1. Répartition des rôles entre les membres de l'équipe projet.*

## 3. MOTIVATIONS DU PROJET

Le projet consiste à concevoir un prototype de robot Lego EV3 qui a pour fonction de détecter, collecter et remettre en place des objets, tout en suivant une ligne noire. Ce projet fait partie intégrante de la formation d'ingénieurs généralistes dispensée à l'École Centrale Casablanca.

Cette initiative d'apprentissage est une évaluation permettant de valider le module "Introduction au génie industriel". Elle offre aux élèves l'opportunité d'acquérir des compétences en robotique et de développer une expertise approfondie dans ce domaine. Grâce à cette expérience, les étudiants peuvent mettre en pratique leurs connaissances et se familiariser avec les concepts clés du domaine du robotisme.

### 3.1. Raison d'être du projet

### 3.2. Expression du besoin

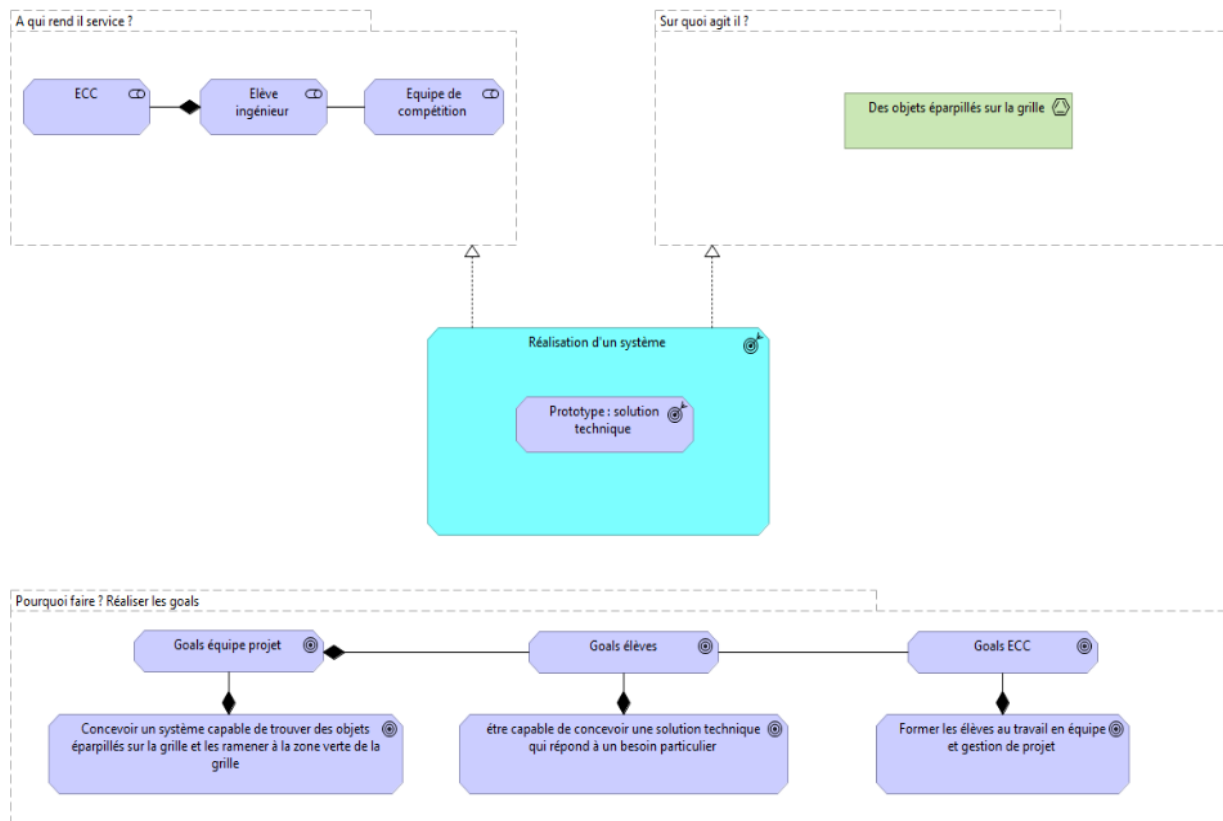


Figure 1. Exemple de diagramme Bête à Corne.

### 3.3. Parties prenantes

Ce projet d'introduction au génie industriel paraît, à première vue, très abstrait qui relève seulement de la programmation, et ne se trouve avoir aucun point commun avec un problème réel mais une analyse des parties prenantes nous permet de conclure le contraire ; il s'avère qu'ils existent des clients, utilisateurs, techniciens et bien d'autres parties prenantes de ce projet qu'on cite par la suite.

#### 3.3.1 Clients

Dans ce contexte, l'école Centrale Casablanca, plus précisément le département de génie industriel agit en tant que client principal en proposant et en lançant ce projet par l'intermédiaire du professeur responsable du cours d'introduction au génie industriel. L'objectif est de permettre aux élèves de mettre en pratique leurs compétences en robotique, en modélisation et en conception en développant une solution qui répond aux exigences définies dans le cahier des charges. Ainsi, l'école Centrale Casablanca joue un rôle essentiel en tant que responsable et moteur de ce projet, cherchant à favoriser l'application des connaissances des étudiants dans le domaine du génie industriel.

#### 3.3.2 Utilisateurs

Les premiers utilisateurs de ce projet, représenté par le prototype de robot chargé de collecter des objets tout en suivant une ligne noire, sont les élèves de l'école Centrale Casablanca. Plus précisément, il s'agit des membres de l'équipe projet, qui sont impliqués dans la conception ainsi que dans la réalisation de la partie matérielle du robot. Ces élèves sont les principaux acteurs qui interagissent avec le prototype,

testent ses fonctionnalités et évaluent sa performance dans le cadre de leur apprentissage et de leur engagement dans le projet.

### **3.3.3 Techniciens de maintenance et de service**

Les membres de l'équipe projet jouent également un rôle crucial en tant que techniciens chargés de la maintenance du robot. Ils veillent à prévenir toute dégradation du robot et sont responsables de son entretien régulier. De plus, ces techniciens peuvent développer des programmes ou des codes informatiques supplémentaires pour faciliter l'utilisation du robot ou fournir des services supplémentaires visant à expliquer ses fonctionnalités. Ainsi, les membres de l'équipe projet assument à la fois le rôle de concepteurs et de techniciens de maintenance, contribuant ainsi à la pérennité et à l'amélioration continue du robot.

### **3.3.4 Autres parties prenantes**

Les autres parties prenantes peuvent être représentées par les professeurs encadrants pendant les différents TDs et TP et dont la contribution aide au développement et l'avancement du prototype tout au long de ces 4 séances de la création à la finalisation du produit. Ils ont été des atouts très importants dans la réalisation dans ce projet grâce à leurs conseils et le suivi de nos différents pas et avancées.

## **3.4. Objectifs du produit/système à développer**

Le robot LEGO EV3 assemblé par l'équipe du projet doit également être programmé pour effectuer les tâches requises. Ceux-ci sont : Trouvez l'objet, ramassez-le et ramenez-le à la position de départ le long de la ligne noire.

## **3.5. Diagramme de motivation**



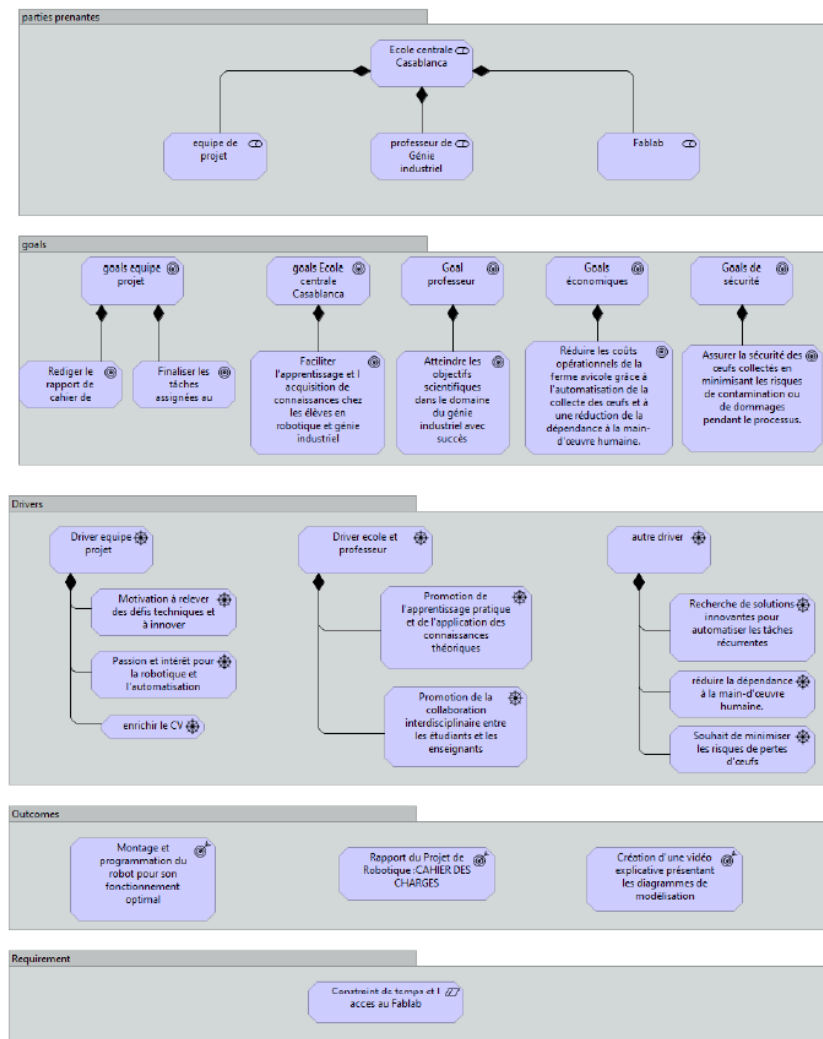


Figure 2. Exemple de diagramme de motivation.

## 4. CONTRAINTES DU PROJET

### 4.1. Environnement de travail prévu

- L'équipe ne peut pas travailler avec le kit de robotique que durant les séances du TD programmer par la scolarité.
- Lorsque le robot se déplace il ne faut pas dépasser la grille.
- Toutes les équipes ne peuvent pas utiliser la maquette du déplacement du robot

### 4.2. Environnement physique de mise en œuvre

- Le robot sera fonctionné dans un tapis de dimensions 2362 x 1118 (mm) ce qui impose une norme sur ses dimensions qui doivent être appropriées à celles du tapis.
- Pour arriver à suivre les lignes et détecter les objets, le robot doit comporter des capteurs à savoir un capteur d'angle et un capteur sonore. Aussi ces capteurs ne sont pas placés dans la place optimale pour la réalisation du projet
- Pour arriver à porter les objets et éviter leur évation, le robot doit avoir des pinces solides et bien placés.

- La batterie du robot se décharge environ en 1 heure de temps d'utilisation sans compter qu'on a 1h45 min et le temps de charge est très lent.
- Les objets que le robot doit ramasser ne sont pas conçus pour les bras qu'il possède.

### 4.3. Environnement applicatif, informatique et informationnel

Le logiciel LEGO Mindstorms, exécuté sur un ordinateur, permet de programmer le robot de manière autonome, sans nécessité de contact visuel direct. Le robot est capable de se déplacer en suivant les instructions du programme téléchargé dans son système. Avant d'effectuer les tâches assignées, les programmes doivent être chargés dans le robot, lui permettant d'opérer de manière autonome et d'exécuter les actions programmées. Ainsi, le logiciel LEGO Mindstorms offre la possibilité de programmer le robot de manière indépendante et de le doter de fonctionnalités avancées.

### 4.4. Logiciels prêts à l'emploi

Pour développer le robot, nous avons utilisé plusieurs produits, logiciels open source et documents existants. Le logiciel clé utilisé était le logiciel LEGO MINDSTORM EV3, qui nous a permis de concevoir le robot, d'assembler ses composants et d'accéder aux codes correspondants pour les différentes parties de la mission. La manipulation de ce logiciel était essentielle pour mener à bien les tâches du projet, qui présentaient un niveau de difficulté moyen.

En plus de LEGO MINDSTORM EV3, nous avons également utilisé le logiciel Anaconda et Visual Studio Code (VSC). Ces logiciels nous ont permis d'implémenter nos propres codes pour exécuter les fonctionnalités et accomplir les missions du robot. L'utilisation d'Anaconda et de VSC nous a offert un environnement de développement convivial et puissant pour écrire, tester et exécuter nos programmes.

En résumé, le logiciel LEGO MINDSTORM EV3 était notre principal outil pour la conception et l'assemblage du robot, tandis que Anaconda et VSC étaient utilisés pour l'implémentation des codes et l'exécution des missions. Ces outils et logiciels ont été essentiels pour la réalisation du projet dans son ensemble.

### 4.5. Contraintes technologiques

*Tableau 2. Contrainte technologique 1.*

Description	La taille du robot doit être petite.
Raison d'être	Le robot fonctionne sur un tapis aux dimensions de 2362 x 1118 (mm).
Critère d'acceptation	Les dimensions du robot ne doivent pas dépasser 200x100 (mm).

*Tableau 3. Contrainte technologique 2.*

Description	La forme de la pince empêche le robot de se déplacer de manière erratique. Le suiveur de ligne est décalé.
Raison d'être	Le robot doit soulever un objet pendant la mission qu'il doit effectuer.
Critère d'acceptation	Lors du levage d'un objet, celui-ci doit être soulevé à au moins 2 mm au-dessus du sol. Le Robot doit toujours suivre la ligne.

#### 4.6. Contraintes de calendrier

Pour les contraintes de calendrier, c'était le nombre de séances de TP accordées à ce projet qui représentait un obstacle vu que pendant la première séance, nous avons bien accompli l'assemblage des pièces du robot mais l'exécution du code était un facteur qui a freiné notre avancement, nous devons bien trouver l'angle de recherche de la ligne noire.

**Explication :** Lorsque nous posons le robot de manière aléatoire sur la grille. Il effectue des tours de 30 degrés de gauche vers la droite jusqu'à ce qu'il trouve la ligne.

Notre technique a été d'augmenter la vitesse de recherche de la ligne noire et ensuite le stabiliser dans le code car ses paramètres par défaut le sortaient de la grille ou le faisaient tourner pendant longtemps.

Pendant la deuxième séance, puisque les robots ne sont pas propres à chaque groupe, nous avons trouvé le nôtre qui a été démonté, donc nous avons dû refaire tout l'assemblage dès le début, mais cette fois plus rapidement. Ensuite nous avons implémenté la fonctionnalité des bras. Cette commande était assurée par le fonctionnement de deux moteurs couplés.

Au début nous commençons par essayer de les faire fonctionner de manière indépendante mais en même temps ceci nous a posé des problèmes au niveau de la synchronisation. Après grâce aux précieux conseils de nos encadreurs nous avons la fonction qui nous permettait de coupler les moteurs et de les coupler convenablement.

Pendant la troisième séance nous nous sommes attelés à soulever des objets. Il est important qu'aucun objet n'est construit par défaut dans le kit. On devait trouver la bonne configuration d'objet pour rendre plausible la suite de nos projets.

La dernière séance s'est agencée autour de la construction du chemin que le robot doit suivre en prenant l'objet. Grâce au code qui nous a été donné comme modèle nous avons pu créer un code général qui permet de parcourir tous les points de la grille.

En conclusion nous avons réussi la mission demandée.

#### 4.7 Diagrammes pieuvre

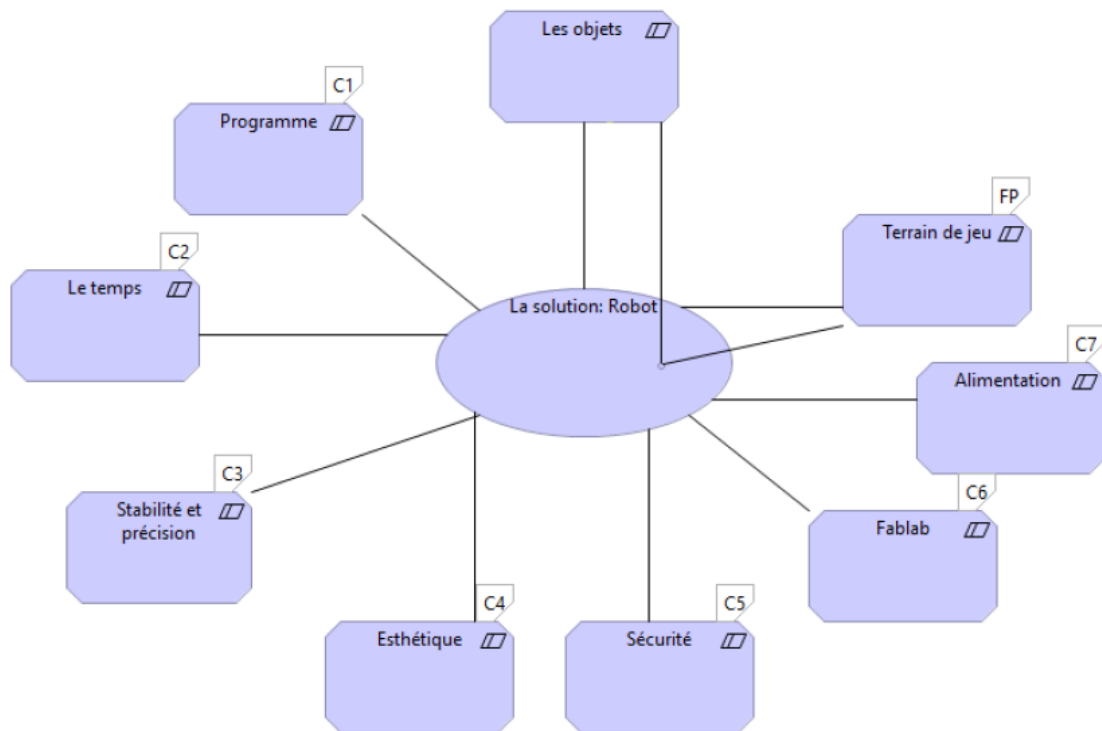


Figure 3. Diagramme pieuvre.

Fonctions	Descriptions
<b>FP1(fonction principale)</b>	<b>Le robot doit être capable de suivre les lignes de la grille afin de collecter les objets et de les ramener à une base dédiée à ces objets</b>
<b>Contraintes et exigences fonctionnelles</b>	
<b>FC1</b>	<b>Les limites de la grille ne doivent pas être dépassées</b>
<b>FC2</b>	<b>Le robot doit être capable de déplacer les objets malgré leurs poids et leurs dimensions</b>
<b>FC3</b>	<b>La durée de réalisation du robot ne doit pas dépasser les 4 séances Dans les salles de TP dédiées</b>
<b>FC4</b>	<b>La construction du robot obligatoirement passer par des étapes spécifiées par le professeur de génie industrielle.</b> <ul style="list-style-type: none"> <li>- 1ere mission : le robot doit suivre une ligne noire en amenant avec lui l'objet qui rencontrera sur sa trajectoire</li> <li>- 2eme mission : Le robot doit ramener l'objet à un endroit précis tout en suivant la ligne noire</li> </ul>
<b>FC5</b>	<ul style="list-style-type: none"> <li>- <b>L'équipe projet doit être constituée de 5 étudiants de 1ere année</b></li> </ul>

FC6	Le matériel est disponible seulement les séances de TP.
FC7	Le robot doit être chargé avant le début du TP sinon son le chargement de sa batterie peut empiéter sur le temps prévu.

## 5. EXIGENCES FONCTIONNELLES

### 5.1. Cas d 'usage

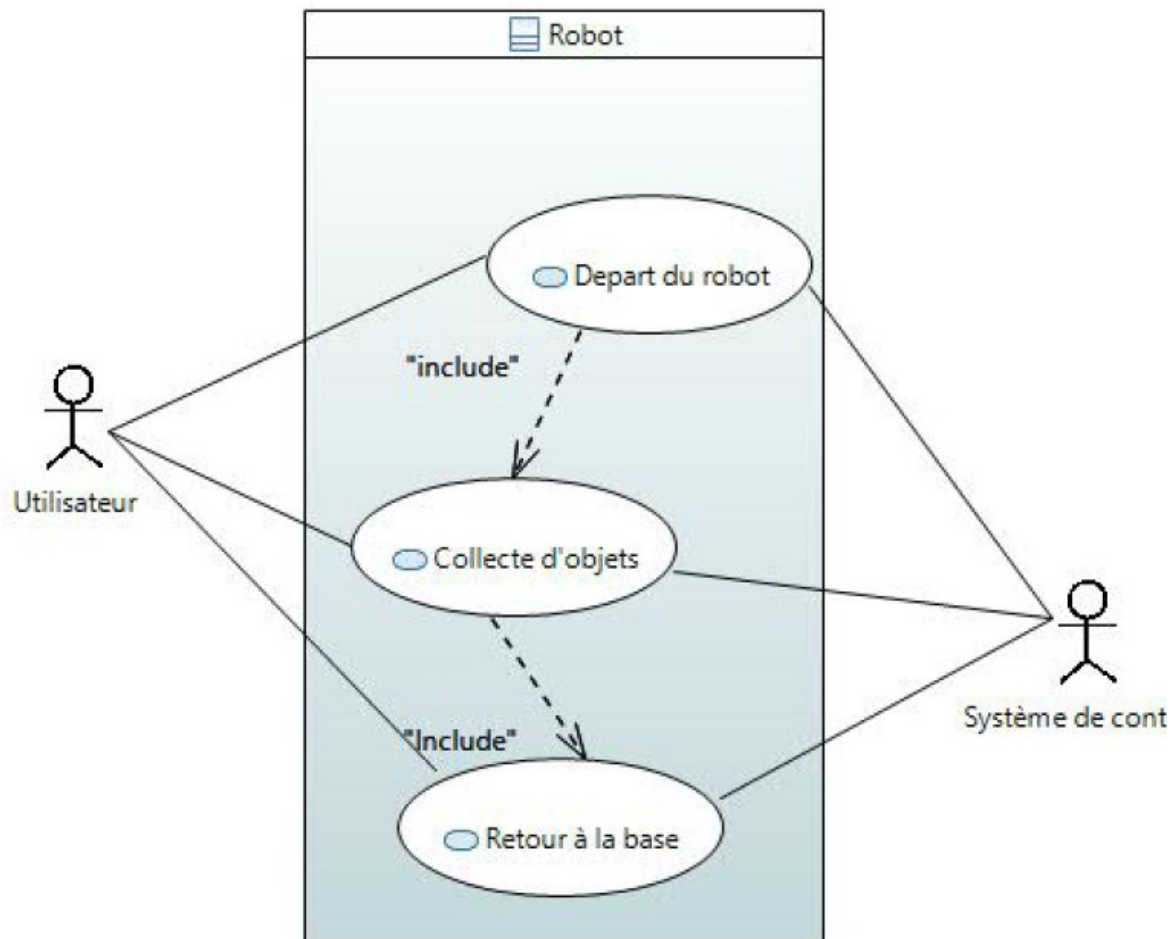


Figure 4. Cas d'usage du robot.

## 5.2. Diagramme de capacités

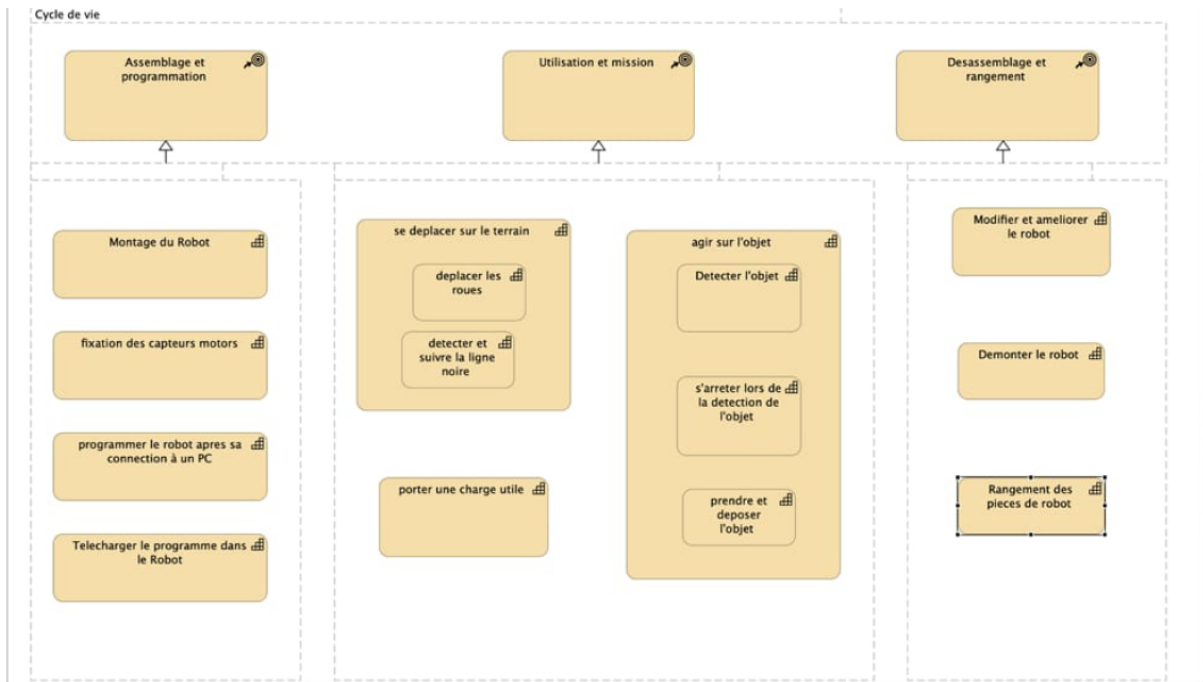


Figure 5. Diagramme de capacités.

## 5.3. Class Diagramme

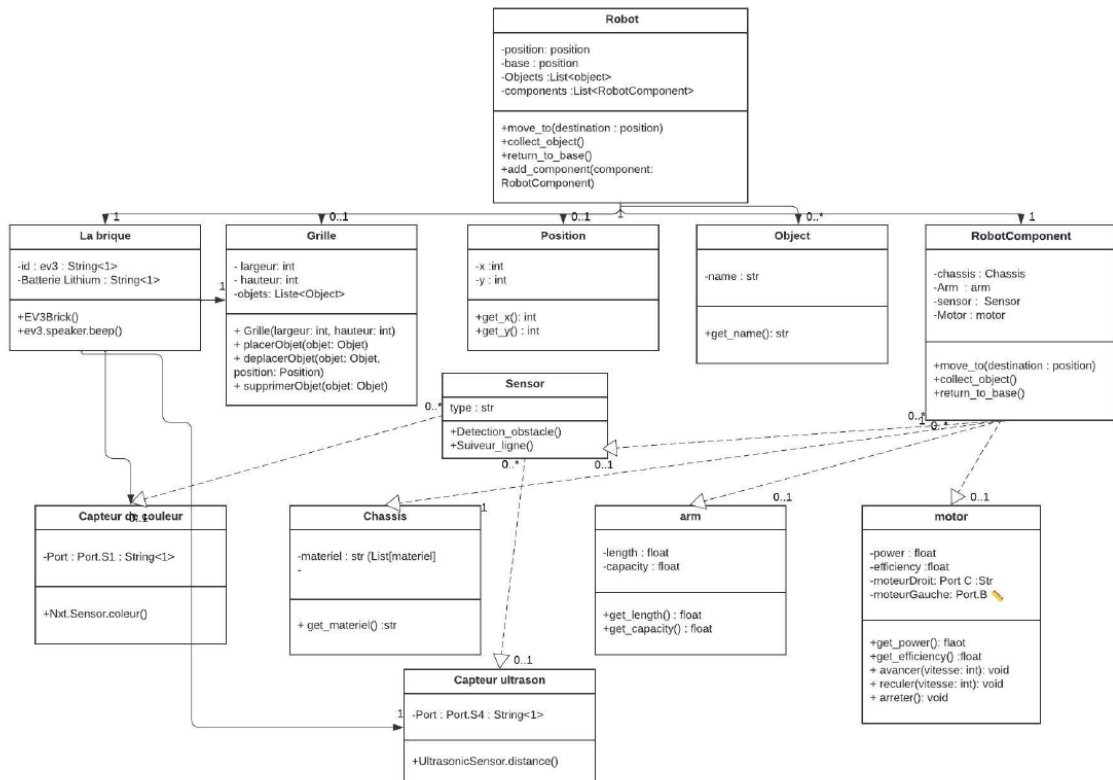


Figure 6. Diagramme de class.

#### 5.4. State Machine Diagramme

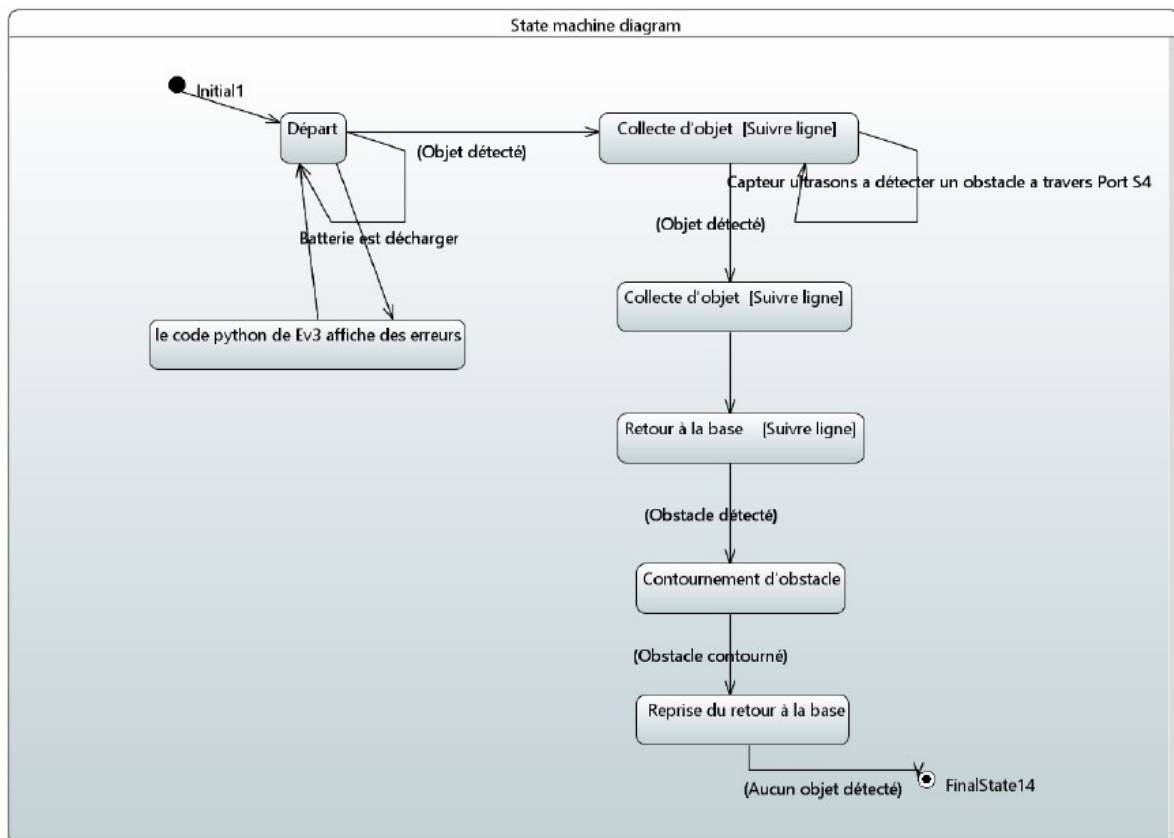


Figure 7. State Machine Diagramme.

## 5.5. Activity Diagramme

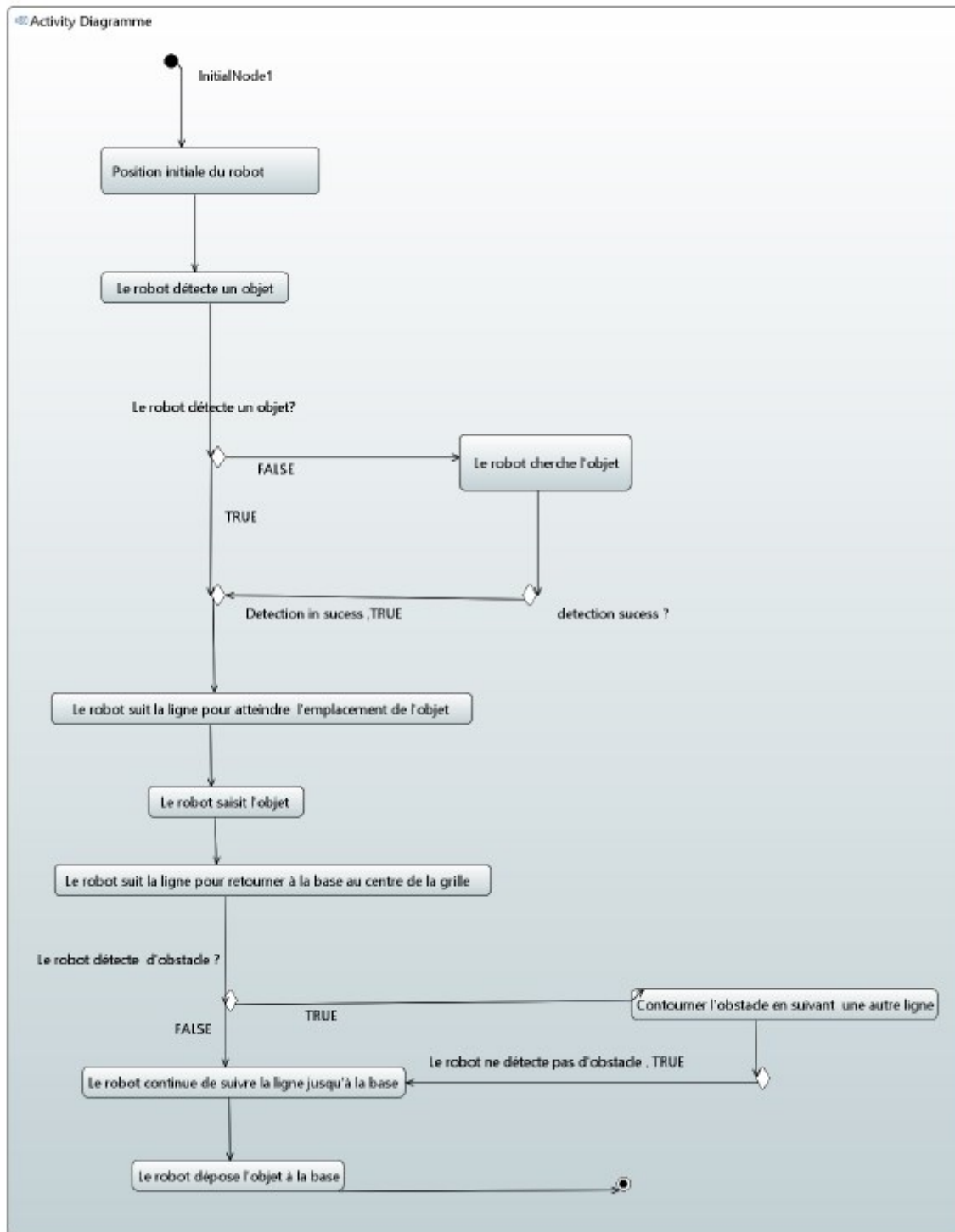


Figure 8. Activity Diagramme.

## 6. EXIGENCES NON FONCTIONNELLES

### 6.1. Lois, normes et standards influençant le produit

Au Maroc, il n'existe pas de lois spécifiques limitant la création et l'utilisation des robots. Cela est également le cas pour notre projet au sein de l'École Centrale Casablanca. Cependant, il est important



de souligner que notre robot doit respecter les règles éthiques qui représentent des normes sociales à ne pas négliger.

Bien que les réglementations juridiques spécifiques concernant les robots ne soient pas encore en place au Maroc, il est essentiel de prendre en compte les principes éthiques lors de la conception et de l'utilisation du robot. Cela implique de garantir la sécurité des utilisateurs, de respecter la vie privée, d'éviter toute discrimination et de prendre en compte les conséquences sociales et environnementales de son utilisation.

Ainsi, même en l'absence de normes juridiques spécifiques, il est crucial d'adopter une approche éthique dans le développement et l'utilisation des robots, en tenant compte des principes et des valeurs qui guident les interactions sociales et les responsabilités morales.

#### 6.1. Ergonomie et convivialité :

Dans les spécifications de notre robot, certaines exigences concernent son apparence et son impact sur le client.

##### 1. Apparence :

La taille du robot ne doit pas dépasser la taille du carré de départ de la grille. De plus, le robot doit être composé de capteurs et de bras.

##### 2. Ambiance :

Lors de l'exécution de sa mission de collecte d'objets, le robot ne doit pas générer de bruit excessif ni perturber son environnement.

#### 6.2. Facilité d'utilisation et facteurs humains :

L'assemblage du robot doit être facile pour permettre le remplacement des pièces et l'ajout de capteurs. Le code à compiler doit être compréhensible et clair.

#### 6.3. Performances et qualité du produit :

Le robot doit être capable de mener à bien ses missions, c'est-à-dire se déplacer sur le terrain de jeu, détecter les obstacles et les objets dispersés, puis les ramener au centre de la grille. La stabilité et la rapidité sont des critères importants qui influencent la qualité du produit.

#### 6.4. Adéquation du produit avec son environnement :

Le robot doit suivre la ligne noire et ne pas dépasser les limites de la grille. Il doit également faire la distinction entre les obstacles et les objets, et transporter les objets jusqu'à une base située au centre de la grille.

#### 6.5. Maintenance, support, portabilité, installation du produit :

Pour faciliter la maintenance, la plaque de contrôle doit être visible, facile à manipuler et accessible, sans être cachée par d'autres pièces ou fils. De plus, pour éviter tout désassemblage accidentel, les pièces doivent être solidement fixées. Un montage simple permettra de remplacer facilement des parties du système.

## 6.6. Sécurité :

Dans le cadre de notre projet, le robot ne présente aucun danger pour les utilisateurs. Cependant, si nous envisageons son utilisation dans des contextes agricoles, par exemple pour la récolte, il doit fonctionner sans nuire à l'opérateur.

## 6.7. Exigences culturelles et politiques :

Le robot doit respecter les normes culturelles et éthiques de la société marocaine, mais il n'existe pas de lois limitant son ouverture aux autres cultures et à leur diversité. Du point de vue politique, le robot que nous concevons n'est pas soumis à des standards politiques, car il s'agit à l'origine d'un projet visant à évaluer les compétences des élèves en modélisation, conception et robotique.

# 7. ANALYSE FONCTIONNELLE

## 7.1. Caractérisation des exigences

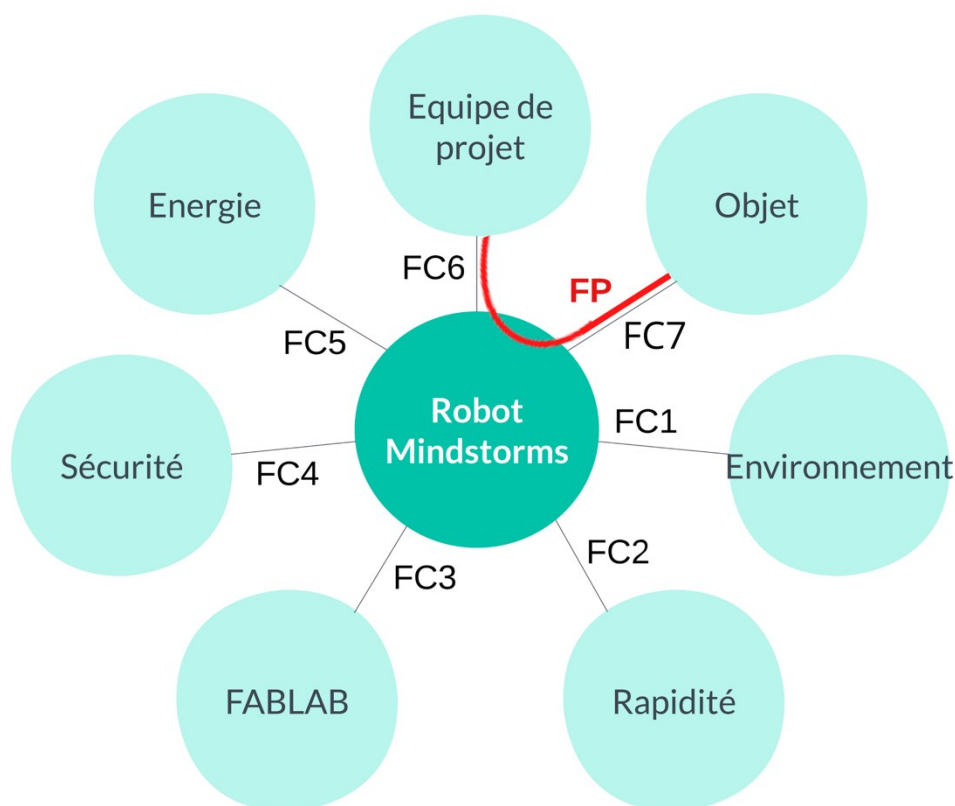


Figure 9. Analyse fonctionnelle externe par le diagramme de pieuvre.

Exigence	Critère	Niveau d 'exigence	Flexibilité
FP : doit permettre de détecter la ligne noire, détecter l'objet, puis le déplacer en parcourant tout le circuit	• Poids des objets manipulés	• 7 Kg	• $\pm 1$ g
FC1 : doit permettre de respecter les contraintes du terrain de jeu (limite extérieure, etc.)	Dimensions (Longueur, largeur, Hauteur) du Robot (doit tenir dans une case de la grille)	(274, 140, 72) mm	( $\pm 20$ , $\pm 10$ , $\pm 10$ ) mm
FC2 : la solution technique doit être rapide dans l'exécution des tâches pour permettre à l'équipe de gagner	• Plage de vitesse du robot • Plage d'accélération du robot	$V \hat{=} [0.4 ; 0.8] \text{ ms}^{-1}$ $A \hat{=} [0 ; 0.2] \text{ ms}^{-2}$	$\pm 0.1 \text{ ms}^{-1}$ $\pm 0.05 \text{ ms}^{-1}$
FC3 : doit se soumettre aux règles du Fablab	• Règles d'accès : il est interdit d'accéder au kit en dehors des heures programmées, sauf demande préalable et autorisation • 1 kit pour chaque groupe • 4 séances de TP pour chaque groupe		
FC4 : doit respecter les règles de sécurité	Eviter les collisions en introduisant un interrupteur de mouvement accessible par l'opérateur ou le jury		
FC5 : doit avoir une autonomie suffisante en énergie durant la compétition	Charger la batterie avant l'utilisation		
FC6 : doit être programmable par l'opérateur (équipe de projet)	Utiliser un EV3 pour coder		
FC7 : doit être adaptable aux dimensions de l'objet	Dimensions (Longueur, largeur, Hauteur) des Objets manipulés	• (7.8, 1,52, 6.72) mm	• ( $\pm 3$ , $\pm 5$ , $\pm 5$ ) mm

*Tableau 4. Caractérisation des exigences.*

## 7.2. Etude de l'existant

Une grande partie de la conception et la réalisation de notre robot était faite grâce au logiciel LEGO MINDSTORM EV3 qui nous permis d'avoir plusieurs solutions pour réaliser les fonctions principales.

Nous Avons tout fait par code python en usant de la bibliothèque de LEGO MINDSTORM. Les librairies offertes étaient suffisantes pour la réalisation de notre tâche.

## 7.3. Analyse de compromis

FP : Détection des objets

- Critère de comparaison : Capacité à détecter l'objet.

- Possibilités : Utiliser un radar pour détecter l'objet indépendamment de sa position, le détecter lorsqu'il est positionné sur la ligne noire, ou le détecter dans une position spécifique.
- Choix de réalisation : Utiliser un radar placé à l'avant du robot pour détecter l'objet.

#### FP : Déplacement sur le tapis

- Critère de comparaison : Capacité à se déplacer le long de la ligne noire.
- Possibilités : Utiliser un capteur de couleur pour se déplacer le long de la ligne noire et effectuer des oscillations pour corriger le déplacement.
- Choix de réalisation : Utiliser un capteur de couleur face au tapis et un suiveur de ligne qui tient compte des intensités de couleur du tapis.

#### FP : Déplacement de l'objet sur le tapis

- Critère de comparaison : Masse et dimensions des objets.
- Possibilités : Utiliser une pince pour déplacer les objets ayant une masse et des dimensions spécifiques de leurs positions initiales à leurs positions finales.
- Choix de réalisation : Utiliser une pince pour saisir l'objet après sa détection et le déplacer.

#### 7.4. Functional Analysis System Technique (FAST)

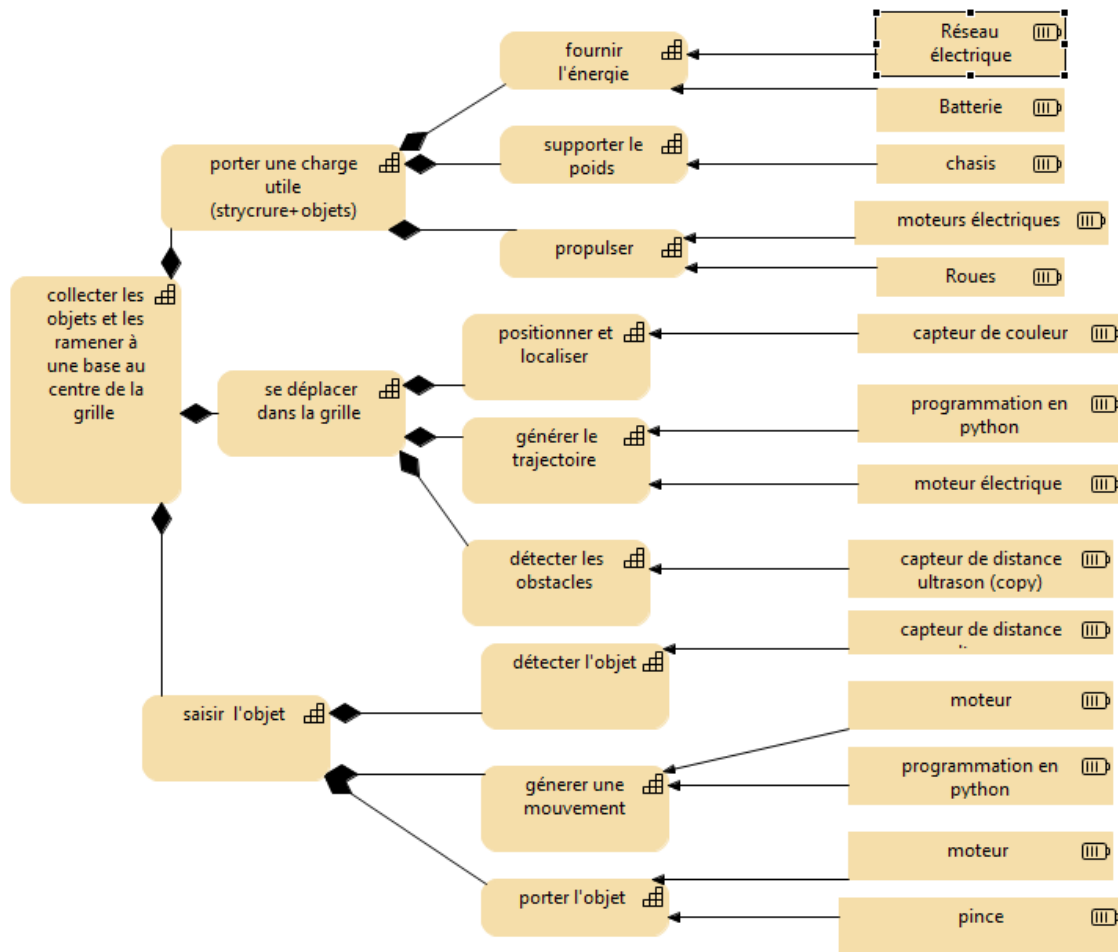
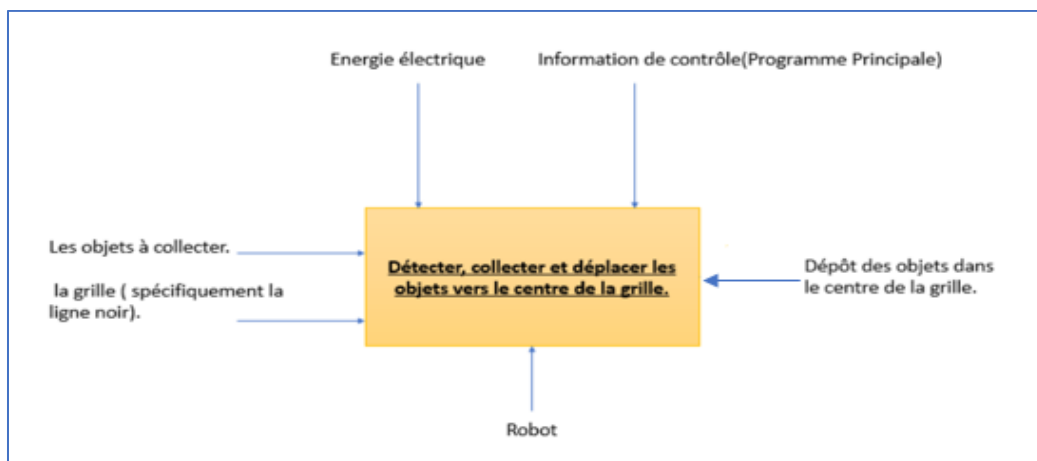


Figure 10. Diagramme FAST.

#### 7.5. Structured Analysis and Design Technique (SADT)



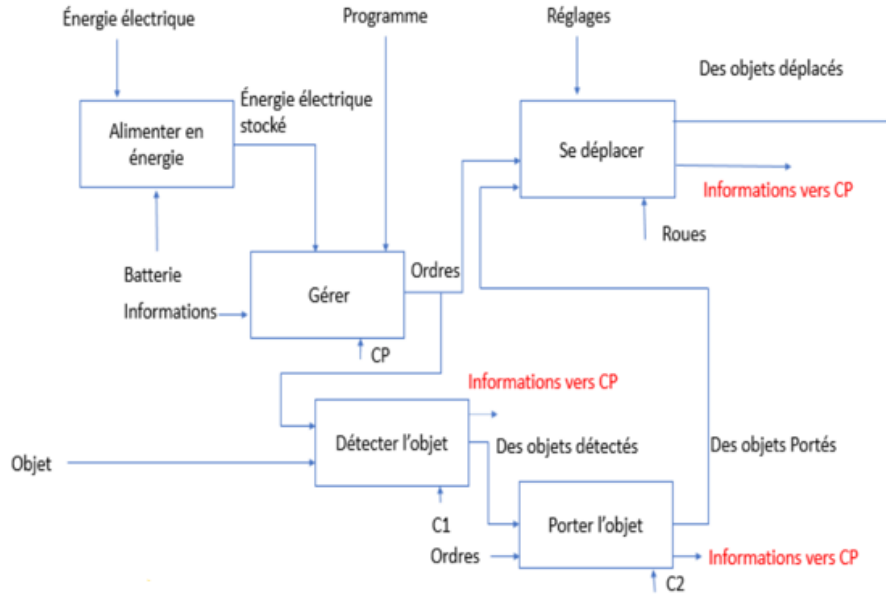
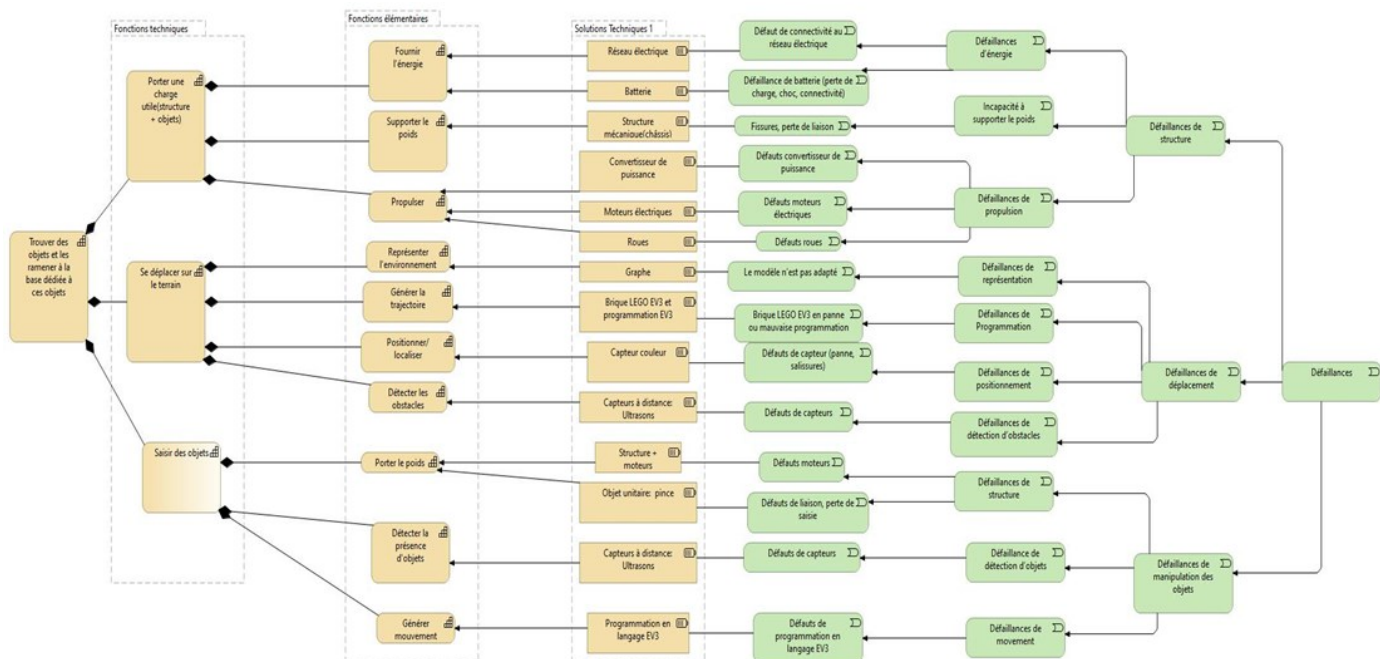


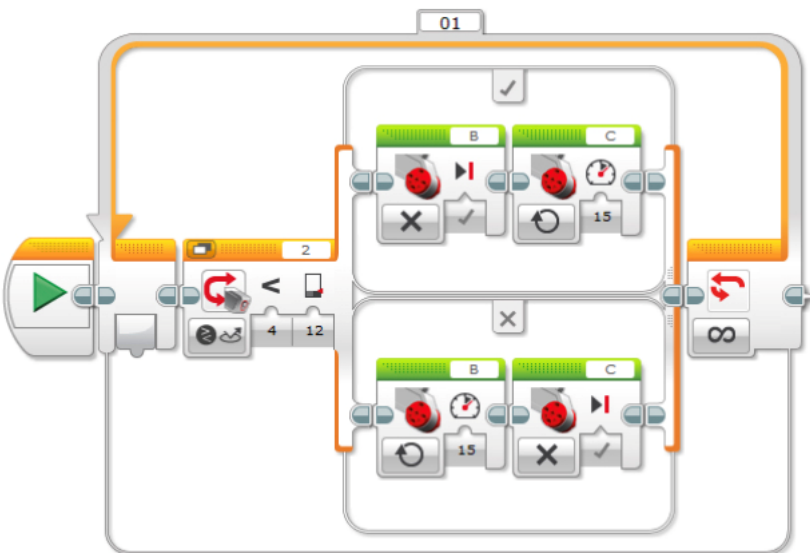
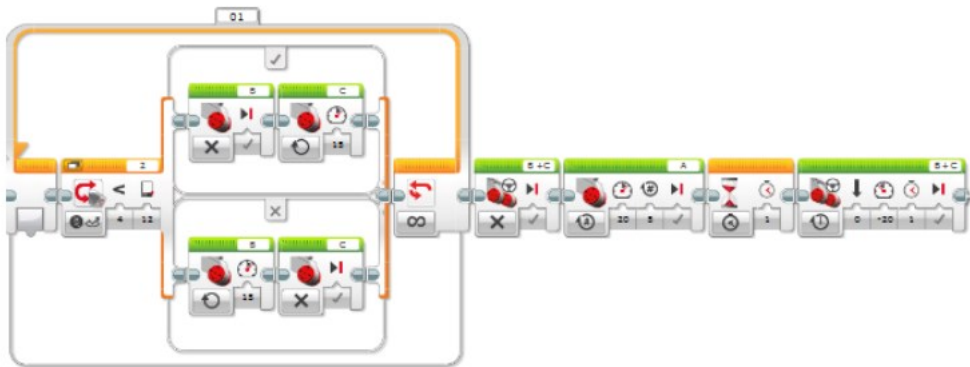
Figure 11. Diagramme SADT.

## 7.6. Analyse des modes de défaillance et de leurs effets



## 8. ARCHITECTURE LOGIQUE ET PHYSIQUE

Dans notre projet, nous avons réalisé le code d'un suiveur de ligne, puis celui d'un détecteur d'obstacles en finissant par un ramasseur d'objets et en voici la conception des différents codes :



## 8.1. Conception :

- Le processus de conception commence par l'assemblage des différentes parties du robot à l'aide du kit Lego, telles que les bras, le corps, etc.
- Il est important d'adopter une approche modulaire en développant des codes indépendants pour chaque fonction du robot, ce qui facilite la compréhension et la maintenance.
- Une fois les différentes parties assemblées, des tests sont effectués pour évaluer la puissance du robot.
- L'objectif est de trouver le bon équilibre entre la vitesse, la puissance et la précision du robot afin qu'il puisse accomplir les missions assignées avec succès.
- Des missions connexes sont également effectuées pour valider les capacités du robot.

## 8.2. Développement :

Le développement du robot se déroule en plusieurs étapes. Tout d'abord, un suiveur de ligne est programmé pour permettre au robot de suivre une ligne noire. Ensuite, la détection et la manipulation des objets sont implémentées pour que le robot puisse ramasser les objets dispersés sur son chemin. Enfin, le code est adapté pour répondre à différentes tâches spécifiques.

Les tâches 1, 2 et 3 consistent à permettre au robot de se déplacer le long de la ligne noire rectiligne, de détecter la présence d'objets et de les déplacer vers la bonne position dans la boucle.

Ci-après quelques extraits du code développé :

```
main.py
1  #!/usr/bin/env pybricks-micropython
2  from pybricks.hubs import EV3Brick
3  from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
4  | | | | InfraredSensor, UltrasonicSensor, GyroSensor)
5  from pybricks.parameters import Port, Stop, Direction, Button, Color
6  from pybricks.tools import wait, StopWatch, DataLog
7  from pybricks.robotics import DriveBase
8  from pybricks.media.ev3dev import SoundFile, ImageFile
9
10
11  # This program requires LEGO EV3 MicroPython v2.0 or higher.
12  # Click "Open user guide" on the EV3 extension tab for more information.
13
14
15  # Create your objects here.
16  left_motor = Motor(Port.B)
17  right_motor = Motor(Port.C)
18  gripper_motor = Motor(Port.D)
19
20  # Initialize the color sensor.
21  line_sensor = ColorSensor(Port.S3)
22  obstacle_sensor = UltrasonicSensor(Port.S4)
23
24  # Initialize the drive base.
25  robot = DriveBase(left_motor, right_motor, wheel_diameter=55.5, axle_track=104)
26
27  # Calculate the light threshold. Choose values based on your measurements.
28  BLACK = 9
29  WHITE = 85
30  threshold = (BLACK + WHITE) / 2
31
32  # Set the drive speed at 100 millimeters per second.
33  DRIVE_SPEED = 100
34
35
36  PROPORTIONAL_GAIN = 1.2
```

### Explication du code :

Nous avons utilisé le capteur de couleur pour mesurer l'intensité de la couleur blanche et de la couleur noire. Cette mesure est essentielle pour la programmation précise du robot afin qu'il puisse suivre la ligne entre les parties blanches et noires du parcours. Cette information permet de définir les seuils de couleur et d'adapter les actions du robot en conséquence pour maintenir une trajectoire précise le long de la ligne.



```

3
4 d=obstacle_sensor.distance()
5
6 while obstacle_sensor.distance() <= d:
7
8     d = obstacle_sensor.distance()
9     # Calculate the deviation from the threshold.
10    deviation = line_sensor.reflection() - threshold
11
12    # Calculate the turn rate.
13    turn_rate = PROPORTIONAL_GAIN * deviation
14
15    # Set the drive base speed and turn rate.
16    robot.drive(DRIVE_SPEED, turn_rate)
17
18    # You can wait for a short time or do other things in this loop.
19    wait(10)
20
21 while obstacle_sensor.distance() > 300:
22     while obstacle_sensor.distance() <= d:
23
24         d = obstacle_sensor.distance()
25         # Calculate the deviation from the threshold.
26         deviation = line_sensor.reflection() <= threshold
27         # Calculate the turn rate.
28         turn_rate = PROPORTIONAL_GAIN * deviation
29
30         # Set the drive base speed and turn rate.
31         robot.drive(DRIVE_SPEED, turn_rate)
32
33         # You can wait for a short time or do other things in this loop.
34         wait(10)
35
36 while obstacle_sensor.distance() > 300:
37     robot.drive(200,0)
38

```

### Explication du code :

Une fois le robot détecte la présence d'un objet, il enclenche une séquence d'actions préétablies. Initialement, il se déplace en suivant la ligne noire et s'arrête à une distance de 1 cm de l'objet repéré. Ensuite, le bras du robot entre en action. Grâce à la fonction préprogrammée "gripper motor.rununtil stalled (-200)", le moteur moyen descend le bras pour saisir solidement l'objet identifié. Cette fonction permet de contrôler le mouvement du moteur jusqu'à ce qu'il rencontre une résistance, garantissant ainsi que le bras atteigne la position souhaitée pour la collecte de l'objet.

Une fois que l'objet est solidement saisi, le robot utilise une autre fonction préprogrammée pour remonter le bras et libérer l'objet collecté. Cette fois, la fonction utilisée est "gripper motor.rununtilstalled (200)", qui ordonne au moteur moyen de faire monter le bras jusqu'à ce qu'il rencontre une résistance.

Enfin, pour poursuivre son trajet, le robot doit effectuer une rotation de 90° tout en suivant la ligne noire. Cela est réalisé à l'aide de la fonction préétablie "robot.turn(90)". Cette fonction permet de commander précisément au robot de réaliser une rotation de 90° dans la direction souhaitée, tout en maintenant son trajet le long de la ligne noire.

### 8.3. Tests et validation :

Après avoir réalisé plusieurs essais et améliorations pour optimiser les performances du robot, des tests sont effectués pour valider son fonctionnement et sa capacité à accomplir les étapes requises. Voici un résumé des tests réalisés :

- Test du fonctionnement du capteur à ultrasons : Il a été observé que parfois la pince du robot ne saisit pas correctement l'objet en raison de l'incertitude liée à la position angulaire du robot après avoir suivi la ligne. Bien que cette imprécision soit considérée comme négligeable, elle peut être améliorée ultérieurement.

- Test du suiveur de ligne : Le robot peut rencontrer des difficultés à distinguer le blanc sur le côté d'une ligne droite du blanc à la fin d'un virage. Ce problème peut être adressé dans les futures itérations du développement.

## Partie QHSE :

Le robot automatisé que nous avons développé, capable de suivre une ligne, de détecter les obstacles et de ramasser des objets, a le potentiel d'être utilisé dans divers domaines tels que la restauration et les services commerciaux.

En ce qui concerne les aspects de qualité, d'hygiène, de sécurité et d'environnement (QHSE) qui doivent être pris en compte :

- Qualité : Il est important que le robot puisse suivre la ligne noire de manière fluide et atteindre sa destination de manière fiable. Cela implique d'appliquer des pratiques de gestion de la qualité continues, telles que l'approche KAIZEN, pour optimiser les performances du robot.

- Hygiène : Les normes d'hygiène et de santé/sécurité de référence, comme la norme ISO 45001, doivent être prises en compte pour garantir la sécurité des opérations liées au robot.

- Sécurité : Les principales exigences de sécurité sont liées à la prévention des dysfonctionnements du robot afin d'éviter les accidents de travail. Des mesures appropriées doivent être prises pour garantir la sécurité des utilisateurs et des opérateurs.

- Environnement : Le robot doit être conçu de manière à réduire sa consommation d'énergie et à respecter les normes environnementales. L'application d'un système de management environnemental (SME) conforme aux normes ISO 14001 et ISO 14004 est recommandée.

## CONCLUSION ET PERPESCTIVES

En conclusion, ce projet nous a permis d'appliquer les concepts du génie industriel dans la conception et le développement d'un robot suiveur de ligne. Bien que le robot puisse accomplir les trois premières missions, il présente des limitations en termes d'adaptabilité à de nouvelles missions ou conditions sans une reprogrammation avancée. Des améliorations sont nécessaires pour résoudre les problèmes de détection des obstacles et pour augmenter la vitesse de déplacement du robot. La prise en compte des aspects QHSE est essentielle pour garantir la qualité, l'hygiène, la sécurité et l'impact environnemental du robot. Ce robot n'est qu'une introduction au réel problème qui se pose aujourd'hui dans les usines et des grandes firmes comme Amazon qui utilise aujourd'hui des robots pour déplacer des objets au lieu d'employer des personnes. Des milliers de robots sont mis en marche donc il faut créer des chemins pour ne pas qu'ils se croisent et effectuent le travail de manière optimale en parcourant le moins de distance possible car il se pose le problème est celui de l'énergie dépensé. Ce problème tout

autant le propriétaire de ces machines que nous, élèves ingénieurs centraliens qui est d'un point de vue mathématique résoudre cette équation et d'une autre part d'un point de vue environnemental sauvegarder le plus d'énergie possible.

## 9. BIBLIOGRAPHIE – WEBOGRAPHIE

-Tutoriel-Robot de base-ev3-rem-driving-base

-[MINDSTORMS EV3 Support | Everything You Need | LEGO® Education](#)

-[Exploration de Mars par Curiosity — Wikipédia \(wikipedia.org\)](#)

-Lien :

[https://centrale-](https://centrale-casablanca.edunao.com/pluginfile.php/4452/mod_folder/content/0/Tutoriel%20-%20Robot%20de%20base%20-%20ev3-rem-driving-base.pdf?forcedownload=1)

[casablanca.edunao.com/pluginfile.php/4452/mod\\_folder/content/0/Tutoriel%20-%20Robot%20de%20base%20-%20ev3-rem-driving-base.pdf?forcedownload=1](https://centrale-casablanca.edunao.com/pluginfile.php/4452/mod_folder/content/0/Tutoriel%20-%20Robot%20de%20base%20-%20ev3-rem-driving-base.pdf?forcedownload=1)

-Robertson, S., & Robertson, J. (2013). *Mastering the requirements process* (3rd Ed). Addison-Wesley.