

Rapport de SysML

Conception d'un robot de manutention

Sommaire

Table des figures.....	2
1. Introduction.....	3
2. Diagrammes de conception.....	5
2.1.1. Stockage de produits	5
.....	5
2.1.2. Test et réglage	7
2.1.3. Gestion d'obstacles	8
2.1.4. Maintenance curative.....	9
2.2. Diagramme de blocs Logiciel	10
2.3. Matrice de traçabilité	11
2.4. Diagramme de blocs Physique	12
2.5. Matériel utilisé	13
2.6. Diagramme de blocs interne	18
3. Conclusion	19

Table des figures

Figure 1 : Diagramme de séquence conceptuel Stocker les.....	5
Figure 2 : Diagramme de séquence conceptuel CU Tester et régler.....	7
Figure 3 : Diagramme de séquence conceptuel Gérer obstacle	8
Figure 4 : Diagramme de séquence conceptuel Réaliser une maintenance curative	9
Figure 5 : Diagramme de bloc.....	10
Figure 6 : Matrice de traçabilité	11
Figure 7 : Diagramme de bloc physique	12
Figure 8 : Détails concernant le servo-moteur NXT	13
Figure 9 : Système pour mesurer les déplacements du robot	13
Figure 10 : Datasheet du servo-moteur NXT.....	14
Figure 11 : Architecture du servo-moteur.....	14
Figure 12 : Capteur de couleur.....	15
Figure 13 : Capteur ultrason.....	16
Figure 14 : Capteur suiveur de ligne.....	17
Figure 15 : Préhenseur du robot	17
Figure 16 : Architecture physique du robot	18

1. Introduction

L'objectif de ce projet était de faire des choix matériels et logiciels afin de permettre à un robot de se déplacer dans un environnement et de réaliser une tâche. La tâche du robot étant de transporter et de déposer dans des zones spécifiques des pièces composées de Lego, il existe deux types de pièces : celles de grandes tailles et celles petites tailles, sachant qu'il existe une zone de dépôt pour chaque type de Lego (ie une zone de dépôt pour les pièces de grande taille, une autre pour celles de petites tailles).

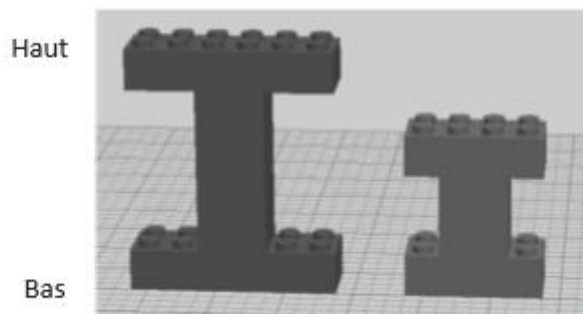
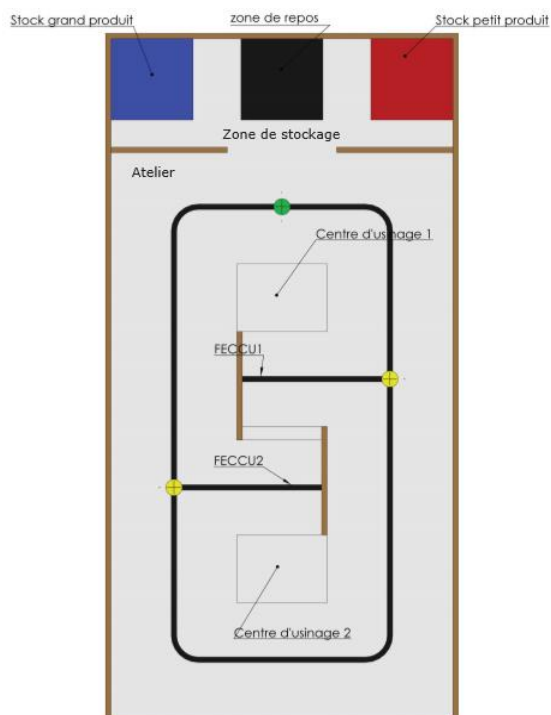


Photo d'un grand produit et d'un petit produit (couleurs non contractuelles)

Ci-dessous les détails de l'environnement où se déplacera le robot :



Nous remarquons que l'environnement est divisé en deux espaces : une zone où seront situées les zones de dépôt et la zone de repos, et une zone qui correspond à l'atelier, ce sera là que le robot devra évoluer pour récupérer les différentes pièces.

Dans l'atelier, il y a des marqueurs colorés (ou pastilles) au sol qui permettront au robot de se déplacer : une ligne noire, qui sera le chemin suivi par le robot, et des marqueurs couleurs, qui permettent de signaler au robot les différentes intersections où il lui sera possible/nécessaire de tourner (par exemple pour aller récupérer des pièces).

Les zones appelées "Centre d'usinage" seront les zones où les pièces seront assemblées, le robot ne pourra pas accéder à ses zones et aura simplement accès à ce qui est nommé FECCUi dans le dessin. Ce sont les endroits où les pièces en attente d'être amenées en zone de dépôt seront entreposées.

Dans la suite de ce rapport, nous détaillerons les diagrammes de séquence expliquant la façon dont le robot va interagir et évoluer dans son environnement afin de pouvoir réaliser la tâche qui lui est attribuée :

- Un diagramme expliquant comment le robot va réaliser le stockage, sera également spécifié le comportement du robot quand il rencontre un marqueur coloré au sol et quand il doit suivre une ligne noire ;
- Un diagramme de séquence explicitant le déroulement des tests et réglages à faire sur le robot (par exemple pour les capteurs couleurs) ;
- Un diagramme de séquence qui explique comment réagit le robot en cas de détection d'obstacle ;
- Un diagramme expliquant comment se passe la maintenance du système.

Puis seront précisés, à l'aide de diagramme de blocs, les choix logiciels et les choix matériels qui ont été faits. Nous détaillerons en plus les choix de moteur et de capteurs qui ont été faits pour répondre à cette problématique, avant d'expliciter l'architecture physique complète de notre système.

NB : Dans les différents diagrammes de blocs les ports d'entrée/sortie, ainsi que les liens entre les différents composants seront représentés.

2. Diagrammes de conception

2.1.1. Stockage de produits

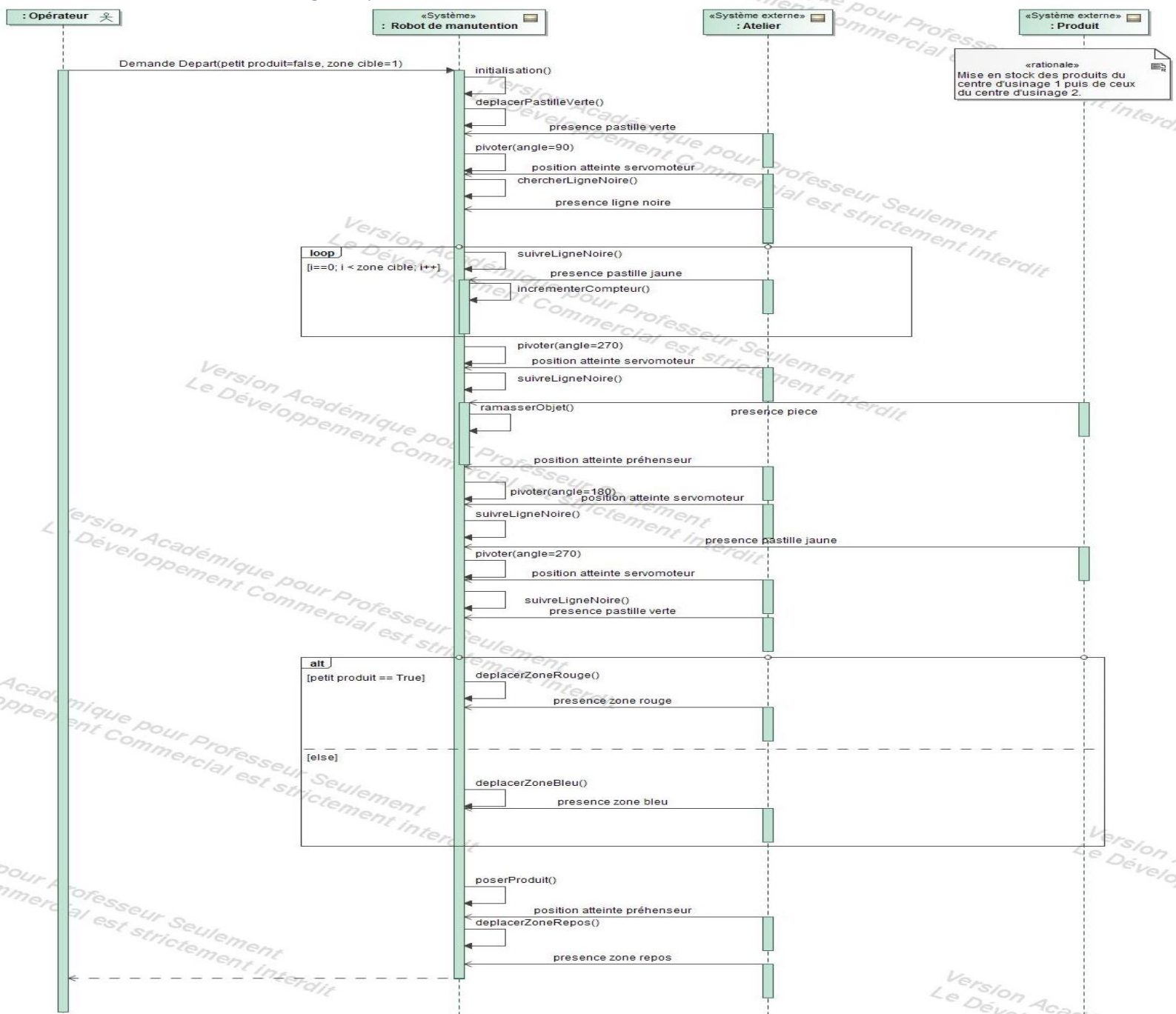


Figure 1 : Diagramme de séquence conceptuel Stocker les

Le diagramme ci-dessus explicite le comportement du robot, depuis la demande de démarrage de l'utilisateur au retour de la zone de repos.

La première tâche du robot consiste à s'initialiser, sortir de la zone de repos puis retrouver la pastille verte, qui se situe sur le parcours à suivre du robot dans l'atelier.

Une fois cette pastille verte trouvée, le robot devra suivre la ligne noire jusqu'à arriver à une prochaine pastille de couleur jaune. Cette pastille jaune signale qu'une zone des pièces usinées en

attente d'être mise en dépôt est présente. Le robot devra alors effectuer une rotation pour se diriger vers les objets à récupérer avant de les ramasser s'ils sont bien présents. Le robot se dirigera ensuite vers la zone de dépôt adéquate (à choisir selon la taille des pièces) en suivant la ligne noire et en se servant de la pastille verte comme point de repère (la pastille verte témoigne de l'arrivée devant les zones de dépôt), avant de retourner en zone de repos.

2.1.2. Test et réglage

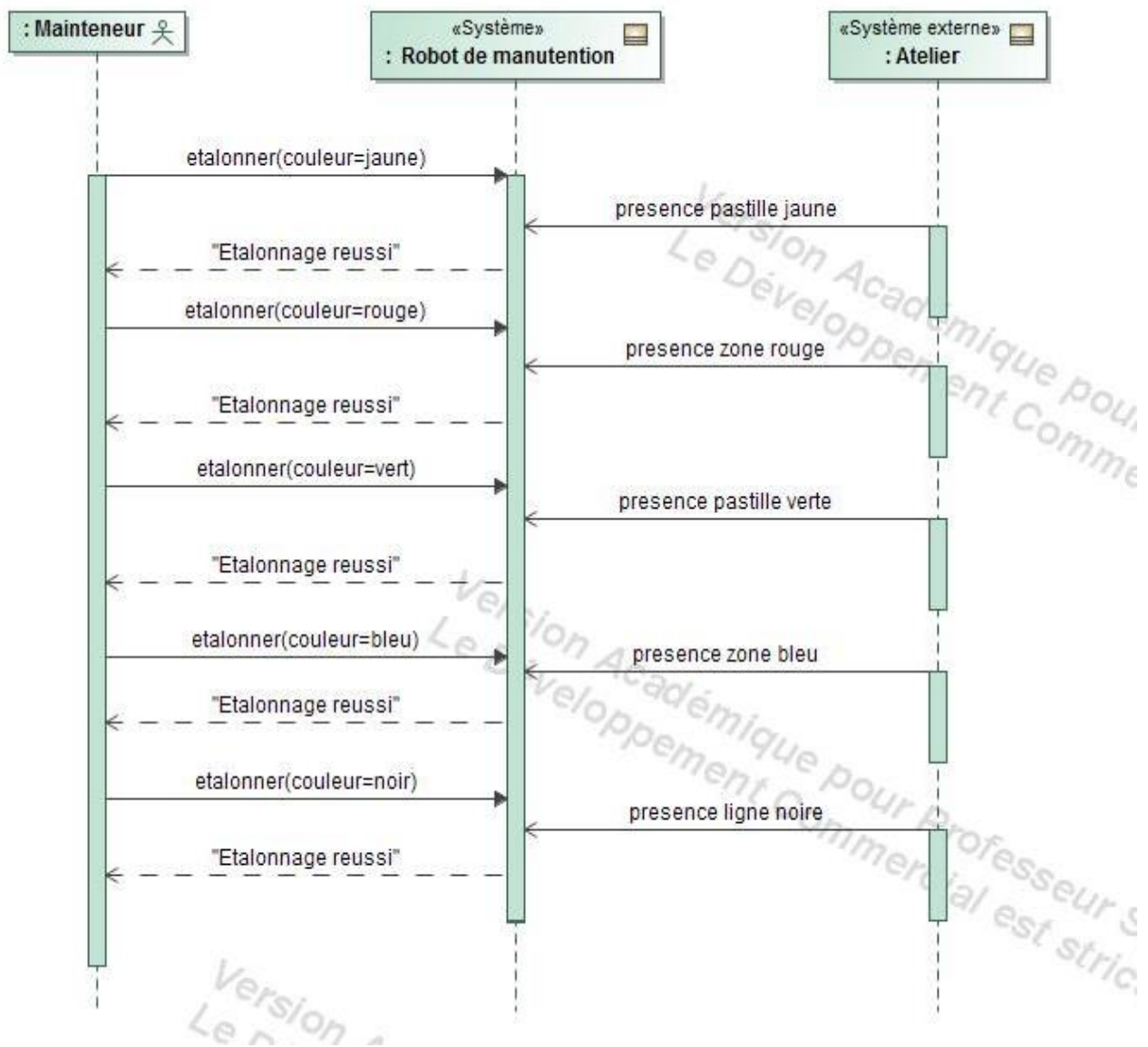


Figure 2 : Diagramme de séquence conceptuel CU Tester et régler

Dans ce diagramme, nous expliquons la façon dont les capteurs du robot sont étalonnés. Il faut en effet régler les capteurs couleurs du robot pour qu'ils soient sensibles aux couleurs vert (pastille verte), jaune (pastille jaune), rouge (zone de dépôt rouge), bleu (zone de dépôt bleue) et noir (ligne noire). Les déplacements du robot reposent sur la bonne reconnaissance de ces couleurs.

2.1.3. Gestion d'obstacles

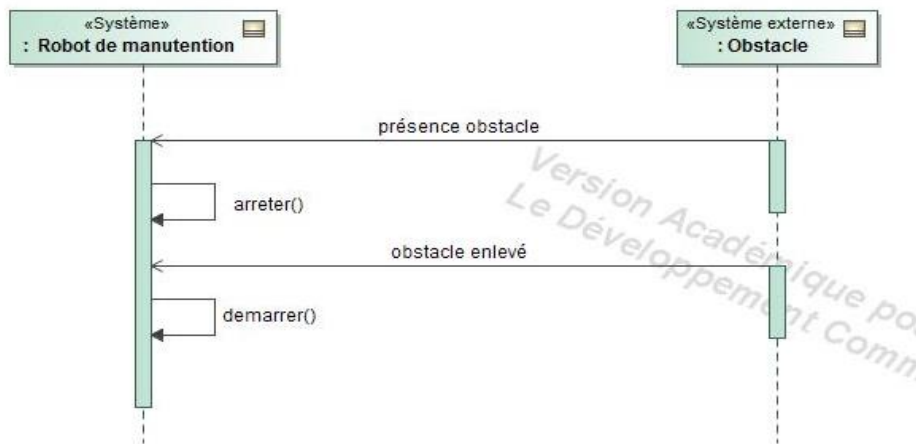


Figure 3 : Diagramme de séquence conceptuel Gérer obstacle

Ce diagramme explique succinctement la façon dont le robot réagira lorsqu'il rencontrera un obstacle. Le robot s'arrêtera et ne redémarrera que quand son parcours sera dégagé.

2.1.4. Maintenance curative

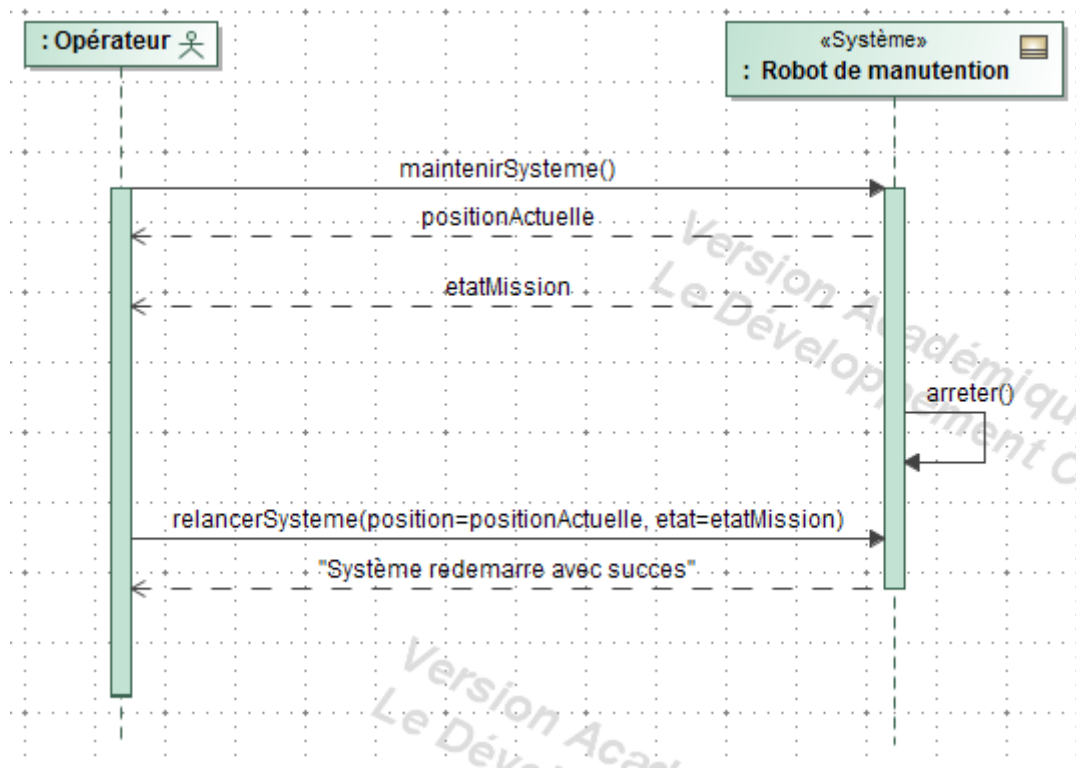


Figure 4 : Diagramme de séquence conceptuel Réaliser une maintenance curative

Sur ce diagramme, nous voyons comment se déroule une maintenance curative du système. Lorsque l'opérateur effectue une maintenance le robot de manutention l'informe de la position actuelle et de l'état mission avant de s'éteindre. Lorsqu'il voudra relancer le système, l'opérateur enverra une commande avec en paramètre la position actuelle et l'état de la mission pour que le robot reprenne sa tâche là où il l'avait interrompu.

2.2. Diagramme de blocs Logiciel

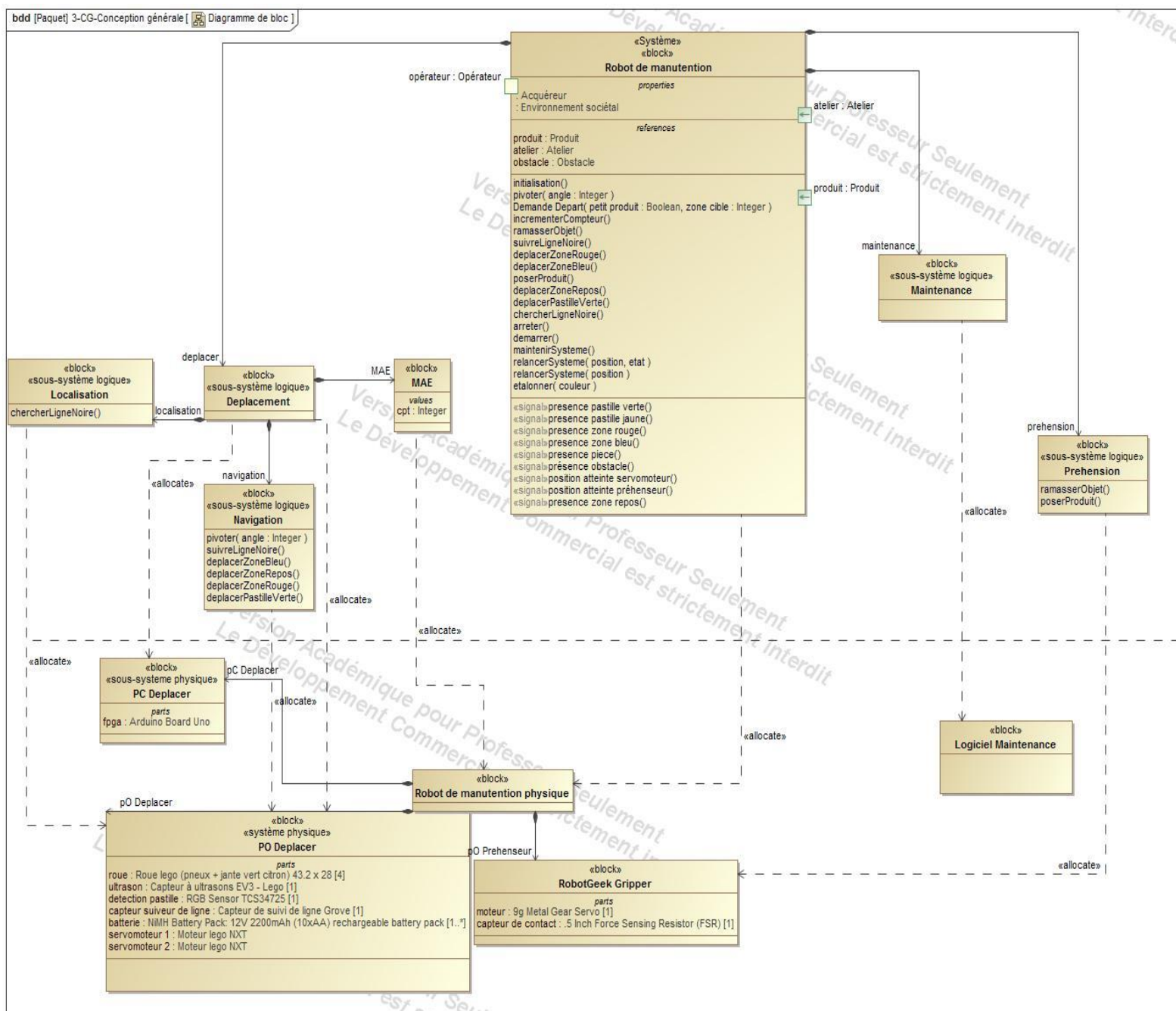


Figure 5 : Diagramme de bloc

2.3. Matrice de traçabilité

		Logiciel	Maintena...	PC Deplacer	PO Deplacer	Robot de manute...	RobotGeek Gripper
logique		1	1	3	2	1	
Déplacement	2		↗	↗			
Localisation	1			↗			
MAE	1				↗		
Maintenance	1	↗					
Navigation	1			↗			
Prehension	1					↗	
Robot de manut	1						↗

Figure 6 : Matrice de traçabilité

La matrice de traçabilité répertorie les briques logicielles qui seront installées dans les différentes parties physiques. Pour valider notre conception, chaque élément logique doit être associé à un élément physique.

Ces exigences correspondent aux associations de type « allocate » sur le diagramme de blocs précédent.

2.4. Diagramme de blocs Physique

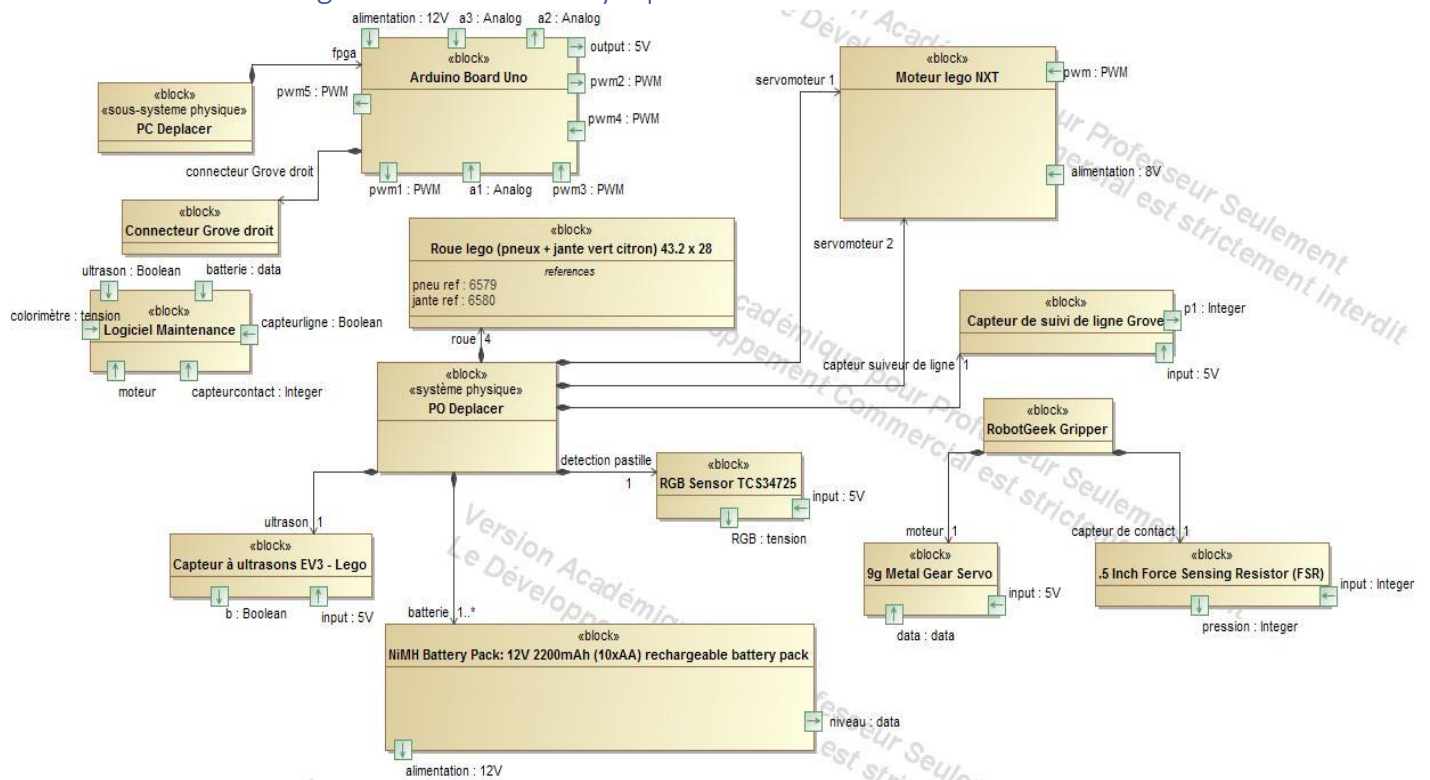


Figure 7 : Diagramme de bloc physique

2.5. Matériel utilisé

Arduino UNO : utilisé en tant que FPGA, ce microcontrôleur permet de programmer notre robot de manutention

Moteur lego NXT :

Servo-moteur NXT

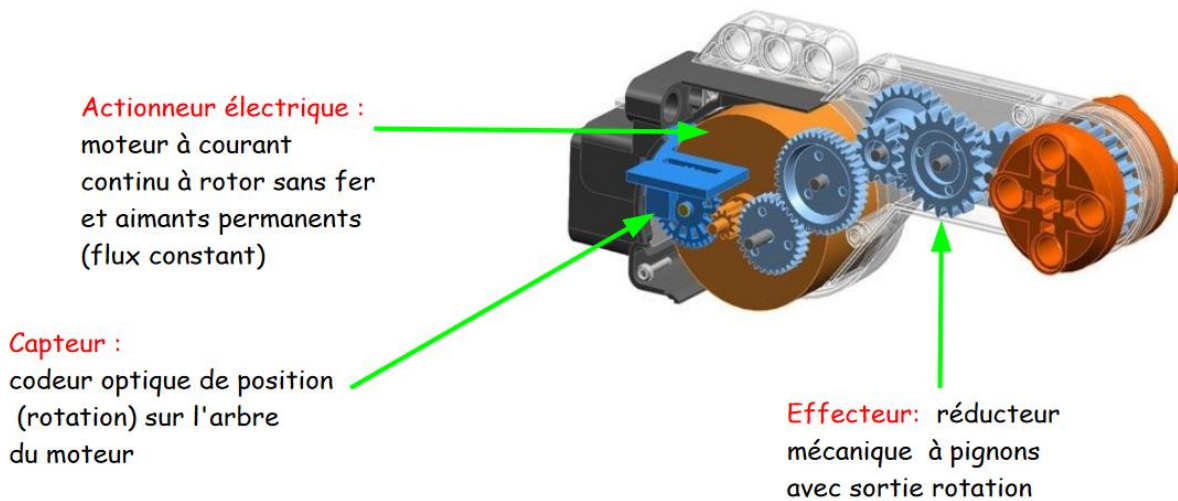


Figure 8 : Détails concernant le servo-moteur NXT

Servo-moteur NXT : Codeur optique

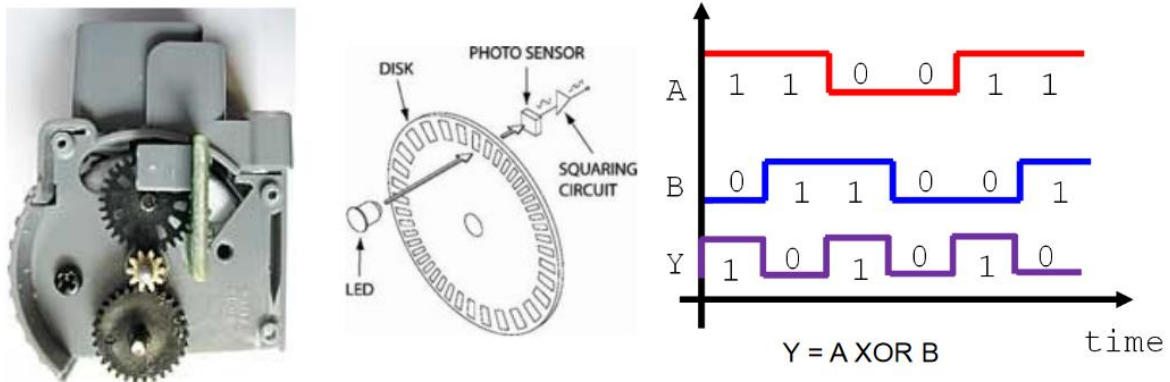


Figure 9 : Système pour mesurer les déplacements du robot

Servo-moteur NXT : Câblage général

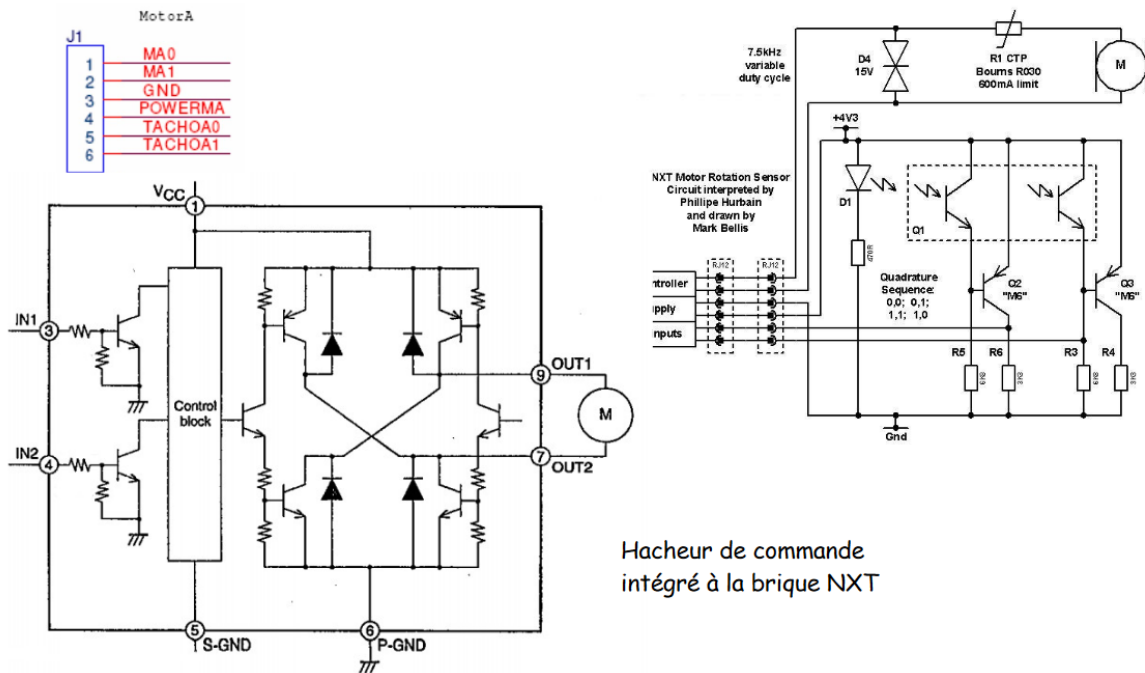


Figure 10 : Datasheet du servo-moteur NXT

Servo-moteur NXT : Hiérarchie du contrôle

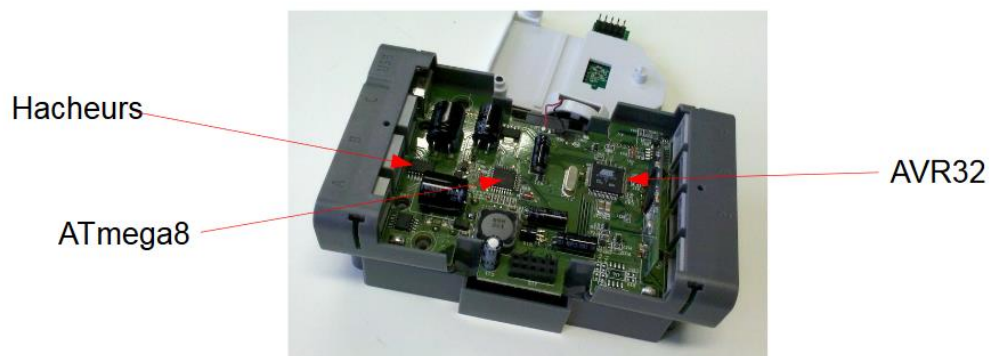
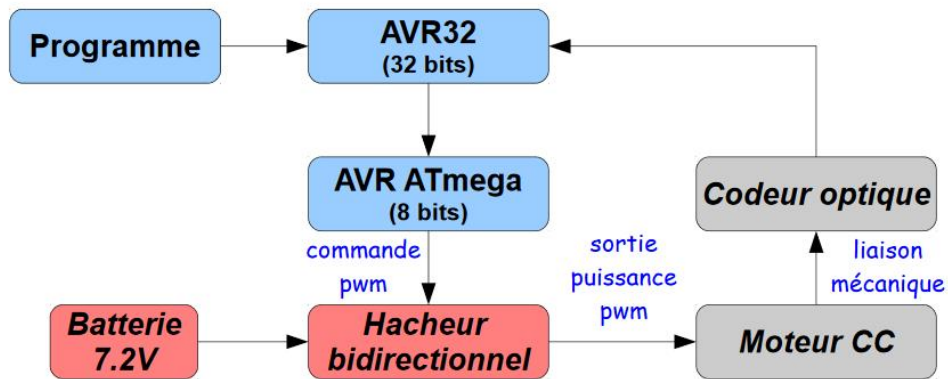


Figure 11 : Architecture du servo-moteur

Servo-moteur NXT: équations

Equation électrique: $L \frac{di_a}{dt} + R i_a + e_f = u_a$

avec $e_f = K_e \omega$

Equation mécanique: $J \frac{d\omega}{dt} + f \omega = C_m - C_r$

avec $C_m = K_m i_a$

En régime statique, les dérivées sont nulles.

$$R I_a + K_e \Omega = U_a$$

$$K_m I_a - f \Omega = C_r$$

La relation statique est donc :

$$\begin{bmatrix} R & K_e \\ K_m & -f \end{bmatrix} \begin{bmatrix} I_a \\ \Omega \end{bmatrix} = \begin{bmatrix} U_a \\ C_r \end{bmatrix}$$

Les valeurs des paramètres peuvent être déduites de quelques expériences

i_a	courant rotor	A
e_a	tension rotor	V
e_f	force contre électromotrice	V
θ	angle arbre sortie	rad
ω	vitesse arbre sortie	rad/s
C_m	couple moteur	N.m
C_r	couple résistant	N.m
L	inductance rotor	H
R	résistance rotor	Ω
K_m	constante de couple méca	N.m/A
K_e	constante de fem	V/rad/s
J	moment d'inertie	kg.m ²
f	coefficient de frottement visqueux	Nm/rad/s

RGB Sensor TCS34725 :

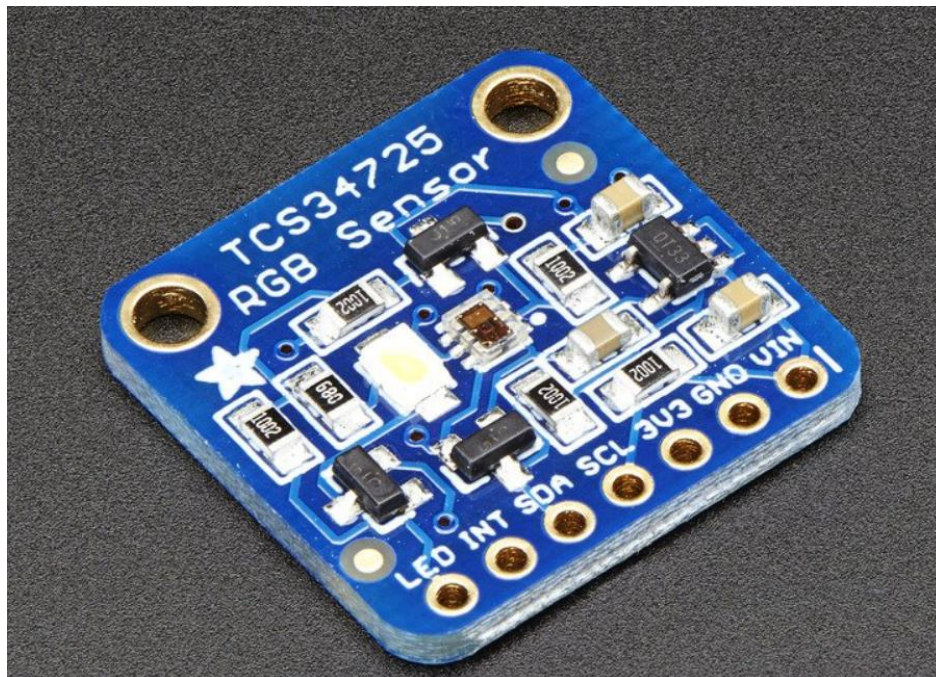


Figure 12 : Capteur de couleur

Un filtre de blocage IR est intégré sur la puce et est localisé sur les photodiodes de détection de couleur cela permet de calculer la composante IR spectrale de la lumière entrante et permet de réaliser des mesures de couleurs avec précision.

Capteur à ultrasons EV3-Lego :



Figure 13 : Capteur ultrason

Le **capteur numérique à ultrasons EV3** génère des vagues d'ultrasons dans l'environnement proche. En analysant les échos de ces ondes, il détecte et mesure la distance le séparant d'objets et d'obstacles à proximité.

Ce capteur peut aussi fonctionner comme un sonar, ou déclencher un programme lorsqu'il détecte une onde sonore.

Capteur de suivi de ligne Grove + Connecteur Grove droit :

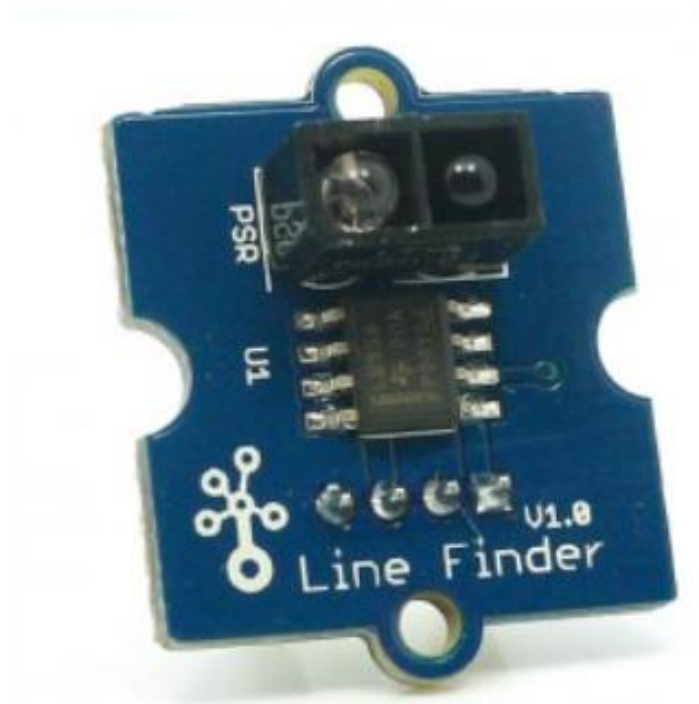


Figure 14 : Capteur suiveur de ligne

Deux LEDS infrarouges aux fonctions différentes : l'une est en effet émettrice et l'autre, réceptrice. Le signal digital émis par le capteur de ligne Grove à sa platine permet au robot mobile de suivre une ligne noire sur fond blanc aussi bien qu'une ligne blanche sur fond noir.

Préhenseur RobotGeek Gripper :



Figure 15 : Préhenseur du robot

2.6. Diagramme de blocs interne

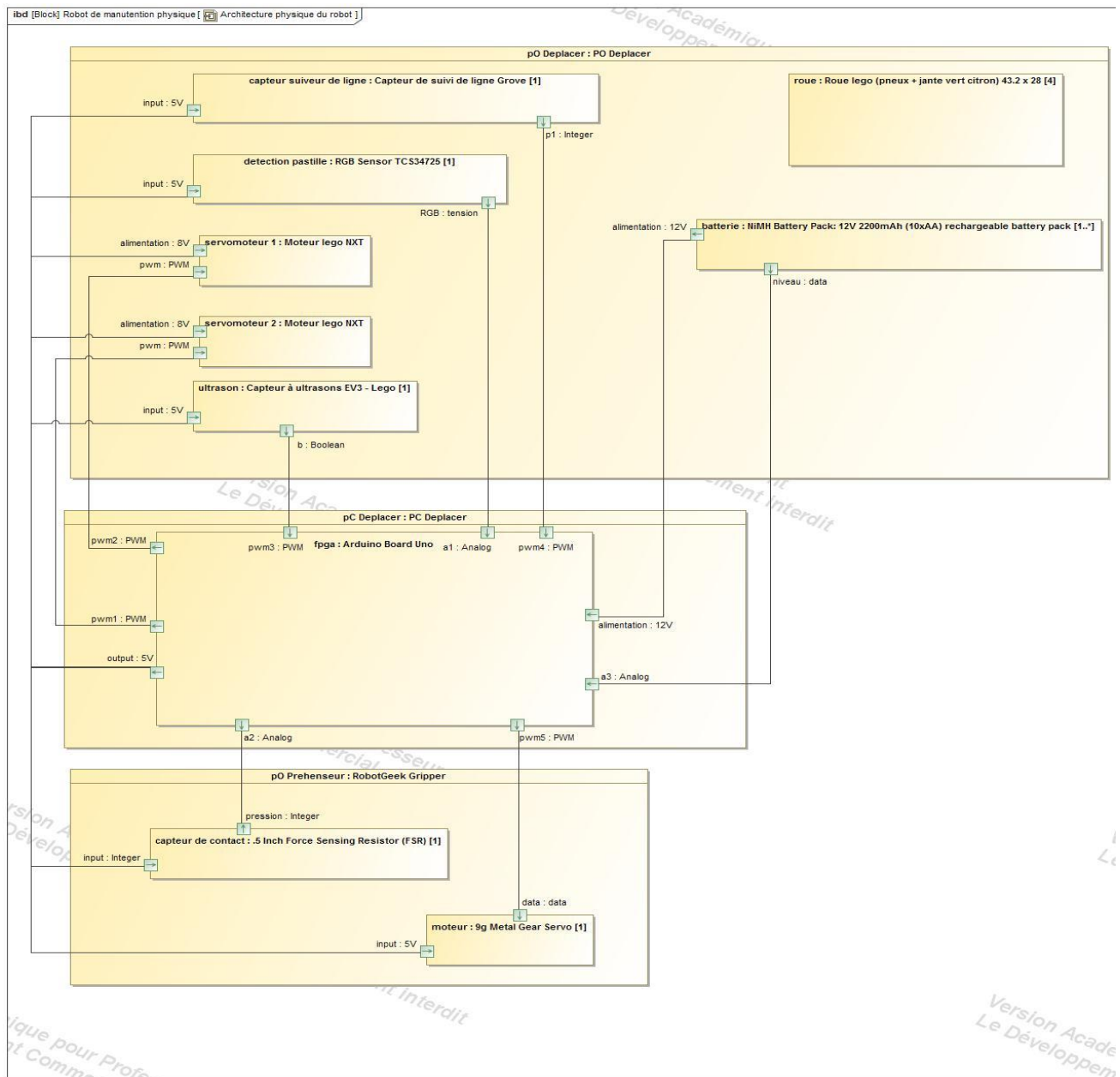


Figure 16 : Architecture physique du robot

Ce diagramme définit les différentes liaisons physiques au travers de ports de communication de chaque matériel physique.

L'Arduino présenté précédemment représente le cerveau du robot, et l'ensemble des informations transitent par ce dernier.

3. Conclusion

Ce projet nous a permis de voir la complexité qu'il y a à réfléchir à la mise en oeuvre d'une tâche par un robot : en plus de devoir penser aux détails du comportement du robot (représenté ici par les divers diagrammes de séquence), nous devons aussi méditer sur les choix matériels : quels capteurs seront les plus adaptés pour répondre à notre problème ? quel moteur choisir ? quel préhenseur choisir ? que est le meilleur choix de batterie pour que le robot puisse répéter la même tâche un nombre spécifique de fois, en prenant en compte qu'elle doit être compatible avec le voltage du reste du matériel ? Que choisir comme carte électronique, sachant qu'elle doit pouvoir communiquer avec les capteurs et autres composants ? etc.

En parallèle, nous devons également expliciter la façon dont se passeraient les interactions entre les différents composants (logiciels et matériels), en spécifiant les ports d'entrée/sortie, les liens entre les composants et en exploitant des matrices de traçabilité pour être sûr que rien n'était oublié.

En résumé, il s'agissait là d'une étude assez complète du problème soumis, en s'arrêtant avant la phase de développement où il nous aurait fallu programmer.