

UNIVERSITÉ PAUL SABATIER

Conception et Intégration des systèmes critiques

SysML - Robot de manutention

Étudiants :

Khalil BEN GHORBEL

Camille CEULENEER

Rafik CHOUCANE

Houssayen SOUSSI

Pour :

Vincent ALBERT

Table des matières

Introduction	1
1 Description du projet	1
1.1 Mission et objectifs du système	1
1.2 Contexte du système	2
1.3 Concepts du système	4
1.4 Objectifs du système	4
1.5 Les exigences du système	6
2 Travail réalisé	7
2.1 Conception générale du système	7
2.1.1 Sous-système Déplacement	7
2.1.2 Sous-système Préhension	8
2.1.3 Sous-système Diagnostique	8
2.2 Conception détaillée du système	8
2.2.1 Gestion de l'énergie	9
2.2.2 Préhension	10
2.2.3 Déplacement	10
2.2.4 Diagnostique	11
2.3 Diagrammes de séquence du système	11
2.3.1 Gérer les Obstacles	11
2.3.2 Tester	12
2.3.3 Se Déplacer	13
2.3.4 Stocker	13
2.3.5 Tester et Régler	14
2.3.6 Suivi de ligne	15
2.4 Matrice Opérations-Exigences	15
Conclusion	16
A Annexes	17
A.1 Batterie LiPo	17
A.2 Moteurs D3542	18
A.3 DRV10975	19
A.4 Capteur de couleurs	20
A.5 HC-SR04	21

Introduction

Dans le cadre de notre cours de SysML, nous avons été amenés à travailler sur un projet de conception système. Un cahier des charges ainsi qu'une analyse des exigences nous ont été fournis en SysML (réalisé avec le logiciel *MagicDraw*).

Notre travail a été de réaliser la conception du système en utilisant le logiciel de conception *MagicDraw*.

1 Description du projet

1.1 Mission et objectifs du système

Le robot de manutention est destiné à assurer le transport de différents produits à l'intérieur d'un atelier en suivant les programmes de production spécifiés.

Voici une représentation de cet atelier industriel :

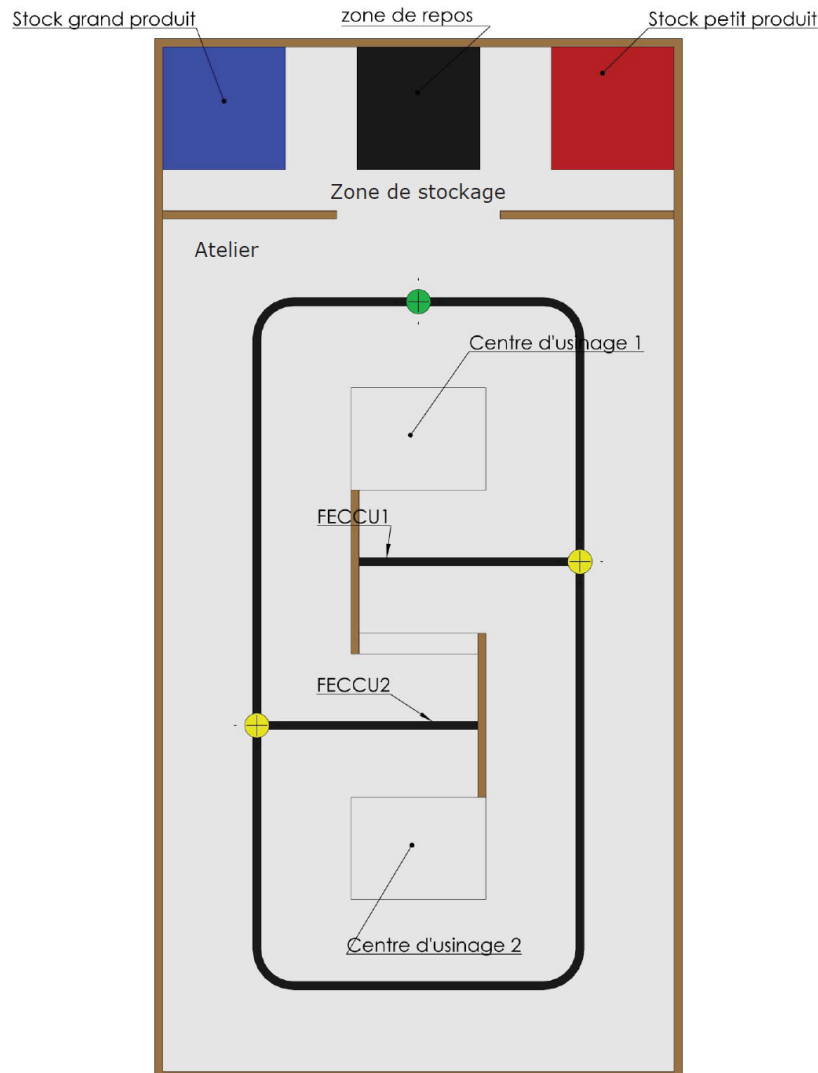


FIGURE 1: Zone d'activité du robot

Les produits, situés dans des centres d'usinage, doivent être saisis et transportés automatiquement par le robot. Ces produits doivent ensuite être déposés dans une des deux zones de stockage de l'atelier. Le choix de la zone de stockage est défini par la taille du produit. Le robot dispose d'une zone de repos, sur laquelle il doit se positionner lorsqu'il est inactif.

1.2 Contexte du système

Le robot de manutention est au centre d'un environnement spécifique connu. Notre système est alors en interaction avec sept éléments principaux :

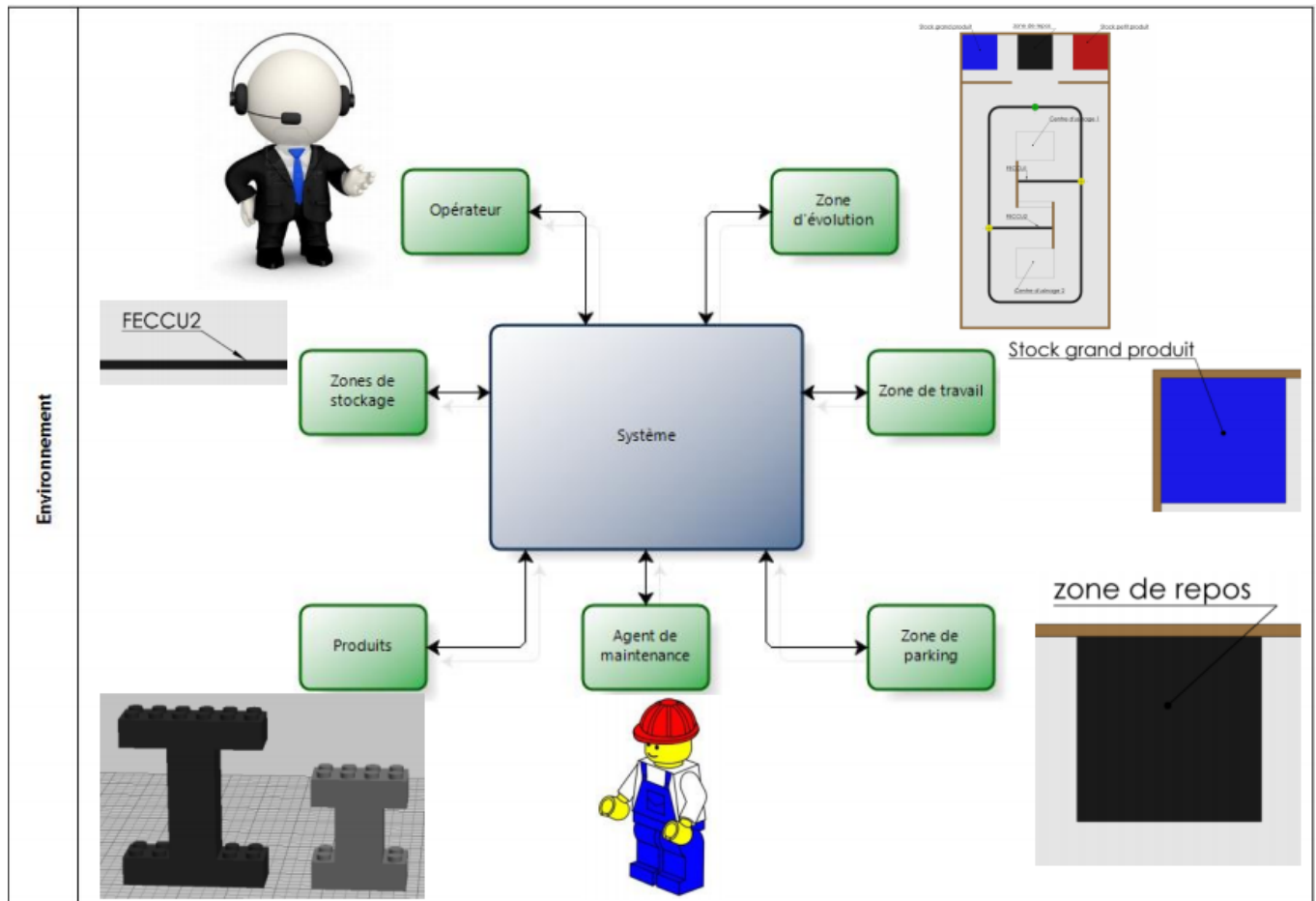


FIGURE 2: Environnement du robot

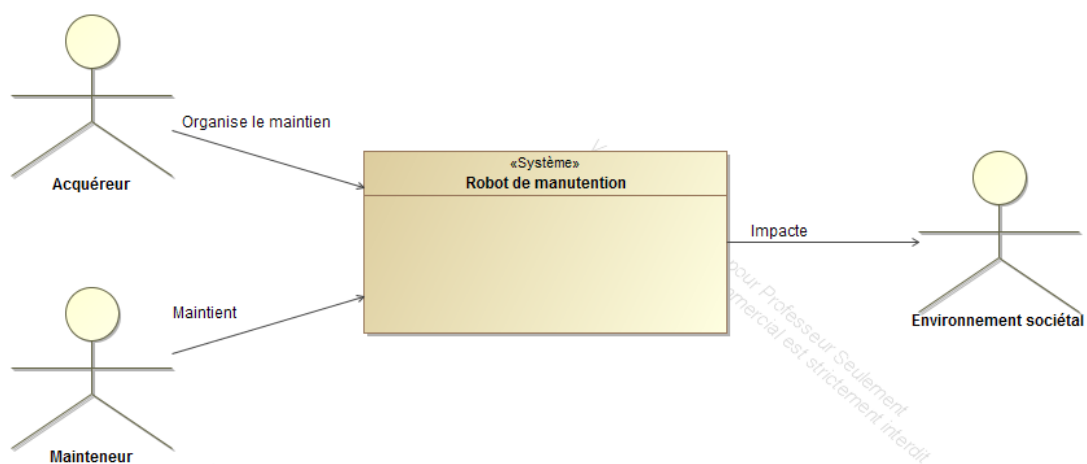


FIGURE 3: Diagramme de contexte en Maintient Opérationnel

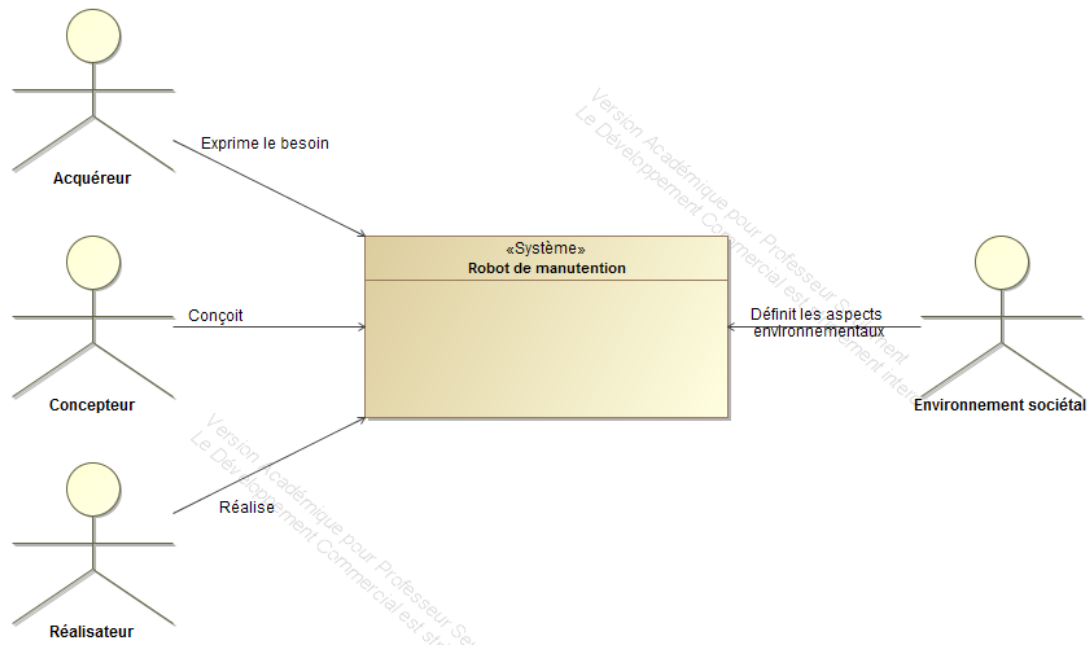


FIGURE 4: Diagramme de contexte en Conception et Réalisation

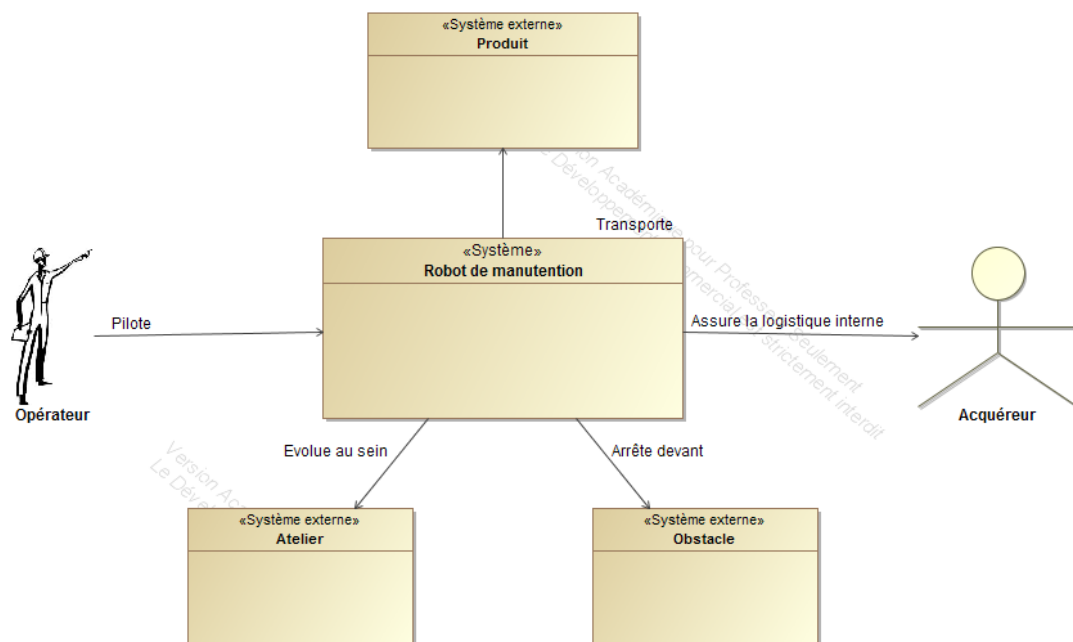


FIGURE 5: Diagramme de contexte en Exploitation

1.3 Concepts du système

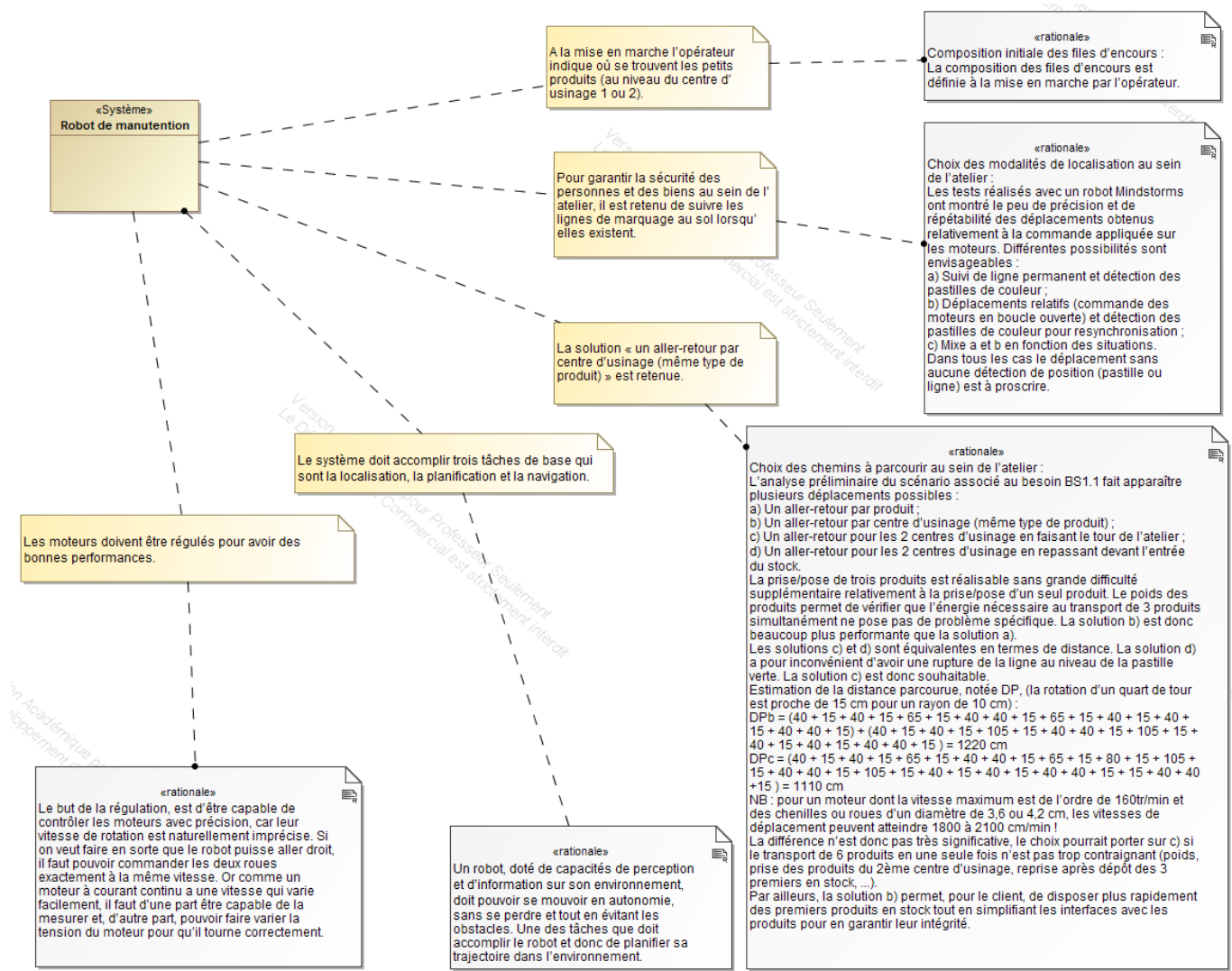


FIGURE 6: Diagramme de Concepts du système

1.4 Objectifs du système

Initialement, le robot est en état de veille jusqu'à ce qu'une demande de test ou une demande de départ ait lieu. Si le robot reçoit une demande de test, il passe à l'état "Test en cours" et revient à l'état "En veille" dès que le test est terminé. S'il reçoit une demande de départ, il passe au macro-état "Stockage de produits en cours". Cet état est lui même composé de 3 sous états. Le robot est initialement à l'état "Mise en stock des produits en cours". Cet état peut être interrompu soit par une détection d'obstacle qui mène à l'état "Mission interrompue - obstacle en cours de détection" duquel on retourne à l'état précédent lorsque l'obstacle est évacué ; soit par une demande de maintenance curative qui mène à l'état "Mission interrompue - maintenance curative en cours" qui une fois terminée fait retourner le robot à l'état "Mise en stock des produits en cours".

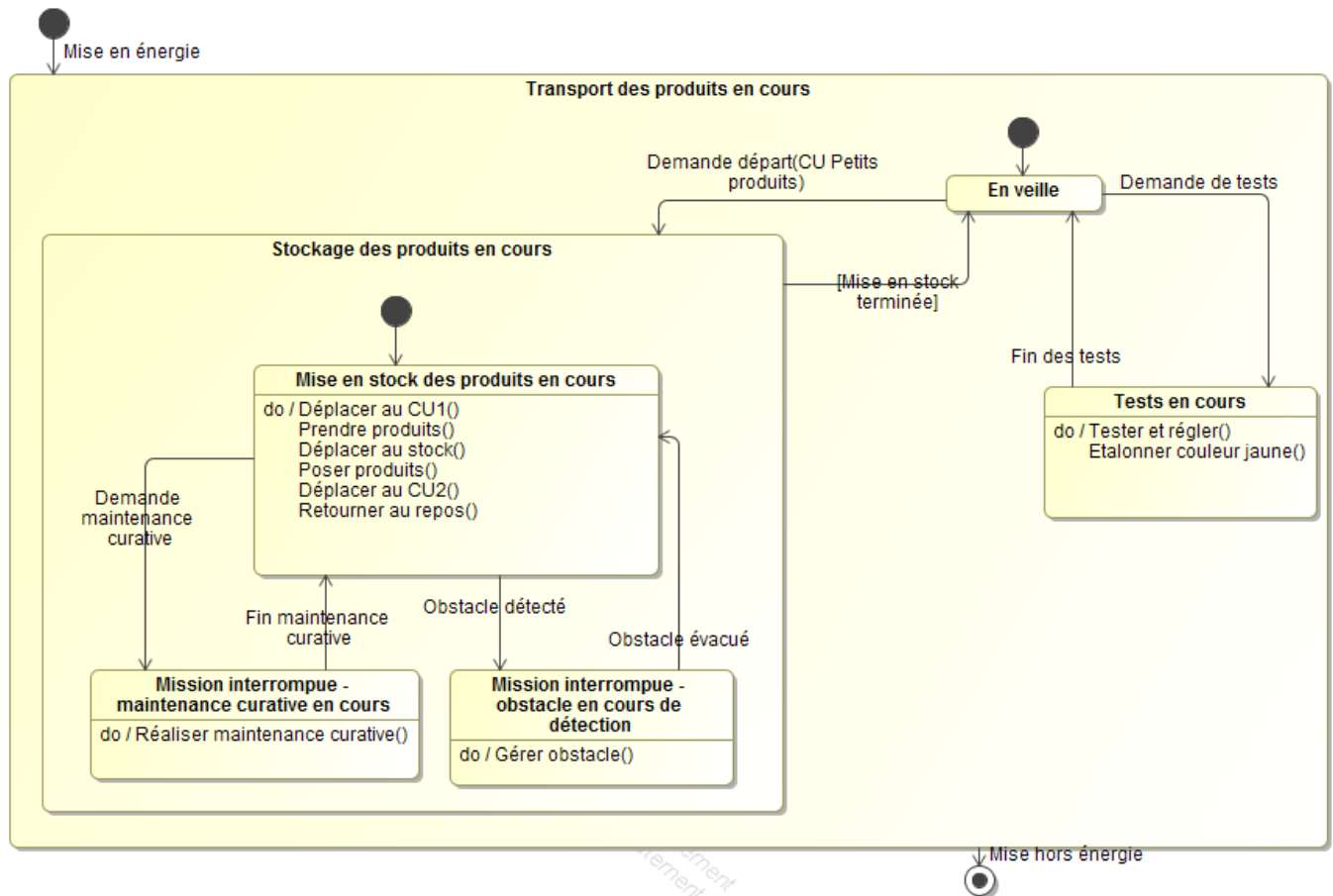


FIGURE 7: Diagramme d'état du robot en exploitation

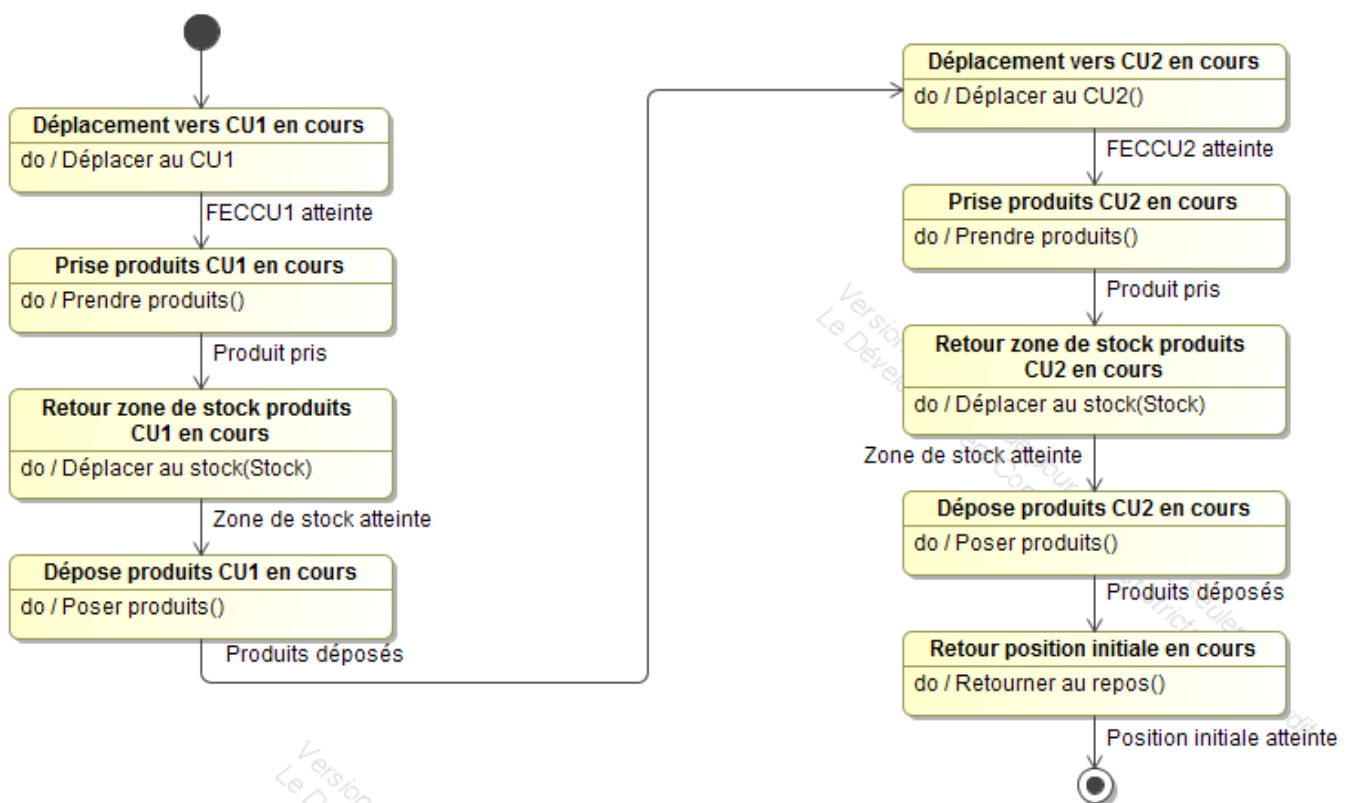


FIGURE 8: Diagramme du Cas d'Utilisation "Mettre en stock"

1.5 Les exigences du système

Voici la liste des exigences du système tel qu'elles nous ont été fournies au début du projet :

Valeur	Nom	Nom	Risk
1	EV2	<input type="checkbox"/> Réaliser des essais en faisant varier les conditions d'éclairage	Réaliser des essais pour valider la détection des couleurs dans différentes situations d'éclairage (Luminosité, reflets, ombres, ...).
2	EV1	<input type="checkbox"/> Réaliser au moins 6 missions élémentaires successives	La répétabilité doit être vérifiée pour garantir le bon déroulement des missions sur une durée de 8 heures.
3	EP1	<input type="checkbox"/> Réaliser une mission élémentaire en moins de 4 minutes	Le robot dispose de moins de 4 minutes pour réaliser une mission élémentaire sans rencontrer d'obstacle.
4	EO7.2	<input type="checkbox"/> Repositionner et redémarrer le robot à l'endroit de l'incident	Le repositionnement et le redémarrage du robot à l'intérieur de la zone d'évolution à l'endroit correspondant à l'apparition de l'incident sont alors réalisés par ce même opérateur.
5	EO7.1	<input type="checkbox"/> Mettre en mode maintenance curative par un opérateur	Lors de situation de blocage, un opérateur intervient pour passer en mode maintenance curative (Précision de l'acquéreur).
6	EO6.2	<input type="checkbox"/> Repartir dès l'obstacle évacué	Le robot repart lorsque l'obstacle est évacué.
7	EO6.1	<input type="checkbox"/> Arrêter devant un obstacle	Le robot s'arrête devant un obstacle.
8	EO5	<input type="checkbox"/> Transporter les produits d'un CU sans contact avec le sol	Le robot transporte les produits issus d'un centre d'usinage sans contact avec le sol.
9	EO5.1	<input type="checkbox"/> Porter indifféremment 3 petits ou grands produits	Le robot porte sans contact avec le sol de 1 à 3 petits ou grands produits suivant le déroulement de la mission élémentaire.
10	EO4	<input type="checkbox"/> Réaliser la mission élémentaire en autonomie	Le robot réalise sa mission en toute autonomie (sans téléguidage).
11	EO3	<input type="checkbox"/> Suivre les lignes des allées de circulation	Le robot emprunte des allées de circulation en suivant les lignes lorsqu'elles existent.
12	EO2	<input type="checkbox"/> Suivre les lignes pour éviter les zones des CU	Le robot suit les lignes pour éviter des zones des CU.
13	EO1	<input type="checkbox"/> Mettre en marche par un opérateur en situant les petits produits	Le robot est mis en marche par un opérateur en indiquant où sont les petits produits (CU1 ou 2).
14	EI5	<input type="checkbox"/> Disposer d'une ergonomie adaptée pour la maintenance curative	L'opérateur doit disposer d'une ergonomie simple d'utilisation pour rentrer ou sortir du mode maintenance curative.
15	EI4	<input type="checkbox"/> Détecter l'obstacle à 10cm	La distance de détection est de 10 cm.
16	EI3	<input type="checkbox"/> Adapter le préhenseur aux produits	Le système de préhension des produits garantit leur préservation.
17	EI2	<input type="checkbox"/> Détecter les couleurs au sol	Les différentes couleurs des pastilles, des zones de stockage et des lignes sont détectées.
18	EI1	<input type="checkbox"/> Disposer d'une ergonomie adaptée pour le mise en marche	L'opérateur met en marche en indiquant où sont les petits produits simplement.
19	EF1	<input type="checkbox"/> Transporter les produits de l'atelier vers leur zone de stockage re...	La mission du robot consiste à transporter des produits dans un atelier (défini schématiquement en annexe 2), depuis deux files d'encours, en sorte respectivement de chaque centre d'usinage (CU1, CU2), pour les déposer dans une zone de stockage où les produits seront triés selon leur taille.
20	EF1.2	<input type="checkbox"/> Tester et régler	Ce mode permet de tester et régler les fonctionnalités élémentaires du système (déplacement, prise/pose des produits, localisation dans l'atelier, ...) ou plus simplement de tester et régler les capteurs et les actionneurs.
21	EF1.1	<input type="checkbox"/> Mettre en stock les produits présents sur les centres d'usinage	Ceci caractérise ce qui sera appelé "mission élémentaire". Initialement le robot est en position veille, sur sa zone de repos, à l'intérieur de la zone de stockage. Il y a au maximum 3 produits à transporter depuis chaque centre d'usinage. La composition des files d'encours est connue avant le début de la mission élémentaire. Chacune de ces files contient soit uniquement des petits produits, soit uniquement des grands produits.
22	EF1.1.1	<input type="checkbox"/> Gérer obstacle	Si durant ses déplacements dans l'atelier, le robot rencontre un obstacle le gênant dans son parcours, il devra alors s'arrêter et attendre que l'obstacle soit évacué avant la poursuite de la mission.
23	EF1.1.2	<input type="checkbox"/> Réaliser une maintenance curative	Si le robot est dans une situation de blocage (arrêt, détérioré), un opérateur peut réaliser des opérations de maintenance curative.
24	EF0	<input type="checkbox"/> Améliorer la productivité de la logistique interne	L'automatisation des flux internes de produits a été décidée pour améliorer la productivité.
25	EC9	<input type="checkbox"/> Être testable et maintenable	Le robot est testable et maintenable.
26	EC9.2	<input type="checkbox"/> Disposer d'un dossier technique	Le service maintenance doit avoir accès au dossier technique complet du système. (Besoin de l'acquéreur)
27	EC9.1	<input type="checkbox"/> Faciliter les actions de maintenance	Faire en sorte que les interventions de maintenance soit faciles à réaliser, comme par exemple l'accès à l'alimentation. (Besoin du mainteneur)
28	EC8	<input type="checkbox"/> Assembler et configurer en moins de 20 minutes	Le robot est assemblé et configuré en moins de 20 minutes.
29	EC7	<input type="checkbox"/> Prendre et poser les produits verticalement dans les zones	Les produits sont posés verticalement dans les zones.
30	EC6	<input type="checkbox"/> Opérer avec les produits spécifiés	La définition des produits est donnée en annexe 1.
31	EC5	<input type="checkbox"/> Opérer sans maintenance périodique au moins 6 missions success...	Le robot peut opérer sans maintenance périodique au moins 6 missions successives sur une journée de 8 heures.
32	EC4	<input type="checkbox"/> Dimensionner le robot au repos dans un cube de 30 cm d'arrête	Les dimensions du robot au repos sont comprises dans un cube de 30 x 30 x 30 cm.
33	EC3	<input type="checkbox"/> Assembler avec les composants des kits LEGO	Le robot est assemblé à partir des composants du kit LEGO® Mindstorms NXT 2.0 et l'aménagement des zones d'encours utilise seulement des pièces présentes dans le kit LEGO® 5549.
34	EC3.1	<input type="checkbox"/> Utiliser des piles rechargeables	Pour minimiser les coûts et l'impact environnemental utiliser des piles rechargeables (Besoin de l'acquéreur et de l'environnement sociétal).
35	EC2	<input type="checkbox"/> Utiliser un robot mobile	Il est demandé d'automatiser les flux internes en utilisant un robot mobile de manutention.
36	EC11	<input type="checkbox"/> Régler les paramètres de détection	Les paramètres de détection permettent de tenir compte des conditions de l'environnement (éclairages, reflets, ...).
37	EC10	<input type="checkbox"/> Opérer dans l'atelier et la zone de stockage spécifiés	Les dimensions et couleurs de l'atelier et de la zone de stockage sont spécifiées sur le schéma de l'annexe 2.
38	EC1	<input type="checkbox"/> Evoluer dans un environnement industriel simulé	Le contexte est l'automatisation de la logistique interne d'un environnement industriel simulé. Celui-ci est constitué d'un atelier et d'une zone de stockage attenante.
39	EC1.1	<input type="checkbox"/> Opérer dans des conditions climatiques déterminées	Les conditions climatiques sont : 15°C <= température <= 32°C, 1000mb <= pression <= 1030mb, 40% <= hygrométrie <= 75% (Précision de l'acquéreur).

FIGURE 9: Table des exigences

2 Travail réalisé

2.1 Conception générale du système

Nous avons commencé par réaliser la conception générale du robot dans une approche top-down. Voici le diagramme de conception générale du robot :

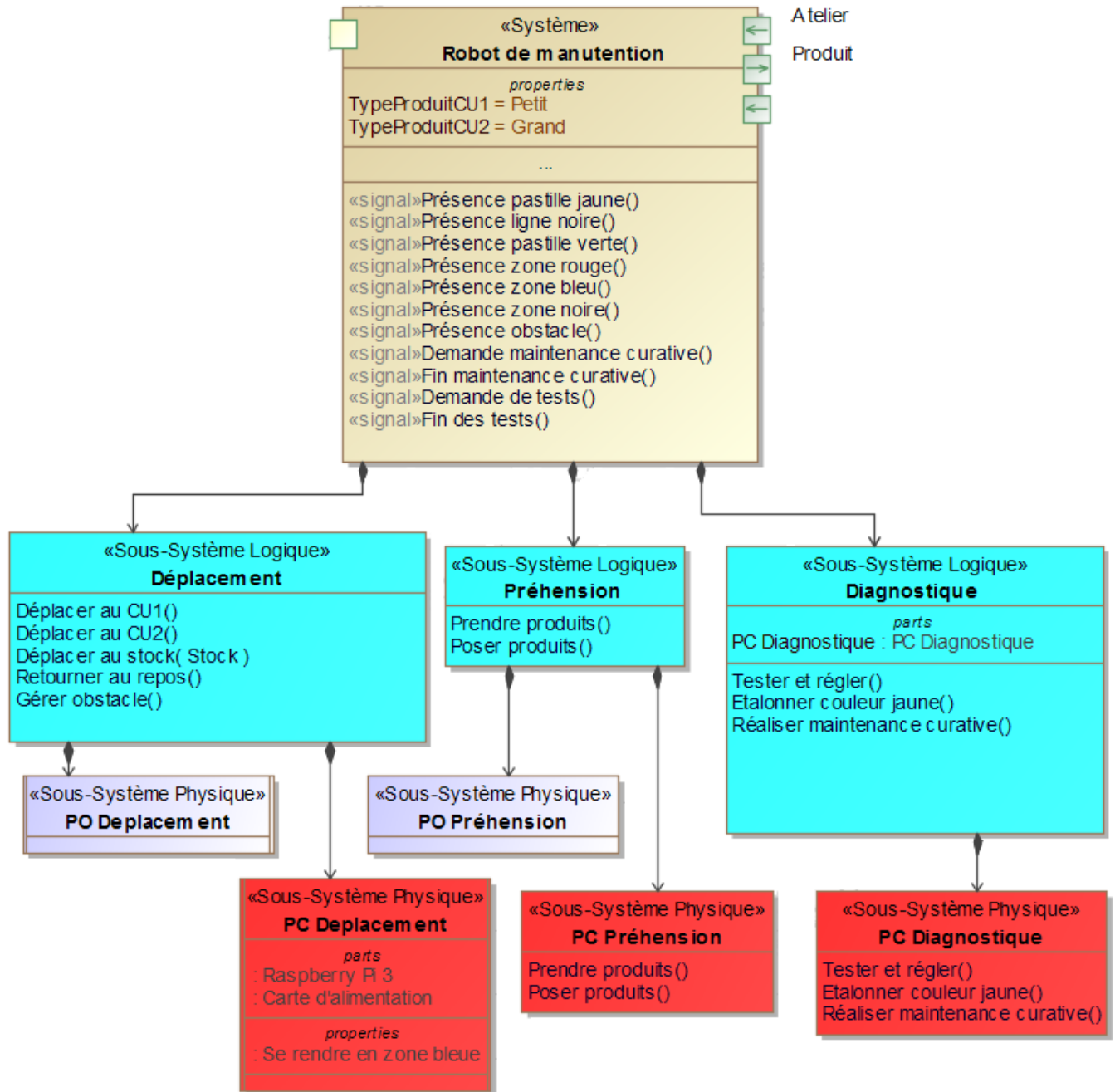


FIGURE 10: Conception générale

Nous l'avons élaboré à partir de la définition des besoins et de l'analyse des exigences. Le robot est composé de trois sous-systèmes : *Déplacement*, *Préhension* et *Diagnostic*.

2.1.1 Sous-système Déplacement

Le sous-système *Déplacement* gère le déplacement du robot dans toute sa zone d'activité.

Il doit gérer les déplacements dans les différentes zones (Stockages, Repos, Centre d'Usinage 1 et Centre d'Usinage 2) ainsi que l'évitement des éventuels obstacles.

2.1.2 Sous-système Préhension

Le sous-système *Préhension* gère la prise et la pose des différents produits.

2.1.3 Sous-système Diagnostique

Le sous-système *Diagnostique* gère les différentes tâches de maintenance du robot.

Il doit permettre d'effectuer des réglages sur le robot, d'étalonner les différents capteurs et de faire de la maintenance curative.

2.2 Conception détaillée du système

Après avoir fait des choix de conception générale, nous avons développé les sous-systèmes du robot dans une conception détaillée. Voici le diagramme de conception détaillée du robot :

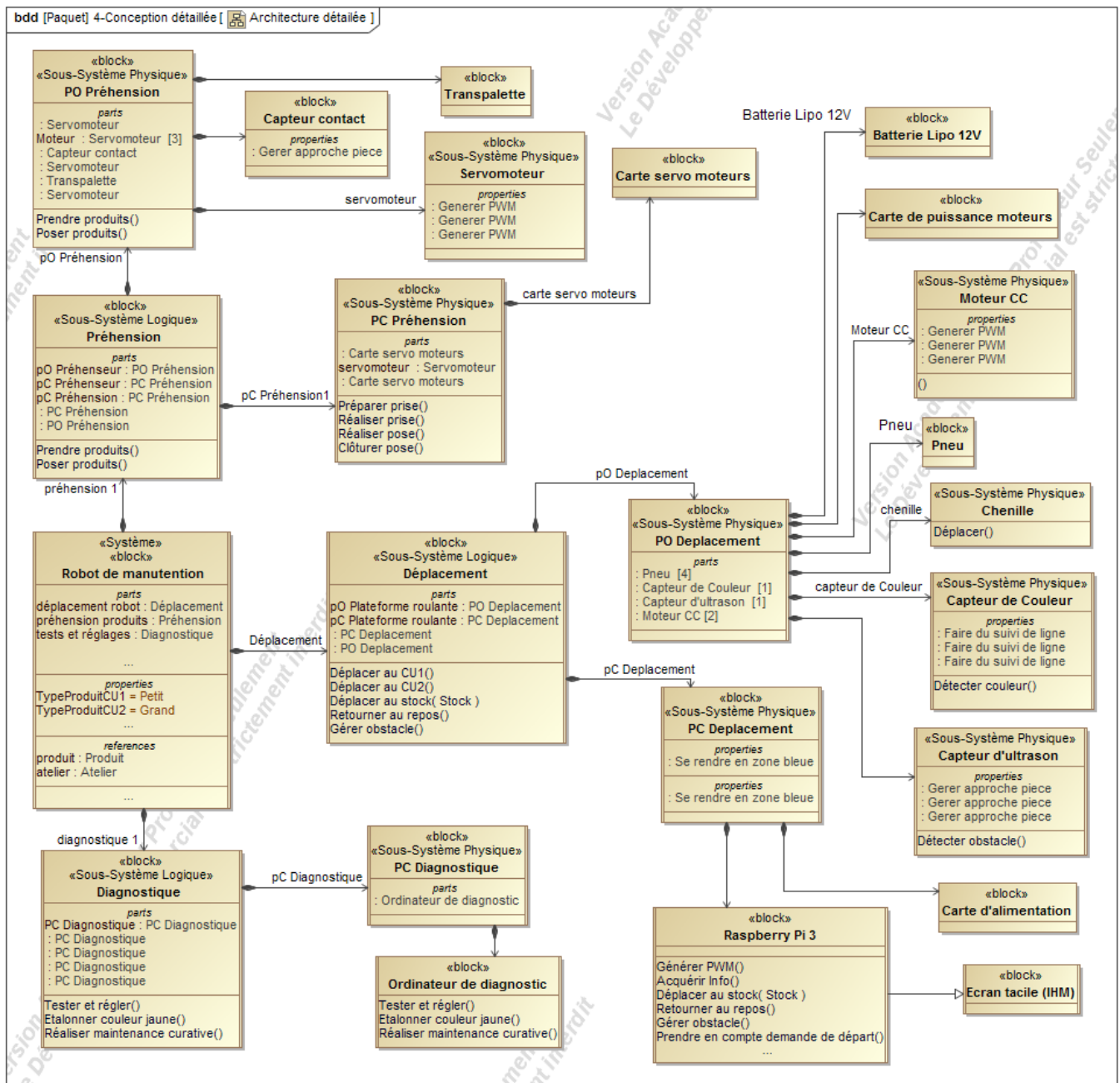


FIGURE 11: Conception détaillée

Et voici le diagramme de bloc interne du robot :

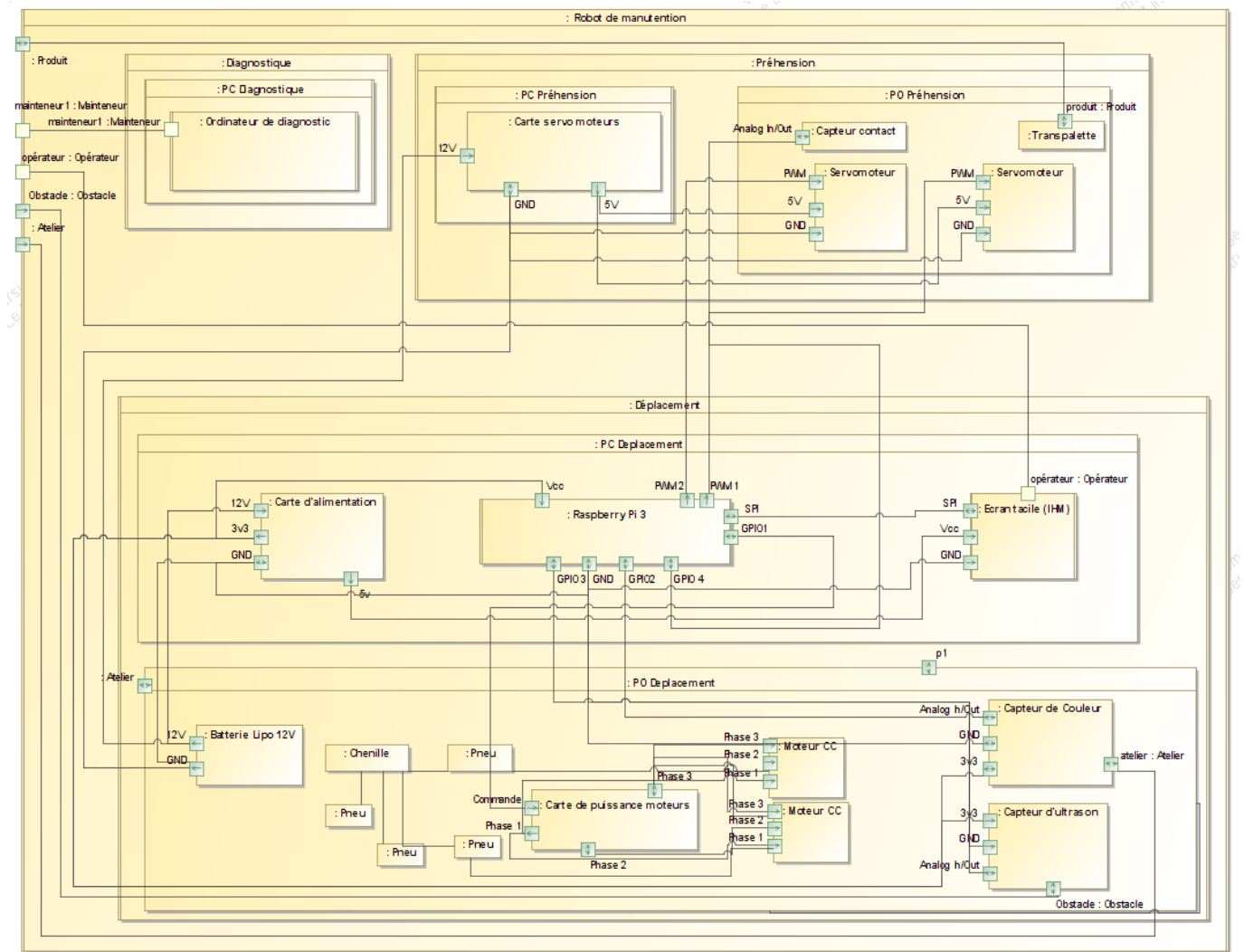


FIGURE 12: Diagramme de bloc interne du robot

Nous allons d'abord décrire cette architecture de façon globale avant de détailler chaque élément la constituant.

L'architecture du robot est organisée autour d'un ordinateur unique : un Raspberry Pi 3. Chaque élément du système y est relié à l'aide de ports d'entrées/sorties. L'alimentation générale du robot est assurée par une batterie Lithium Polymère (LiPo) (composée de plusieurs cellules, documentation en annexe A.1) délivrant une tension de 11,1 volts.

2.2.1 Gestion de l'énergie

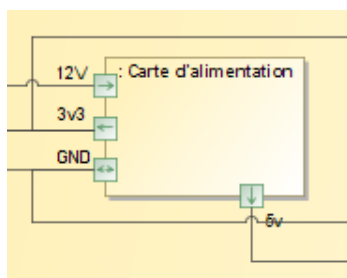


FIGURE 13: Carte d'alimentation

Une carte d'alimentation permet d'adapter la tension fournie par la batterie LiPo pour permettre l'alimentation des tous les éléments constituant le robot. Cette carte permet d'abaisser la tension de la batterie de 12 volts à 5 volts et 3,3 volts, permettant ainsi le fonctionnement des autres sous-systèmes du robot.

2.2.2 Préhension

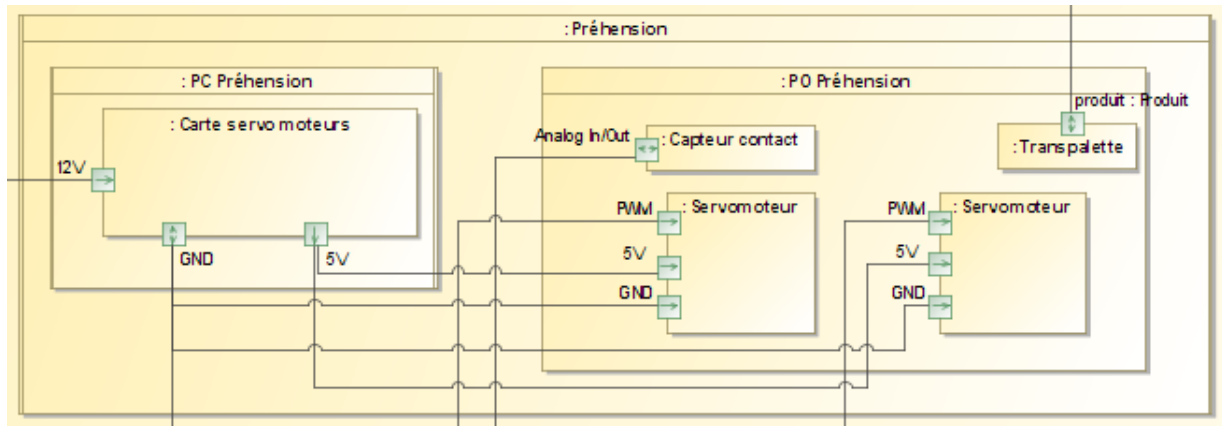


FIGURE 14: Préhension

Le sous-système *Préhension* est composé de capteurs et d'actionneurs. Les capteurs permettent au robot de s'assurer que les produits sont bien placés sur le transpalette. Des servomoteurs permettent de prendre et de poser les produits. Ils sont alimentés par une carte dédiée, et commandés par des signaux PWM générés par le calculateur embarqué.

2.2.3 Déplacement

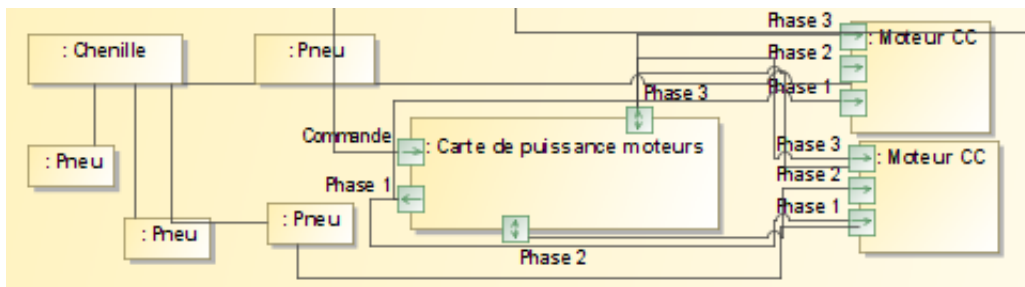


FIGURE 15: Déplacement

Le robot est capable de se déplacer grâce à des chenilles entraînées par des moteurs DC brushless D3542 (documentation en annexe A.2). Ces derniers sont pilotés par le calculateur à travers une carte de puissance dédiée. Les composants principaux de cette carte sont des drivers de moteurs *DRV10975* de chez Texas Instrument dont les principales caractéristiques sont présentes en annexe A.3. Ils peuvent être commandés de différente façon par le Raspberry PI.

L'acquisition des données d'environnement se fait grâce à plusieurs capteurs. Un capteur de couleurs : un TCS230 (voir annexe A.4) et des transducteurs à ultrasons HC-SR04 dont la documentation est en annexe A.5. Les capteurs fournissent leurs informations au calculateur embarqué.

2.2.4 Diagnostic

La partie diagnostique du robot est gérée de façon logicielle par le Raspberry Pi.

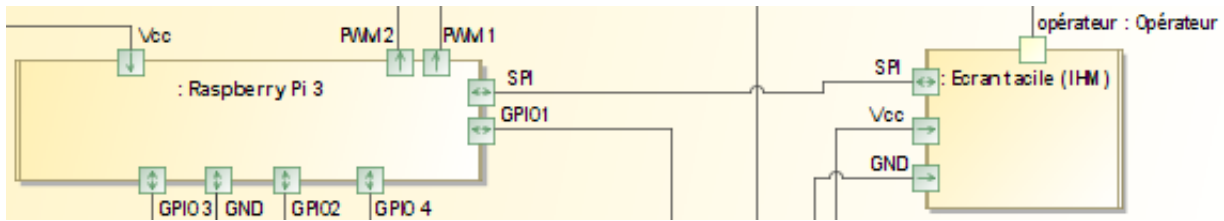


FIGURE 16: Diagnostic

Un écran tactile permet à l'opérateur d'effectuer des réglages et de lancer des opérations de maintenances de façon triviale. Il est interfacé au Raspberry PI via une liaison SPI.

2.3 Diagrammes de séquence du système

2.3.1 Gérer les Obstacles

Voici le diagramme de séquence du Cas d'Utilisation *Gérer les Obstacles* :

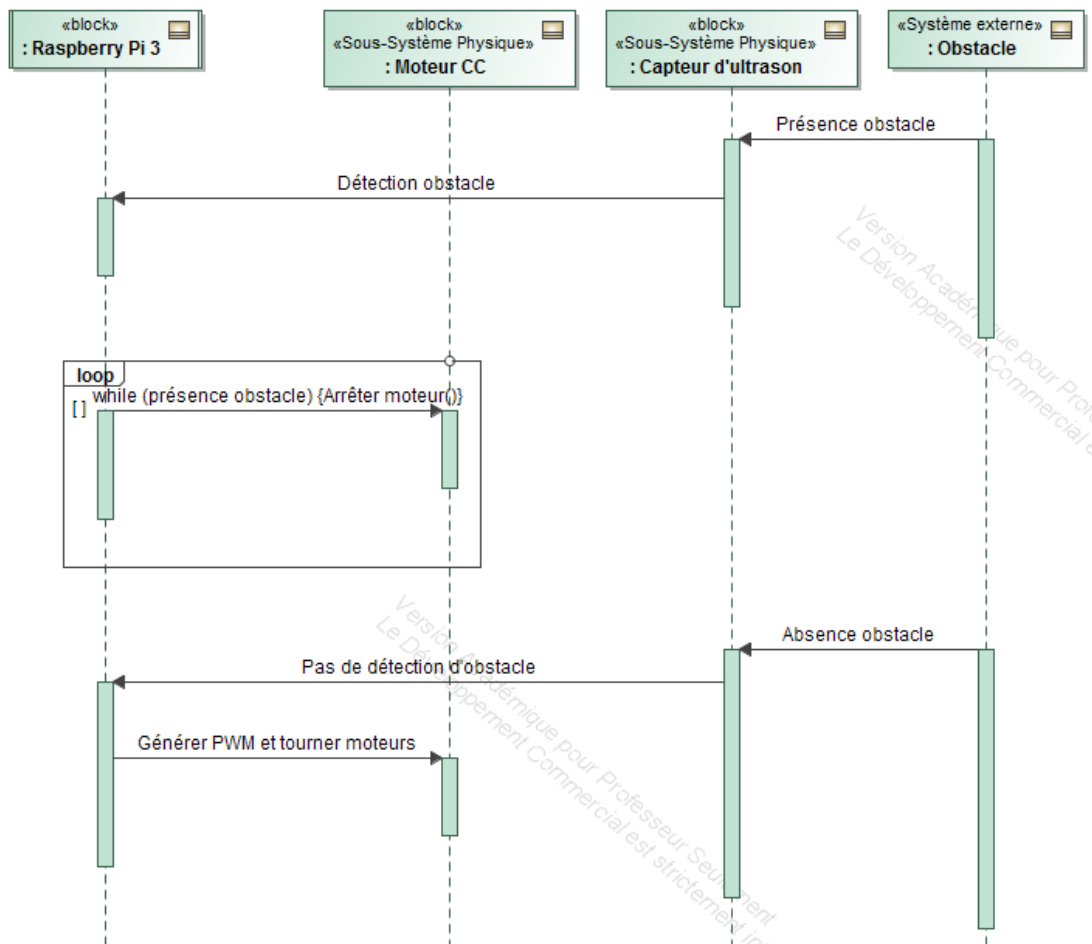


FIGURE 17: Diagramme de séquence - Gérer les Obstacles

2.3.2 Tester

Voici le diagramme de séquence du Cas d'Utilisation *Tester* :

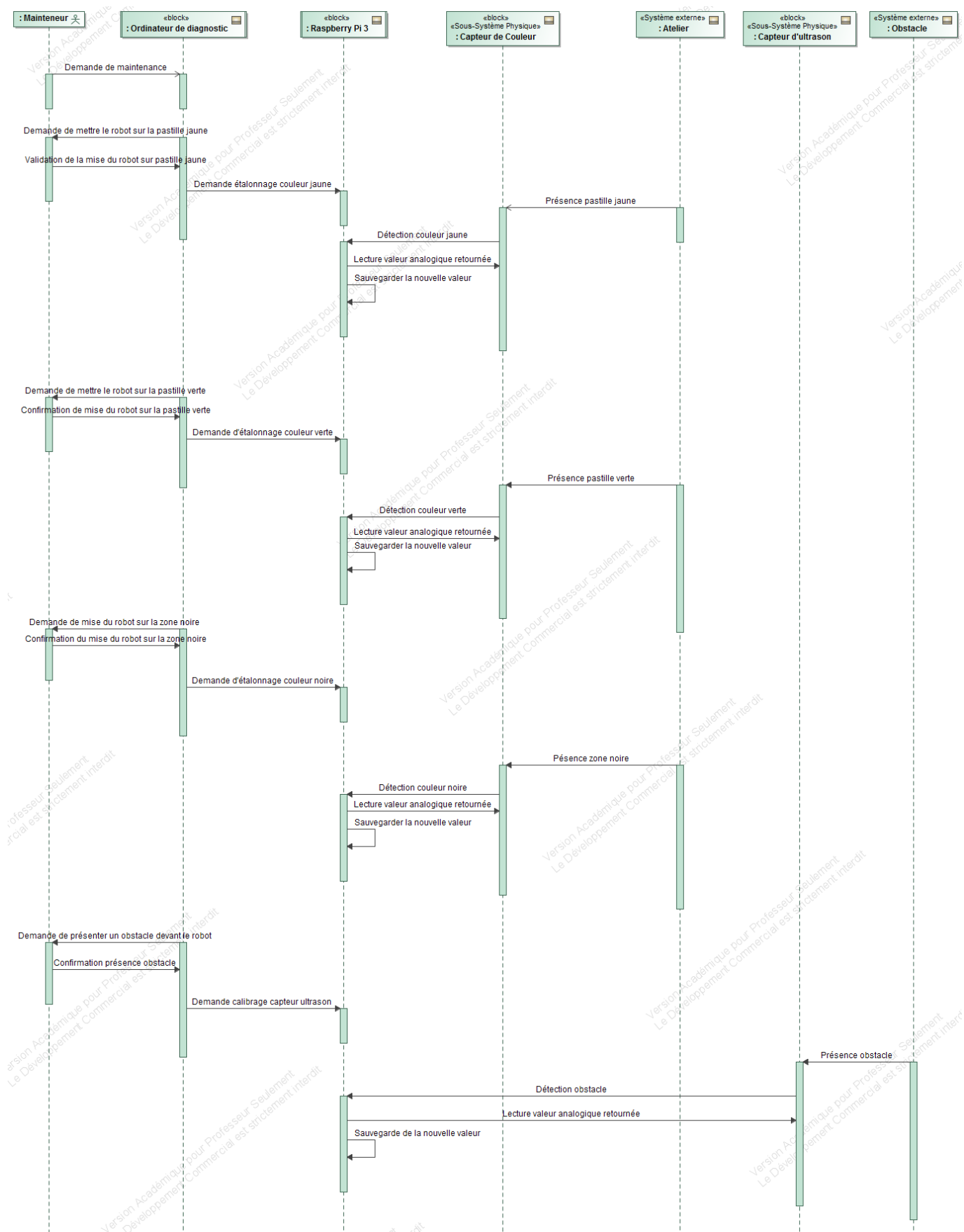


FIGURE 18: Diagramme de séquence - Tester

2.3.3 Se Déplacer

Voici le diagramme de séquence du Cas d'Utilisation *se Déplacer* :

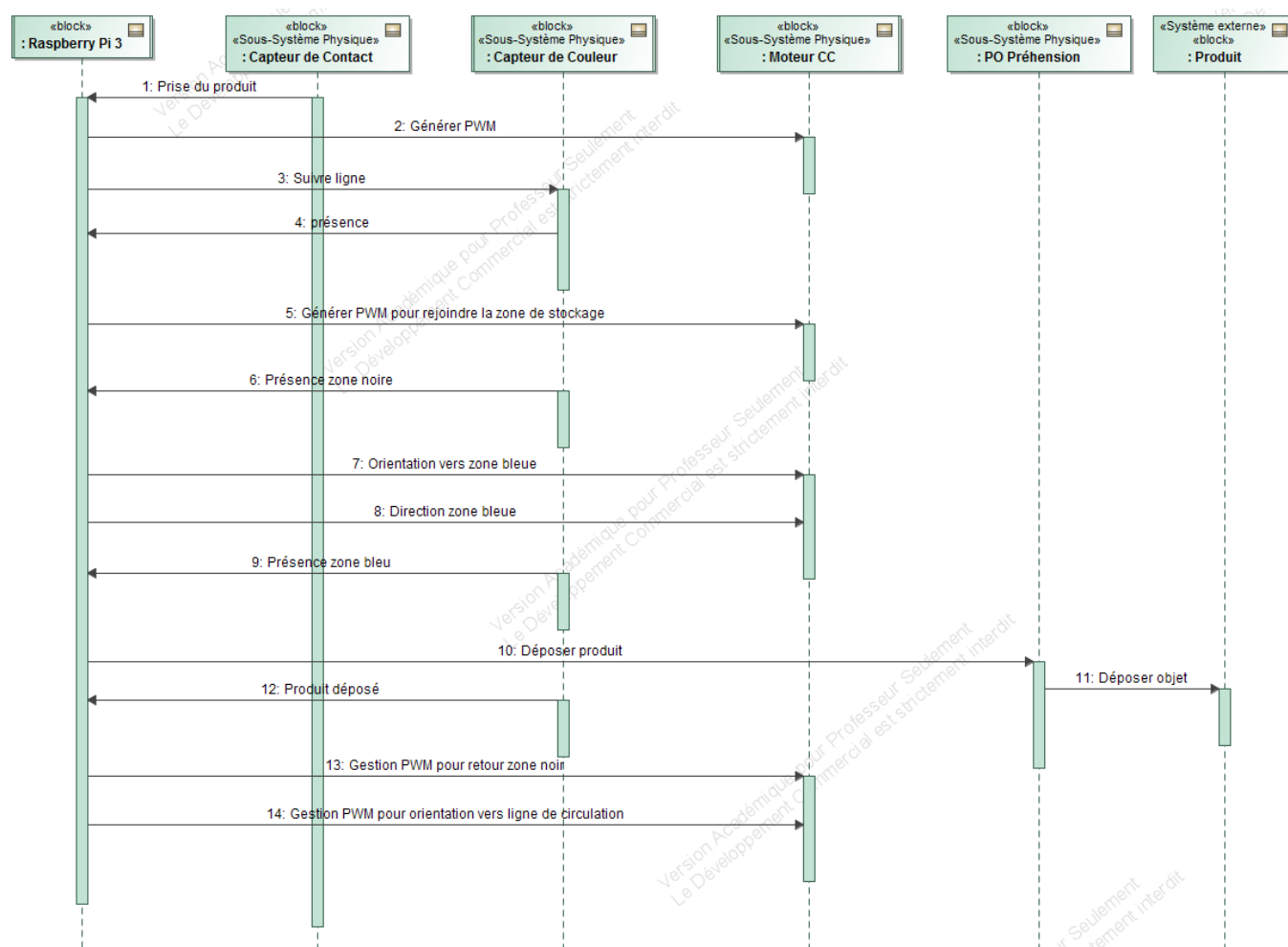


FIGURE 19: Diagramme de séquence - se Déplacer

2.3.4 Stocker

Voici une partie du diagramme de séquence du Cas d'Utilisation *Stocker* :

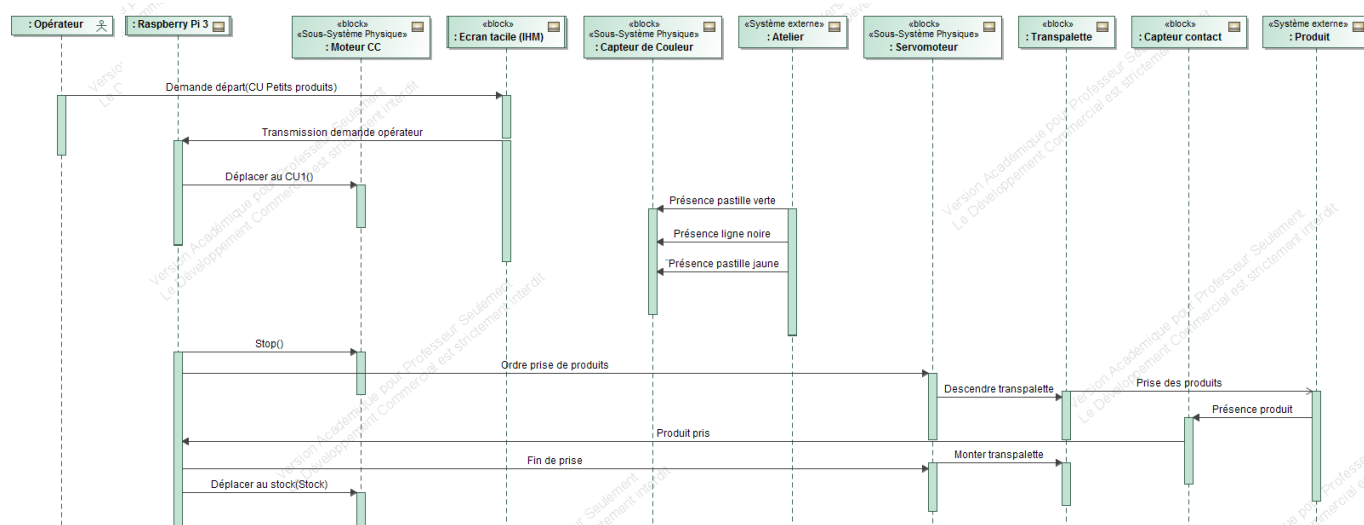


FIGURE 20: Diagramme de séquence - Stocker

2.3.5 Tester et Régler

Voici le diagramme de séquence du Cas d'Utilisation *Tester et Régler* :

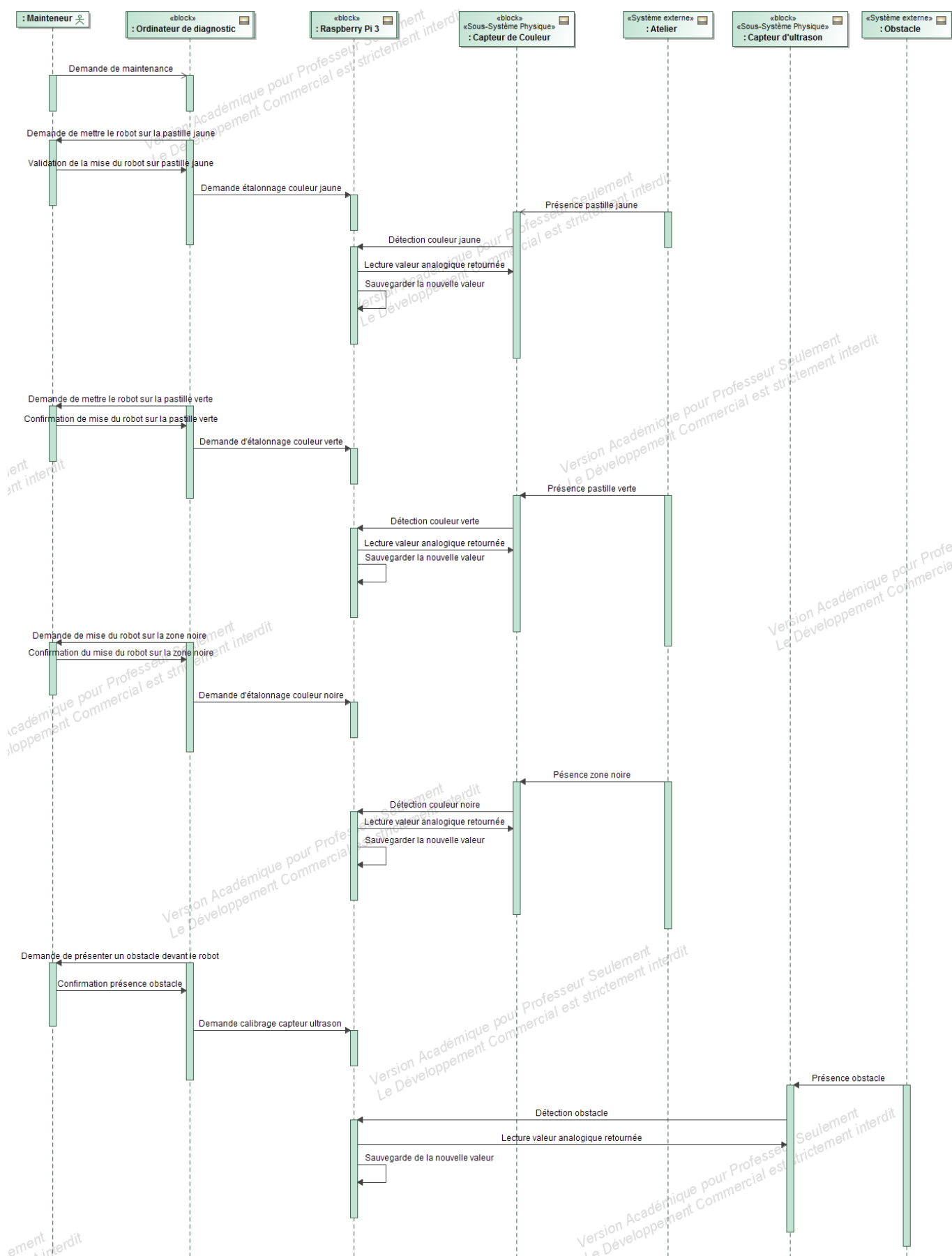


FIGURE 21: Diagramme de séquence - Tester et Régler

2.3.6 Suivi de ligne

Voici le diagramme de séquence du Cas d'Utilisation *Suivi de ligne* :

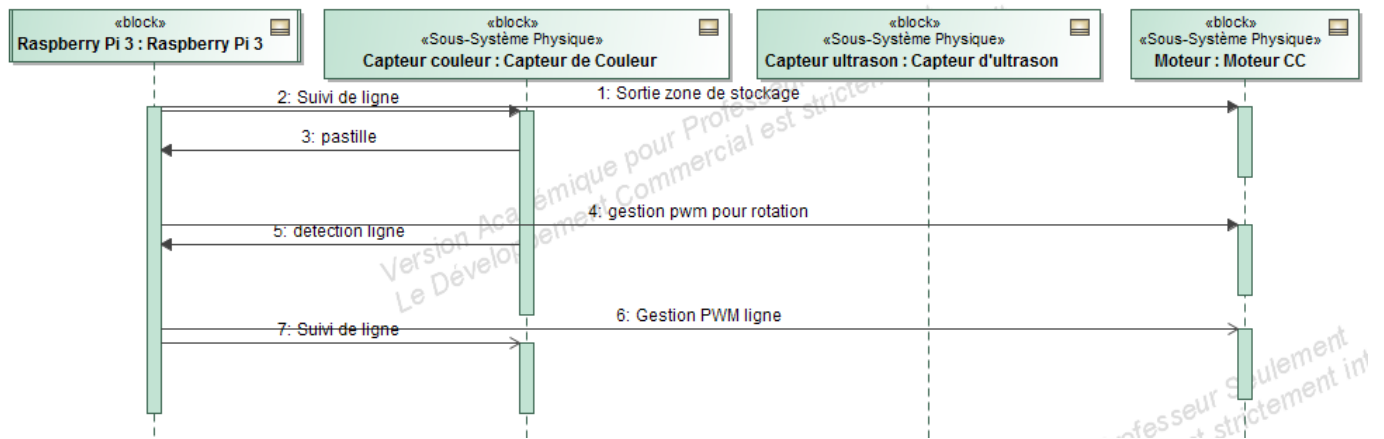


FIGURE 22: Diagramme de séquence - Suivi de ligne

2.4 Matrice Opérations-Exigences

Voici la matrice des opérations/exigences :

		Exigences système [2-AE-Analyse des exigences]																																						
		<div><div><div>Adapter le préhenseur aux produits</div><div>Améliorer la productivité de la logistique interne</div><div>Arrêter devant un obstacle</div><div>Assembler avec les composants des kits LEGO</div><div>Assembler et configurer en moins de 20 minutes</div><div>Détecter l'obstacle à 10cm</div><div>Détecter les couleurs au sol</div><div>Dimensionner le robot au repos dans un cube de 30 cm d'arrête</div><div>Disposer d'un dossier technique</div><div>Disposer d'une ergonomie adaptée pour la maintenance curative</div><div>Disposer d'une ergonomie adaptée pour la mise en marche</div><div>Être testable et maintenable</div><div>Evoluer dans un environnement industriel simulé</div><div>Faciliter les actions de maintenance</div><div>Gérer obstacle</div><div>Implanter les lois de commande sur FPGA</div><div>Mettre en marche par un opérateur en situant les petits produits</div><div>Mettre en mode maintenance curative par un opérateur</div><div>Mettre en stock les produits présents sur les centres d'usinage</div><div>Opérer avec les produits spécifiés</div><div>Opérer dans des conditions climatiques déterminées</div><div>Opérer dans l'atelier et la zone de stockage spécifiés</div><div>Opérer sans maintenance périodique au moins 6 missions successives ...</div><div>Porter indifféremment 3 petits ou grands produits</div><div>Prendre et poser les produits verticalement dans les zones</div><div>Réaliser au moins 6 missions élémentaires successives</div><div>Réaliser des essais en faisant varier les conditions d'éclairage</div><div>Réaliser la mission élémentaire en autonomie</div><div>Réaliser une maintenance curative</div><div>Réaliser une mission élémentaire en moins de 4 minutes</div><div>Régler les paramètres de détection</div><div>Repatrié des l'obstacle évacué</div><div>Repositionner et redémarrer le robot à l'endroit de l'incident</div><div>Suivre les lignes des allées de circulation</div><div>Suivre les lignes pour éviter les zones des CU</div><div>Tester et régler</div><div>Transporter les produits d'un CU sans contact avec le sol</div><div>Transporter les produits de l'atelier vers leur zone de stockage respect...</div><div>Utiliser des piles rechargeables</div><div>Utiliser un robot mobile</div></div></div>																																						
3-Conception générale		1	1	1	4	4	1	1	4	1	1	1	5	1	4	2																								
PC Robot de manutention gestion prise d'objet																																								
Robot de manutention		19	↗	↗	↗		↗	↗			↗	↗	↗				↗																							
Déplacement																																								
PO Déplacement																																								
PO Plateforme à chenilles		24		↗	↗	↗	↗	↗	↗				↗	↗	↗			↗																						
Diagnostic																																								
PC Diagnostic		3																																						
PC Robot de manutention		31		↗	↗			↗	↗	↗	↗	↗	↗	↗	↗	↗		↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗	↗		↗	↗	↗	↗	↗	↗	↗	↗	
Préhension																																								
PO Préhension		20	↗		↗	↗		↗			↗	↗																												

FIGURE 23: Matrice Opérations-Exigences

Conclusion

Ce projet nous a permis de mettre en œuvre nos connaissances en modélisation système à l'aide du langage SysML.

Nous avons pu, à travers un exemple concret de système complexe et à partir d'une liste d'exigences, modéliser un robot de manutention et parfaire nos compétences en ingénierie système.

Nous avons appris à utiliser le logiciel *MagicDraw*, qui nous a permis de parcourir le développement d'un système, de son cahier des charges et ses exigences, jusqu'à sa conception détaillée, en passant par une conception générale et en décrivant son fonctionnement à l'aide de différents diagrammes.

L'étape suivante aurait pu être la simulation, la réalisation et le test du robot, mais ce n'était pas l'objet de ce projet.

A Annexes

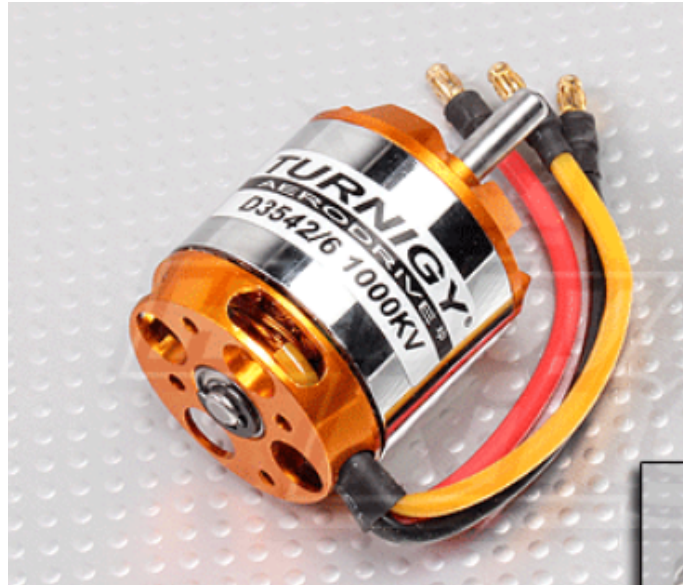
A.1 Batterie LiPo



Type de batterie	Li-Po
Nombre d'éléments	3S
Tension	11,1V
Dimensions	140 x 47 x 27 mm
Masse	370 g
Capacité	4000 mAh et plus
Capacité réelle	5000 mAh
Courant continu	30C et plus
Courant continu réel	175A
Connecteurs fournis	Oui
Type de connecteurs	Deans, T-Plug

FIGURE 24: Batterie LiPo

A.2 Moteurs D3542



Turnigy D3542/6 1000KV Brushless Outrunner Motor

Spec.

Battery: 2~4 Cell /7.4~14.8V

RPM: 1000kv

Max current: 38A

No load current: 2.4A

Max power: 665W

Internal resistance: 0.031 ohm

Weight: 130g (including connectors)

Diameter of shaft: 5mm

Dimensions: 35x42mm

Prop size: 7.4V/11x7 14.8V/10x5

Max thrust: 1260g

FIGURE 25: D3542/6

DRV10975 12-V, Three-Phase, Sensorless BLDC Motor Driver

1 Features

- Input Voltage Range: 6.5 to 18 V
- Total Driver H + L Rdson: <1 Ω
- Drive Current: 1.5 A Continuous (2-A Peak)
- Sensorless Proprietary Back Electromotive Force (BEMF) Control Scheme
- Continuous Sinusoidal 180° Commutation
- No External Sense Resistor Required
- For Flexibility User may Include External Sense Resistor to Monitor Power Delivered to the Motor
- Flexible User Interface Options:
 - I²C Interface: Access Registers for Command and Feedback
 - Dedicated SPEED Pin: Accepts Either Analog or PWM Input
 - Dedicated FG Pin: Provides TACH Feedback
 - Spin Up Profile can be Customized With EEPROM
 - Forward/Reverse Control With DIR Pin
- Integrated Buck/Linear Converter to Efficiently Provide Voltage (5 V/3.3 V) for Internal and External Circuits
- Supply Current 4.5 mA With Standby Version (DRV10975)
- Supply Current 80 μ A With Sleep Version (DRV10975Z)
- Overcurrent Protection
- Lock Detection
- Voltage Surge Protection
- UVLO Protection
- Thermal Shutdown Protection

- Thermally-Enhanced 24-Pin HTSSOP

2 Applications

- Appliance Fan
- HVAC

3 Description

DRV10975 is a three-phase sensorless motor driver with integrated power MOSFETs, which can provide continuous drive current up to 1.5 A. The device is specifically designed for cost-sensitive, low-noise/low external component count applications.

The DRV10975 uses a proprietary sensorless control scheme to provide continuous sinusoidal drive, which significantly reduces the pure tone acoustics that typically occur as a result of commutation. The interface to the device is designed to be simple and flexible. The motor can be controlled directly through PWM, analog, or I²C inputs. Motor speed feedback is available through either the FG pin or I²C.

The DRV10975 features an integrated buck/linear regulator to efficiently step down the supply voltage to either 5 V or 3.3 V for powering both internal and external circuits. The device is available in either a sleep mode or a standby mode version to conserve power when the motor is not running. The standby mode (4.5-mA) version leaves the regulator running and the sleep mode (80- μ A) version shuts it off. Use the standby mode version in applications where the regulator is used to power an external microcontroller.

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
DRV10975	HTSSOP (24)	7.80 mm × 6.40 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

4 Application Schematic

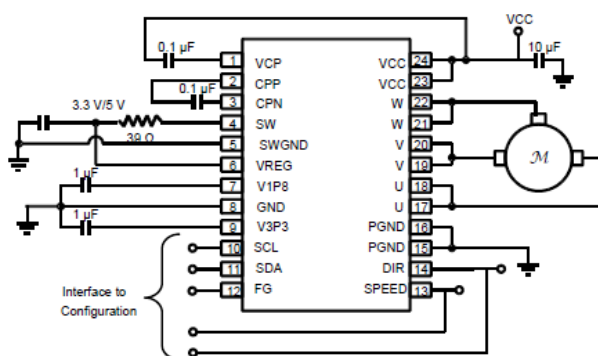


FIGURE 26: Texas Instrument DRV10975

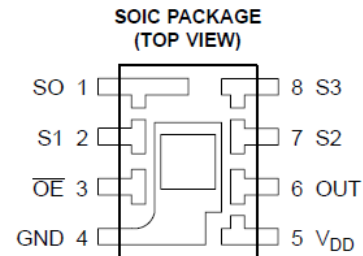
A.4 Capteur de couleurs



TCS230 PROGRAMMABLE COLOR LIGHT-TO-FREQUENCY CONVERTER

TAOS046 - FEBRUARY 2003

- High-Resolution Conversion of Light Intensity to Frequency
- Programmable Color and Full-Scale Output Frequency
- Communicates Directly With a Microcontroller
- Single-Supply Operation (2.7 V to 5.5 V)
- Power Down Feature
- Nonlinearity Error Typically 0.2% at 50 kHz
- Stable 200 ppm/°C Temperature Coefficient
- Low-Profile Surface-Mount Package



Description

The TCS230 programmable color light-to-frequency converter combines configurable silicon photodiodes and a current-to-frequency converter on single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance). The full-scale output frequency can be scaled by one of three preset values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable (OE) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line.

The light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters. The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All 16 photodiodes of the same color are connected in parallel and which type of photodiode the device uses during operation is pin-selectable. Photodiodes are 120 μm x 120 μm in size and are on 144- μm centers.

Functional Block Diagram

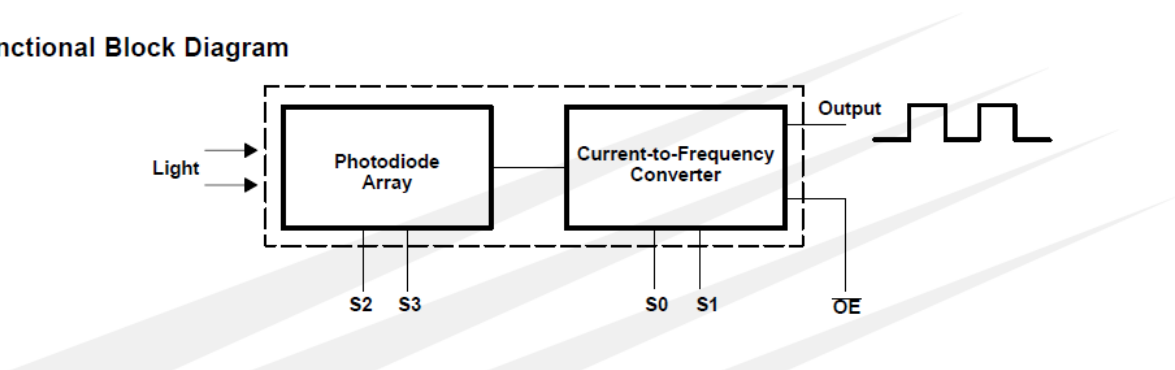
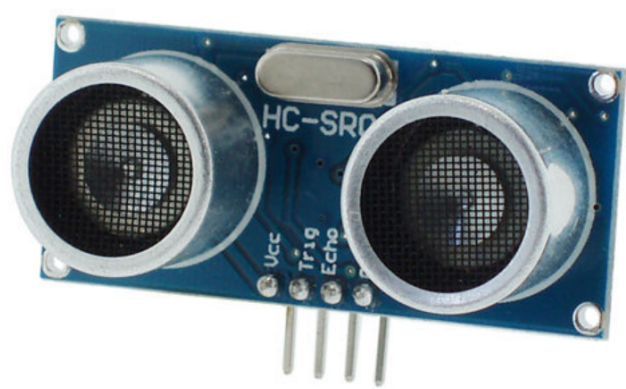


FIGURE 27: TCS230

A.5 HC-SR04



Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

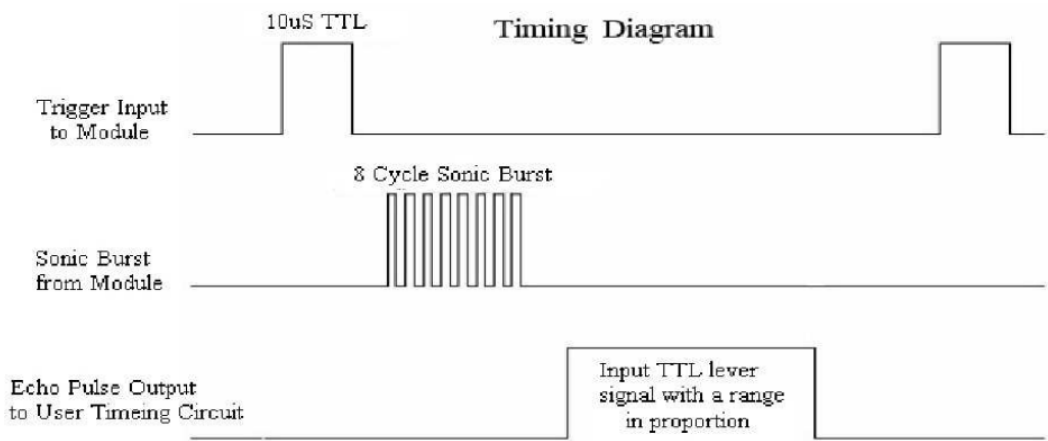


FIGURE 28: HC-SR04