



Design III

Remise 1

présenté à

Philippe Giguère, Dominique Grenier, Denis Laurendeau

par

Équipe 7 — Zière

<i>matricule</i>	<i>nom</i>	<i>signature</i>
111 114 478	Garvin, Sébastien	
111 040 128	Kedzierski, Xavier	
111 066 466	Magnan, Charles-Olivier	
111 071 384	Provencher, Jean-Michel	
111 073 630	Bourque, Emile	
111 075 478	Sylvain, Matthieu	
111 074 361	Brown, Jérémy	
907 196 009	Garneau, Laurent	

Université Laval
31 janvier 2016

Historique des versions		
<i>version</i>	<i>date</i>	<i>description</i>
1.0	24 janvier 2016	Création du document
2.0	31 janvier 2016	Remise 1

Table des matières

1 Diagrammes	1
1.1 Diagramme de contexte	1
1.2 Diagramme de classes	2
1.3 Diagramme de séquences	4
2 Description des cas d'utilisation	6
2.1 Diagramme des cas d'utilisation	6
2.2 Scénarios des cas d'utilisation	7
3 Familiarisation avec équipements et expériences préliminaires	18
3.1 Structure mécanique	18
3.2 Système de préhenseur et d'électroaimant	20
3.3 Contrôle de la caméra	21
3.4 Contrôle des moteurs	23
3.5 Localisation du robot et des îles	23
3.6 Calcul de la trajectoire	24
3.7 Repérage des trésors et de la station de recharge	24
3.8 Alimentation du robot	26
3.9 Transmission du code Manchester	27
3.10 Liste des pièces achetées	28
Bibliographie	29

Chapitre 1

Diagrammes

1.1 Diagramme de contexte

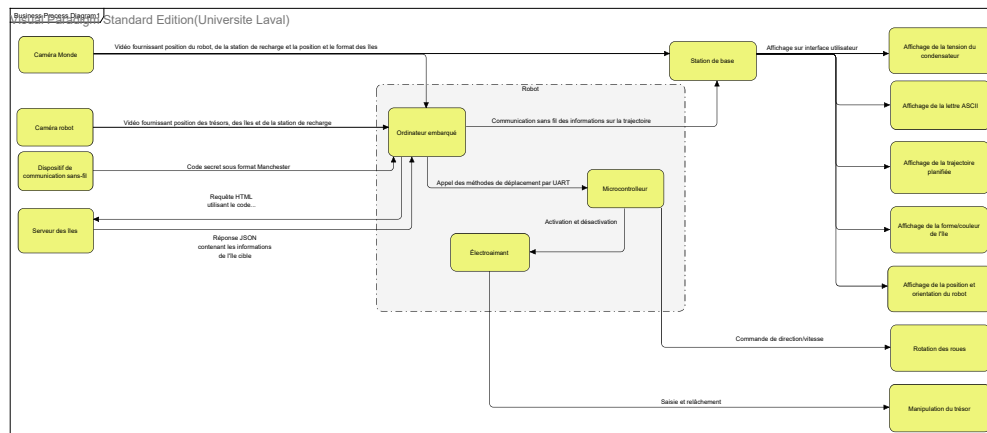


FIGURE 1.1 – Diagramme de contexte

1.2 Diagramme de classes

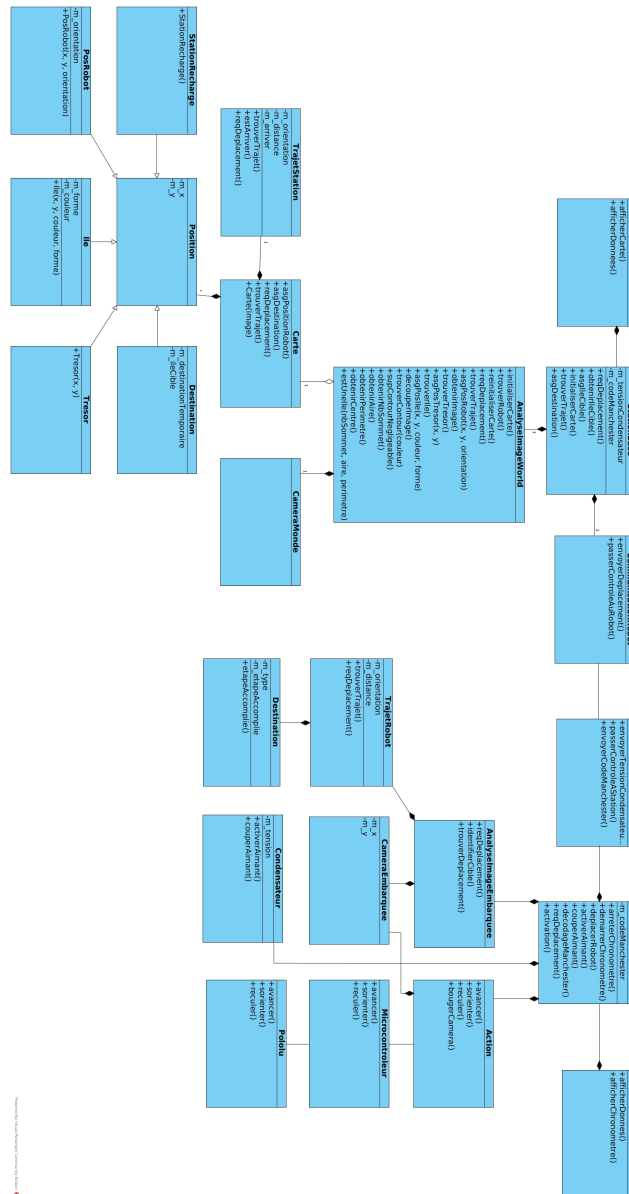


FIGURE 1.2 – Diagramme de classes

La figure 1.2 représente le diagramme de classes suite à la première itération. La structure est susceptible de changer suite aux prochaines itérations, mais voici une brève description de l'architecture actuelle de l'aspect logiciel du projet.

La section de gauche du diagramme sera implémentée sur la station de base, tandis que la section de droite sera implémentée sur le robot. Ces deux système pourront communiquer entre eux à l'aide des classes `CommunicationRobot` et `CommunicationStationBase`.

En ce qui concerne la station de base, le contrôleur du système est représenté par la classe `StationBase`. La classe `AnalyseImageWorld` analyse les images reçues de la `CameraMonde` et génère une carte schématique de la table (`Carte`) à l'aide d'imagerie. La carte est composée de divers éléments qui héritent tous de la classe `Position`. Les trajectoires du robot seront calculées dans la classe `TrajetStation` à l'aide des informations de la classe `Carte`.

Pour ce qui est du robot, il est aussi composé d'un contrôleur (`Robot`). Les demandes de mouvement que devra effectuer le robot passeront toutes par la classe `Action`, qui les acheminera au microcontrôleur Arduino Mega 2560. Lorsque le robot sera près de la destination, `TrajetRobot` calculera les trajets (à l'aide de `Destination` et d'imagerie effectuée dans `AnalyseImageEmbarquer`).

1.3 Diagramme de séquences

Les diagrammes de séquences suivants présentent les différents appels entre les objets du système lors d'un événement précis.

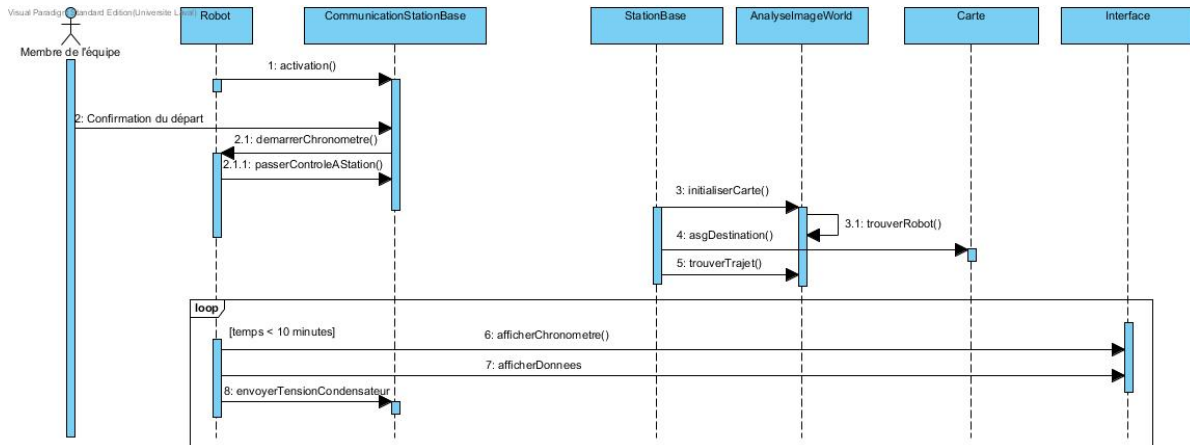


FIGURE 1.3 – Démarrage du Robot

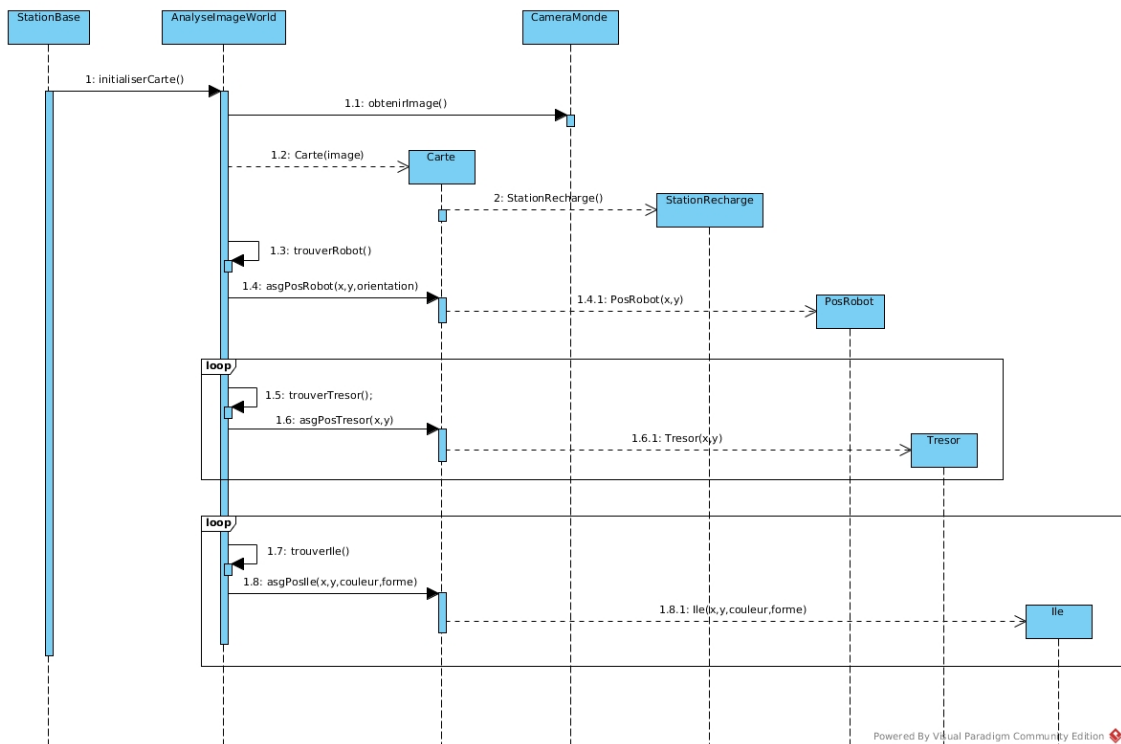


FIGURE 1.4 – Initialisation de la carte de jeu

Chapitre 2

Description des cas d'utilisation

2.1 Diagramme des cas d'utilisation

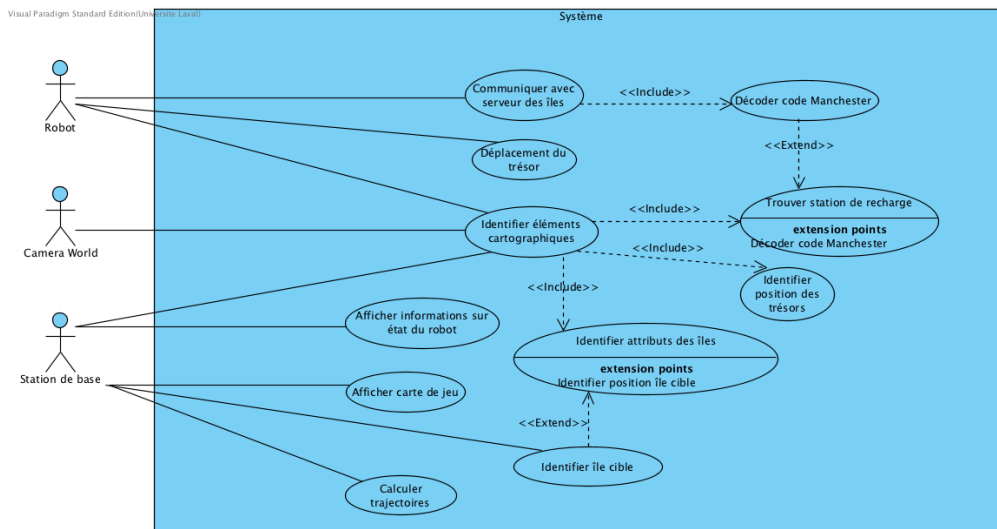


FIGURE 2.1 – Diagramme des cas d'utilisation

2.2 Scénarios des cas d'utilisation

Cas d'utilisation	Identifier les éléments cartographiques
Système	Système de reconnaissance visuelle
Acteurs	Robot, caméra world, station de base
Partie prenante et intérêt	Station de base : Vouloir identifier les éléments présents sur la carte afin que le robot puisse remplir son mandat sans entrer en collision avec quoi que ce soit.
Préconditions	Le robot et la caméra world doivent fournir des images à la station de base
Garantie en cas de succès	Les éléments cartographiques sont affichés sur la carte de la station de base
Scénario principal	<ol style="list-style-type: none"> 1. La caméra world envoie des images à la station de base 2. La surface de jeu, les îles, le robot, ainsi que les trésors visibles sont identifiés par le système de reconnaissance visuelle 3. Le robot envoie des images à la station de base 4. Le système de reconnaissance visuelle analyse ces images pour valider les positions des trésors
Scénarios alternatifs	Aucun scénario alternatif

TABLEAU 2.1 – Cas d'utilisation - Identifier les éléments cartographiques

Cas d'utilisation	Trouver station de recharge
Système	Système de reconnaissance visuelle
Acteurs	Robot, caméra world, station de base
Partie prenante et intérêt	Robot : Faire le plein d'énergie et recevoir le code Manchester
Préconditions	La station de base a identifié la position de la station de recharge et une trajectoire a été calculée
Garantie en cas de succès	La trajectoire menant le robot jusqu'à la station de recharge est affichée sur la carte de la station de base. Le robot se déplace vers la station de recharge sans toucher à aucun obstacle.
Scénario principal	<p>1. La caméra world et le robot envoient des images à la station de base</p> <p>3. Le système de reconnaissance visuelle analyse ces images et détermine l'emplacement de la station de recharge</p> <p>4. Un itinéraire vers la station de recharge peut être calculé</p>
Scénarios alternatifs	<p>3-1. Le système de reconnaissance visuelle analyse ces images et ne peut déterminer l'emplacement de la station de recharge</p> <p>4. La station de base ordonne un déplacement aléatoire au robot</p> <p>5. Le robot se déplace et renvoie des images à la station de base</p>

TABLEAU 2.2 – Cas d'utilisation - Trouver la station de recharge

Cas d'utilisation	Identifier la position des trésors
Système	Système de reconnaissance visuelle
Acteurs	Robot, caméra world, station de base
Partie prenante et intérêt	Robot : Identifier les positions des trésors cachés sur la table de jeu, afin de se déplacer jusqu'à l'un d'eux et le rapatrier sur l'île cible.
Préconditions	La station de base reçoit des images venant de la caméra world ainsi que celle du robot
Garantie en cas de succès	Les trésors sont identifiés sur la carte de la station de base et un trésor cible est choisi parmi ceux-ci. Le robot se dirigera vers ce trésor cible pour le déplacer jusqu'à l'île cible.
Scénario principal	<p>1. La caméra world et le robot envoient des images à la station de base</p> <p>2. Les trésors visibles sont identifiés et une trajectoire est calculée</p> <p>3. Le robot se déplace jusqu'au trésor</p>
Scénarios alternatifs	<p>2-1. Aucun trésor n'est identifié</p> <p>3. La station de base ordonne un déplacement aléatoire au robot</p> <p>4. Le robot se déplace et renvoie des images à la station de base</p>

TABLEAU 2.3 – Cas d'utilisation - Identifier la position des trésors

Cas d'utilisation	Identifier les attributs des îles
Système	Système de reconnaissance visuelle
Acteurs	Robot, caméra world, station de base
Partie prenante et intérêt	Robot : Identifier la couleur, la forme ainsi que la position des îles pour calculer une trajectoire valide.
Préconditions	La station de base reçoit des images venant de la caméra world
Garantie en cas de succès	La couleur, la forme ainsi que la position des îles sont affichées sur la carte de la station de base. Une trajectoire vers un trésor peut par la suite être calculée.
Scénario principal	<p>1. La caméra world envoie des images à la station de base</p> <p>2. La couleur, la forme et la position des îles sont identifiées par le système de reconnaissance visuelle</p>
Scénarios alternatifs	Aucun scénario alternatif

TABLEAU 2.4 – Cas d'utilisation - Identifier les attributs des îles

Cas d'utilisation	Identifier la position de l'île cible
Système	Système de reconnaissance visuelle
Acteurs	Robot, caméra world, station de base
Partie prenante et intérêt	Robot : Une fois en possession d'un trésor, le robot déplacera le trésor jusqu'à l'île cible et déposera le trésor sur celle-ci.
Préconditions	Le code Manchester contenant l'information sur l'île cible a été décodé et transmis à la station de base. Les attributs des îles ont été identifiés avec succès.
Garantie en cas de succès	Une fois la récupération d'un trésor terminée, une trajectoire vers l'île cible peut être calculée.
Scénario principal	<p>1. L'information envoyé par le serveur des îles est transmise à la station de base par le robot</p> <p>2. L'emplacement de l'île cible est identifié par le système de reconnaissance visuelle</p>
Scénarios alternatifs	Aucun scénario alternatif, l'identification de l'île cible doit toujours être réussie

TABLEAU 2.5 – Cas d'utilisation - Identifier la position de l'île cible

Cas d'utilisation	Décoder le code Manchester
Système	Ordinateur embarqué
Acteurs	Robot, caméra world, station de base
Partie prenante et intérêt	Robot : Une fois le code reçu, il sera décodé et une requête au serveur des îles sera effectuée.
Préconditions	Le robot doit s'être rendu à la station de recharge, où le transfert du code Manchester a lieu.
Garantie en cas de succès	Le message contenant l'information sur l'île cible sera par la suite envoyé à la station de base pour déterminer l'itinéraire à emprunter une fois en possession d'un trésor.
Scénario principal	<p>1. Le robot se déplace jusqu'à la station de base en suivant l'itinéraire calculé par la station de base</p> <p>2. Le code Manchester est transmis au robot pendant la recharge du condensateur</p> <p>3. Le robot décode le signal reçu</p> <p>4. Le robot effectue une requête au serveur des îles</p>
Scénarios alternatifs	Aucun scénario alternatif, le décodage du code Manchester doit toujours être réussi

TABLEAU 2.6 – Cas d'utilisation - Décoder le code Manchester

Cas d'utilisation	Communiquer avec le serveur des îles
Système	Ordinateur embarqué
Acteurs	Robot
Partie prenante et intérêt	Robot : Une fois la requête effectuée, l'information sur l'île cible est transmise à la station de base.
Préconditions	Le code Manchester doit avoir été décodé.
Garantie en cas de succès	La station de base est en mesure de déterminer la position de l'île cible et une fois la récupération d'un trésor terminée, une trajectoire vers l'île cible peut être calculée.
Scénario principal	<p>1. Le robot effectue une requête au serveur des îles en utilisant le message décodé</p> <p>2. Le serveur renvoie un message contenant l'information sur l'île cible</p> <p>3. Cette information est transmise à la station de base</p>
Scénarios alternatifs	Aucun scénario alternatif

TABLEAU 2.7 – Cas d'utilisation - Communiquer avec le serveur des îles

Cas d'utilisation	Déplacer le trésor
Système	Système de reconnaissance visuelle
Acteurs	Robot
Partie prenante et intérêt	Robot : Se déplacer vers un trésor, le récupérer et le déposer sur une île cible afin de respecter le but du projet.
Préconditions	Le robot doit avoir fait le plein d'énergie à la station de recharge afin de soulever le trésor. L'île cible doit avoir été identifiée et une trajectoire valide doit être calculée pour se rendre jusqu'au trésor. Par la suite, une seconde trajectoire est calculée pour déposer le trésor sur l'île cible.
Garantie en cas de succès	La station de base est en mesure de déterminer la position de l'île cible et une fois la récupération d'un trésor terminée, une trajectoire vers l'île cible peut être calculée.
Scénario principal	<p>1. Le robot se déplace jusqu'à un trésor en suivant une trajectoire préalablement définie</p> <p>2. Le robot se positionne et soulève le trésor</p> <p>3. Le robot se déplace jusqu'à l'île cible</p> <p>4. Le robot se positionne et dépose le trésor au centre de l'île cible</p>
Scénarios alternatifs	<p>2-1. Le robot se positionne, mais ne peut soulever le trésor</p> <p>3. Le robot se déplace jusqu'à la station de recharge</p> <p>Le robot recharge le condensateur</p>

TABLEAU 2.8 – Cas d'utilisation - Déplacer le trésor

Cas d'utilisation	Afficher les informations sur l'état du robot
Système	Station de base
Acteurs	Station de base
Partie prenante et intérêt	Station de base : Les informations critiques du robot peuvent être étudiées en temps réel.
Préconditions	Une communication doit être établie entre le robot et la station de base afin d'échanger ces informations.
Garantie en cas de succès	Divers informations sont affichées sur la station de base.
Scénario principal	<p>1. Le robot se déplace et communique avec la station de base</p> <p>2. La tension du condensateur est affichée sur la station de base</p> <p>3. La vitesse de chaque moteur est affichée sur la station de base</p>
Scénarios alternatifs	<p>1-1. Le robot décode le code Manchester</p> <p>2. La lettre ASCII décodée est affichée sur la station de base</p> <p>3. Une requête est envoyée au serveur des îles</p> <p>4. La forme et la couleur de l'île cible sont affichées sur la station de base</p>

TABLEAU 2.9 – Cas d'utilisation - Afficher les informations sur l'état du robot

Cas d'utilisation	Afficher la carte de jeu
Système	Système de reconnaissance visuelle
Acteurs	Robot, caméra world, station de base
Partie prenante et intérêt	Étudiants : Permet d'avoir une rétroaction rapide des résultats calculés par le système de reconnaissance visuelle.
Préconditions	La station de base doit recevoir des images provenant de la caméra world et celle du robot.
Garantie en cas de succès	Les îles, les trésors, le robot ainsi que les itinéraires sont affichés sur la carte en temps réel.
Scénario principal	<p>1. La caméra world envoie des images à la station de base</p> <p>2. La surface de jeu, les îles, le robot, ainsi que les trésors visibles sont affichés sur la carte de jeu en temps réel</p>
Scénarios alternatifs	Aucun scénario alternatif

TABLEAU 2.10 – Cas d'utilisation - Afficher la carte de jeu

Cas d'utilisation	Calculer les trajectoires
Système	Système de reconnaissance visuelle
Acteurs	Robot, caméra world, station de base
Partie prenante et intérêt	Robot : Le robot doit se diriger en suivant une trajectoire prédéfinie, sans jamais entrer en contact avec quoi que ce soit.
Préconditions	La station de base doit recevoir des images provenant de la caméra world et celle du robot. La position de la destination doit être connue. Les destinations possibles sont la station de recharge, un trésor ou l'île cible.
Garantie en cas de succès	Le robot se déplace en suivant l'itinéraire sans touché à un obstacle.
Scénario principal	<ol style="list-style-type: none"> 1. Une destination est choisie 2. Le système de reconnaissance visuelle analyse la carte de jeu et repère le robot 3. Le système analyse les obstacles présents sur la carte 4. Une trajectoire est calculée et transmise au robot
Scénarios alternatifs	Aucun scénario alternatif

TABLEAU 2.11 – Cas d'utilisation - Calculer les trajectoires

Chapitre 3

Familiarisation avec équipements et expériences préliminaires

3.1 Structure mécanique

La base mécanique du robot est assurée par les différentes plate-formes et tiges filetées fournies pour le projet, il en revient donc à nous de décider de l'agencement de ces composantes afin de produire un robot mécaniquement efficace. De manière à bien modéliser le robot et ainsi faciliter l'installation de futurs équipements, il fut décidé de réaliser une représentation 3D du prototype. La figure 3.1 présente le dessin ainsi réalisé, bien que grossier il fut construit à partir de mesures exactes effectuées sur le robot. Ce dessin est très utile pour planifier d'avance l'emplacement des différents systèmes qui seront ajoutés au cours du projet en plus de permettre une meilleure planification de l'organisation des circuits électroniques.

On peut remarquer que sur ce dessin l'ordinateur embarqué est situé sur la plate-forme centrale, ce choix est justifié par le fait que nous désirons avoir l'électronique le plus proche possible des moteurs DC afin de limiter la longueur des fils. La batterie LiPO représentée par le bloc bleu est positionnée sur la plate-forme la plus basse de manière à assurer un centre de gravité assez bas. Le bloc jaune représente environ les dimensions des circuits électroniques présents sur cet étage du robot. Bien évidemment, toutes ces composantes seront fixées en place. Bien que le pont en H n'est pas observable sur cette image il est important de noter que celui-ci peut être fixé sous la base en cas de besoin. Le préhenseur présenté à la section 3.2 est représenté sur le devant du robot, la caméra embarquée est installée directement au-dessus de ce dispositif sur l'étage le plus haut. L'utilisation de ce dessin 3D et d'un autre dessin de la borne de rechargement a aussi permis de mesurer, sans faire de prototype de préhenseur, la hauteur adéquate de la bobine de chargement du condensateur afin de bien aligner les deux dispositifs.

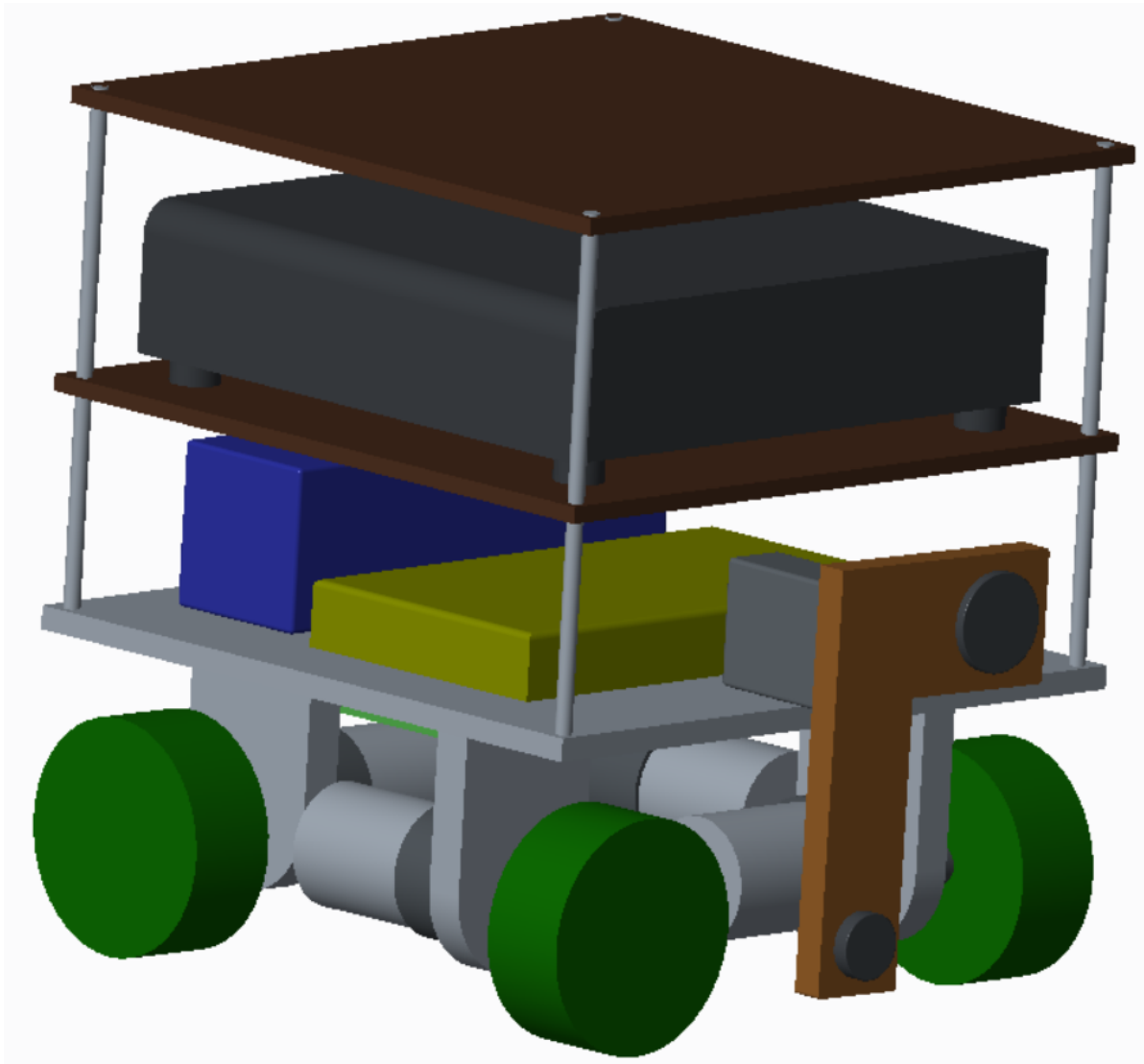


FIGURE 3.1 – Représentation 3D du robot

3.2 Système de préhenseur et d'électroaimant

Afin de rendre le préhenseur le plus simple possible mécaniquement, nous choisissons de le faire en forme d'équerre. Aux deux extrémités de l'équerre sont situés l'électroaimant et le secondaire du transformateur qui permet de charger le condensateur d'alimentation de l'électroaimant. Il devient donc très aisé d'aligner le système de recharge avec la station de recharge ainsi que de soulever les trésors en faisant tourner l'équerre de 90 degrés dans un sens ou dans l'autre. Cette rotation est assurée par un servomoteur situé au centre de l'équerre. La figure 3.1 est une représentation du système mécanique du préhenseur. Le dessin 3D permet de se faire une bonne idée des dimensions du bras du préhenseur de manière à ce qu'il soit efficace. Un servomoteur présentant suffisamment de couple ($6.4Kg \cdot cm$ à 6V) permet d'actionner le préhenseur.

L'électroaimant est un modèle pré-fait acheté en ligne. Il s'agit du modèle *Grove* de la compagnie *seeed* (voir section 3.10). Cet aimant peut soulever une charge de 1kg pour un courant de 400mA. Ce système vient aussi avec un système de contrôle réalisé à partir d'un transistor de commutation. En assumant que la force générée par l'aimant dépend quadratiquement du courant le parcourant et en sachant que le poids des trésors est de 30g l'unité, on estime le courant nécessaire pour soulever les trésors autour de 5mA. Pour avoir une marge de sécurité ainsi que de pouvoir attirer les trésors «à distance», on augmente ce courant à 20mA. Le courant dans l'électroaimant est contrôlé par une source de courant à diode Zener (voir figure 3.2).

L'énergie nécessaire pour soulever un trésor pendant 10 minutes par l'électroaimant est de 60J ($U = 5V \times 20mA \times 10min \times 60s/min$). Un condensateur de 0.5F à 5V contient 62.5J d'énergie ($U = 0.5 \times C \times V^2$). C'est donc cette valeur de condensateur qui est choisie pour alimenter l'électroaimant.

Afin de permettre de faire des tests ainsi que de s'assurer un degré de sécurité, un condensateur de 1F est également considéré dans le design de ce système.

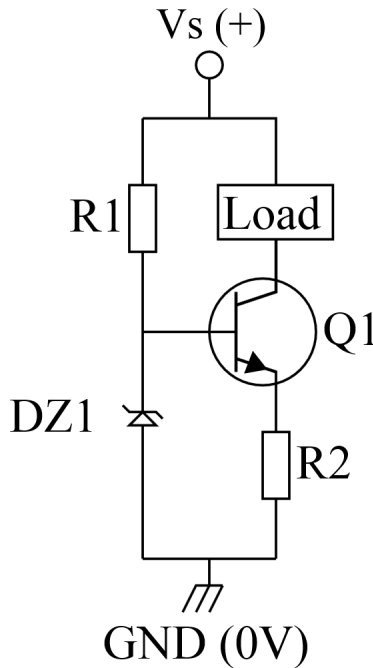


FIGURE 3.2 – Source courant à diode Zener

3.3 Contrôle de la caméra

La caméra embarquée est contrôlée par les servomoteurs fournis qui permettent à la caméra un déplacement sur son axe horizontal et vertical afin d'augmenter le champ de vision du robot. De manière à contrôler le *Pololu Maestro*, des commandes sont envoyées par USB à partir de l'ordinateur embarqué situé sur le robot. Le Pololu est facilement contrôlable par du code en langage *C* géré par l'ordinateur embarqué. De manière à se familiariser avec le fonctionnement de ce système nous utilisons le *Maestro Control Center* afin de voir quelles sont les limites du champ de vision de cette caméra. Sur la figure 3.3 on voit que le robot peut facilement voir le sol directement devant lui, ceci est utile afin de repérer les trésors, les îles et même la station de recharge à des distances très courtes. On remarque aussi qu'il est possible de fixer des limites logicielles de mouvement pour les différents servomoteurs, une fonctionnalité qui facilitera l'implantation du contrôle de la caméra.

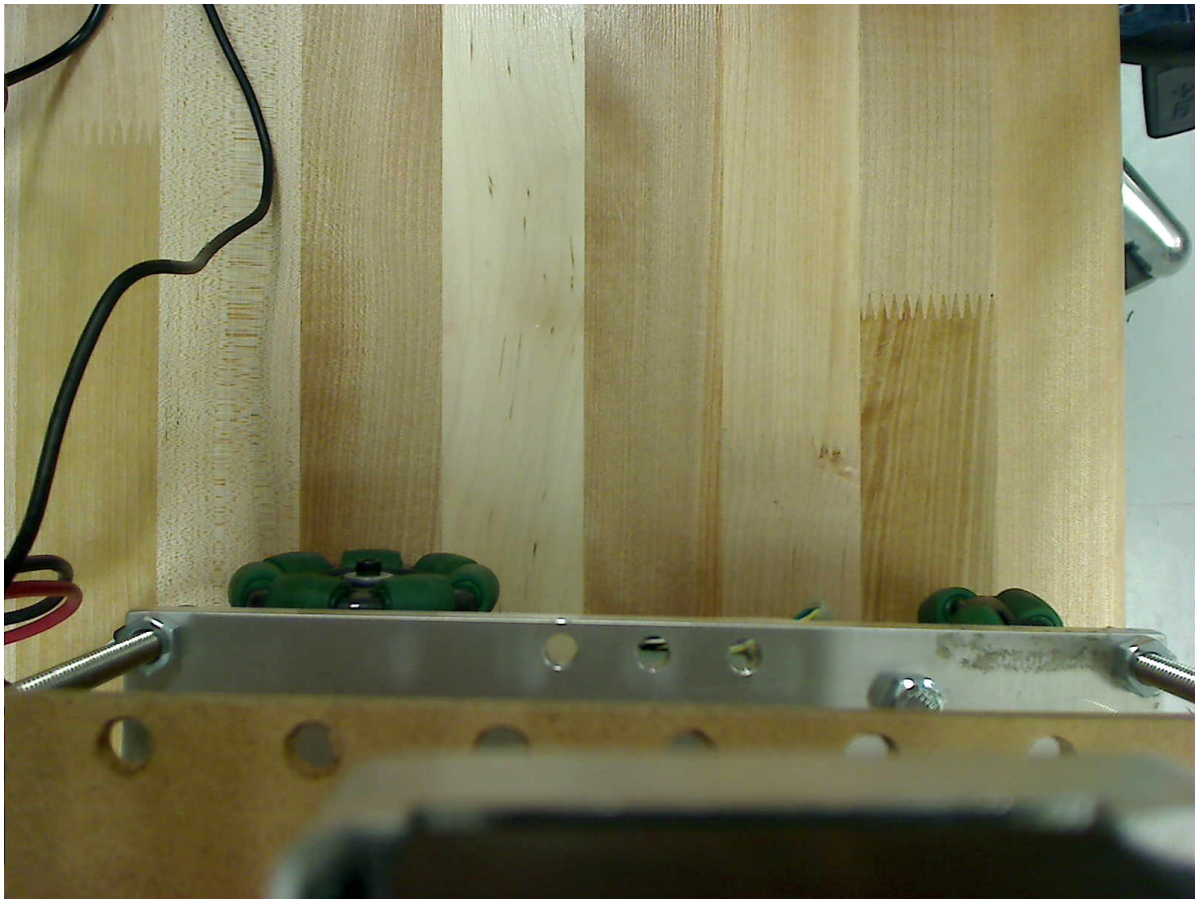


FIGURE 3.3 – Vision au sol du robot à partir de la caméra embarquée

3.4 Contrôle des moteurs

Afin de commander les moteurs du robot, il est nécessaire d'avoir une routine d'asservissement. Pour envoyer les commandes aux moteurs DC, l'Arduino Mega est choisi. Celui-ci possède une vitesse d'horloge de 16 MHz et 256kB de mémoire flash pour 45\$ [2], ce qui est un bon équilibre performance/prix pour les besoins du projet.

Les moteurs utilisés pour entraîner les roues du robot possèdent 6400 valeurs de position par rotation. Si une interruption est effectuée à chaque modification de cette valeur, on risque d'empêcher l'exécution complète d'une boucle d'asservissement entre deux interruptions. Donc, si une interruption est levée à chaque modification de la position, il faut s'assurer que le calcul de la vitesse de rotation se fasse après un nombre multiple d'interruptions, avec un compteur de temps.

Pour ce qui a trait à la communication avec les moteurs DC, un shield Adafruit se connectant directement sur l'Arduino Mega est utilisé. Avec celui-ci, il est possible de générer des ondes modulées (PWM) servant à contrôler les quatre roues individuellement, avec chacune une résolution de 8 bits et jusqu'à 1.2A par canal[3]. Cette division de la commande est utile lors d'un déplacement en diagonale, et pourrait également être utile pour continuer le fonctionnement dans le cas d'une roue défectueuse. Ce dispositif est utilisé pour contrôler les moteurs pour les premiers tests, bien évidemment l'utilisation du pont en H fournit fera aussi l'objet de tests lorsque la batterie et l'alimentation seront disponibles.

En résumé, l'ordinateur envoie des instructions de direction au microcontrôleur Arduino, qui exécute alors une routine de déplacement. Celle-ci est basée sur un asservissement de vitesse, déterminée par des interruptions lors de la rotation des servomoteurs. Les commandes de l'Arduino sont alors communiquées aux roues à l'aide d'un shield Adafruit permettant de leur délivrer de la puissance.

3.5 Localisation du robot et des îles

La caméra Logitech C905 située en hauteur permet de visionner une grande partie de la table. Celle-ci sera branchée par USB à la station de base. Les tests effectués jusqu'à présent nous ont permis de contempler la vision limitée de la table de jeu (figure 3.4). C'est donc pour cette raison que seule une première approximation sur la position des trésors pourra être faite. La localisation plus précise de ceux-ci se fera par la caméra embarquée (idem pour la station de recharge). Voir la section 3.7.

La détection des îles et du robot se fait par la station de base à l'aide de la caméra monde. Pour ce faire, le logiciel OpenCV est utilisé. Premièrement, l'image est découpée de sorte qu'elle ne contienne que la table de jeu. Cet outil très puissant nous permet aussi de détecter les contours d'objets à l'intérieur d'image en utilisant les couleurs pour les distinguer (BGR). Les contours jaunes, bleus, rouges et verts sont détectés afin de repérer les îles. Afin d'éviter les erreurs, une relation entre le nombre de coins, le périmètre et l'aire des contours sera vérifiée. La validation de cette relation confirmera la présence d'une île tout en indiquant

sa géométrie et sa couleur (selon le nombre de coins et l'intervalle BGR utilisé). Nous avons déjà testé ces étapes et elles semblent plutôt efficaces (figure 3.5). Pour localiser le robot, une forme et une couleur particulière seront placées au dessus du robot. Nous procéderons de façon similaire pour le détecter.

3.6 Calcul de la trajectoire

Cette étape est basée sur des recherches, mais n'a pas encore été testée. Nous prévoyons détecter les coordonnées pixels du milieu de chaque île (à l'aide des contours et d'une fonction préétablie de OpenCV). Une simple échelle pourra ensuite convertir les coordonnées pixels en coordonnées réelles.

La planification de la trajectoire sera faite avec un algorithme Dijkstra. Un graphe sera créé (représenté par une grille de points ayant une distance à déterminer entre eux). Évidemment, les points situés à proximité des îles seront exclus du graphe. À l'aide des informations ci-dessus, l'algorithme pourra trouver le plus court chemin jusqu'à la destination.

3.7 Repérage des trésors et de la station de recharge

La caméra Logitech C905 située sur le robot permet de repérer les différents trésors ainsi que la station de recharge. Celle-ci est reliée directement à l'ordinateur embarquée afin d'être alimentée et de fournir les images captées. La fréquence de captation des images reste à déterminer puisqu'il faudra décider à quel intervalle nous devons mettre à jour la vision du robot. L'envoi des commandes servant à contrôler la prise d'image par la caméra sera effectuée par la librairie Pygame de Python.

Afin de détecter les trésors, une première approximation de leur position est effectuée par la caméra monde qui, à l'aide de la librairie cv2 de OpenCV, permet de localiser dans une image un intervalle de couleur BGR. Le choix de la librairie d'OpenCV est justifié par le fait qu'elle possède toutes les fonctions nécessaires à un programme de vision complet et qu'elle s'intègre facilement au reste du code en Python. Les tests préliminaires effectuées avec la caméra monde ont permis de venir à la conclusion que la détection des trésors s'effectuent très bien. Par contre, le premier test de la prise de photo a également permis de constater que la caméra monde ne voit pas le fond de la table et donc, certains trésors ne seront pas détecter par celle-ci, justifiant également la détection des trésors par la caméra embarquée.

Comme mentionné précédemment, les premiers tests effectués afin de détecter les trésors ont démontré qu'il était possible de bien repérer les trésors à partir de la caméra monde comme la figure 3.5 le montre.

Par la suite, les différents pixels correspondant à la couleur des trésors sont placés dans un masque des trésors. Grâce à la position relative de ces points dans le masque des trésors, il est possible d'avoir une position approximative de ces trésors dans la carte virtuelle. Afin de confirmer la détection des trésors ou pour repérer les autres qui seront hors du champ

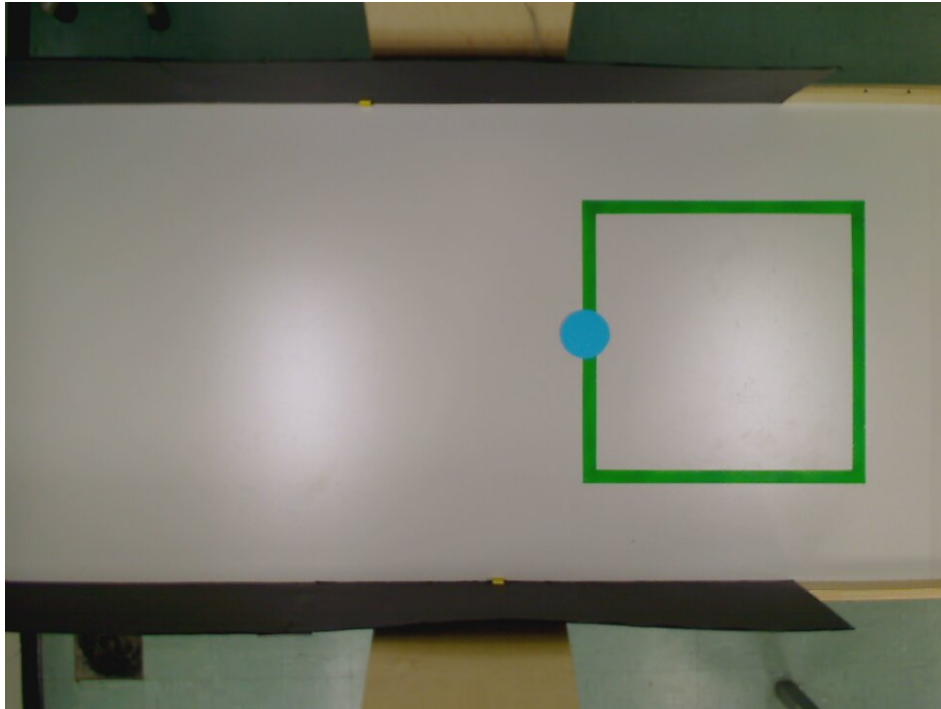


FIGURE 3.4 – Photo test de la caméra monde

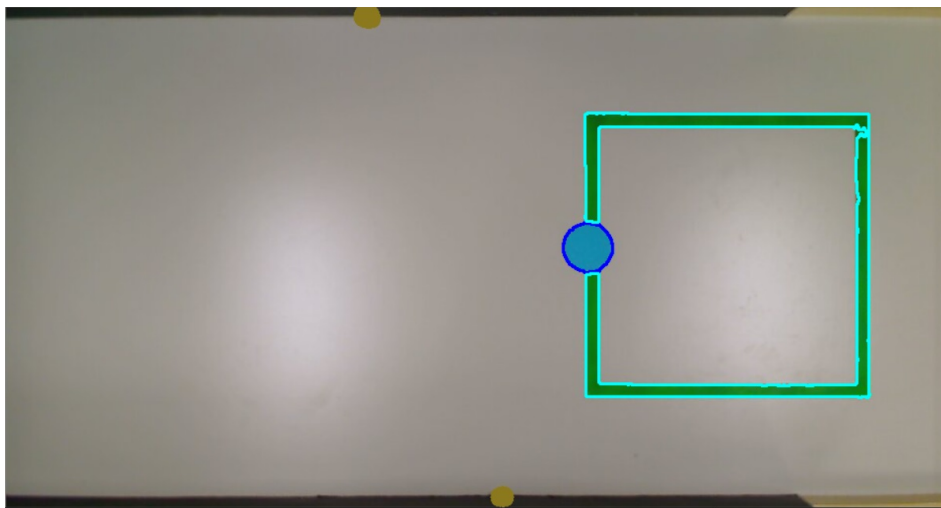


FIGURE 3.5 – Photo test après l'application des masques de détection

de vision du robot, la même opération de détection des couleurs est effectuée ensuite par la caméra embarquée autour des coordonnées approximative détectée par la caméra monde.

La station de recharge, quant à elle, est marquée d'une couleur caractéristique lui permettant de se distinguer du reste du décor. Comme la position et l'orientation du robot sont connues en tout temps et que la station de recharge est toujours située au même endroit, la détection de celle-ci est assez simple. Comme le robot peut être placé n'importe où au départ sur la table, la caméra monde est chargée à l'initialisation du programme de détecter la position et l'orientation de celui-ci. Afin de réaliser cette tâche, un agencement de couleur distinctif est placé sur le dessus du robot afin d'indiquer l'orientation ainsi que la position de celui-ci et sera détecté par notre programme de vision. Le robot peut ensuite se diriger vers les coordonnées fixes de la station de recharge et la repérer facilement par la suite.

3.8 Alimentation du robot

Pour alimenter adéquatement tous les systèmes présents sur le robot, on doit avoir une batterie avec une tension qui se situe entre 21V et 30V pour alimenter le régulateur de l'ordinateur embarqué, qui fonctionne à 19V avec un courant de 3.5A. On doit également avoir assez de puissance pour que les moteurs et l'électronique de contrôle puissent fonctionner pendant au moins dix minutes. La puissance dont le robot a besoin se définit surtout par celle des moteurs, des servomoteurs et de l'ordinateur embarqué. Les moteurs demandent au maximum 800mA à 12V lorsque le rotor est complètement bloqué et les servomoteurs qui servent à contrôler la caméra demandent environ 30mA à 5V lorsqu'ils sont sollicités, soit un court instant. On utilise également un servomoteur plus puissant pour mouvoir le préhenseur, celui-ci risque de consommer plus de puissance à cause de l'effort plus important qui sera fourni. En additionnant la puissance de ces systèmes, la puissance requise est d'environ 110W dans le pire des cas. Nous avons donc choisi une batterie LiPO 6S de 4 500mA, ce qui peut donner 27A, pour un total de 599W en 10 minutes. Les six cellules sont nécessaires pour avoir 22.2V, ce qui permet d'obtenir le 19V requis pour l'ordinateur embarqué. Le 4 500mA est justifié, car on veut pouvoir travailler sur le robot plus longtemps que 10 minutes à des fins de test. Nous avons calculé que cette batterie pourrait nous donner environ 50 minutes d'autonomie.

L'utilisation d'une batterie LiPO nécessite un chargeur intelligent qui refuse de charger la batterie si les cellules ont une tension trop faible, car la batterie pourrait exploser. Nous avons donc acheté un tel chargeur et on utilise également un sac de protection pour diminuer l'impact d'une éventuelle explosion. L'utilisation d'une LiPO requiert également un système qui déclenche une alarme sonore lorsqu'il faut recharger la batterie, pour justement empêcher les risques d'explosions. Cette fonction est assurée par un petit afficheur de tension, conçu spécialement pour les batteries LiPO, qui sonne quand la tension des cellules est trop faible.

Le robot a donc besoin de trois niveaux de tension différents pour fonctionner. On utilise des régulateurs DC-DC pour avoir une tension stable, aux valeurs désirées. Le régulateur

de l'ordinateur embarqué est fourni. Pour le 12V et le 5V, nous avons choisi de prendre trois fois le même régulateur, qui prend 4V à 38V en entrée et peut fournir 1.25V à 36V en sortie (voir liste des pièces 3.10. Il y aura donc deux régulateurs pour le 12V, pour séparer l'alimentation des moteurs et celle du Arduino, et un régulateur pour le 5V. Celui-ci peut fournir un maximum de 5A et il est doté d'un afficheur de tension pour connaître facilement la valeur réglée en sortie. On protège nos systèmes avec des fusibles pour s'assurer de ne rien briser. Nous avons choisi un fusible de 10A directement à la sortie de la batterie pour protéger l'ensemble du système. Pour les moteurs, on utilise un fusible de 5V, ensuite un fusible de 2A pour l'alimentation de l'Arduino et enfin un fusible de 3A pour les circuits électroniques du 5V. Un interrupteur situé directement suite à la batterie permet finalement d'actionner l'ensemble du système. La figure 3.6 présente un diagramme en bloc de l'alimentation.

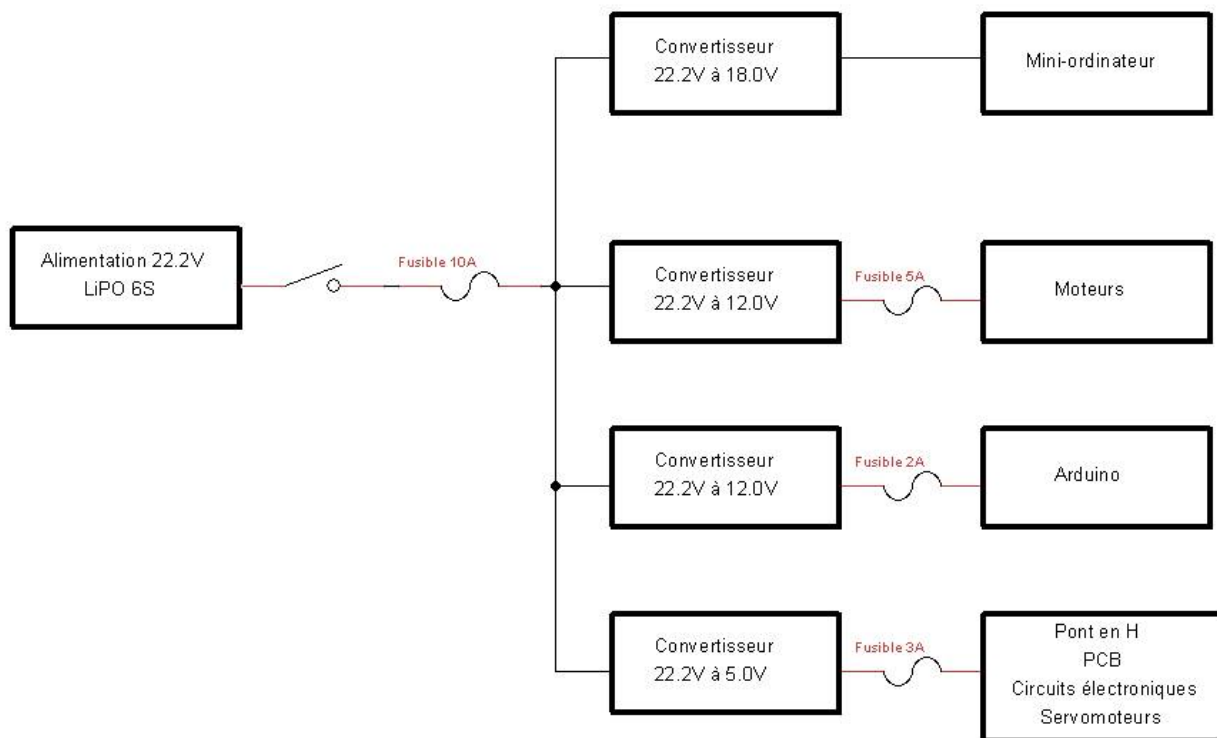


FIGURE 3.6 – Diagramme en bloc de l'alimentation

3.9 Transmission du code Manchester

La transmission sans-fil du code Manchester est assurée par une combinaison LED-photorécepteur. La LED est située sur le dessus de la station de recharge et le photorécepteur est situé sur le dessus du robot. Ce système de communication sans-fil est limité à des distances relativement faibles, mais le robot étant en contact physique avec la station de recharge, cette contrainte ne pose pas de problème. Le dessin 3D présenté précédemment

servira à planifier la disposition exacte de ces composantes une fois les pièces reçues et le concept testé de manière à minimiser les temps de construction et d'assemblage.

La LED est modulé par un microcontrôleur situé sur la station de recharge. Le modèle du microcontrôleur reste à déterminer, mais devra être assez simple et peu coûteux puisque sa tâche sera relativement simple. Ce système de communication est illustré sur le schéma 3.7.

Le document [1] explique brièvement comment réaliser un tel système et est utilisé comme référence pour réaliser notre système.

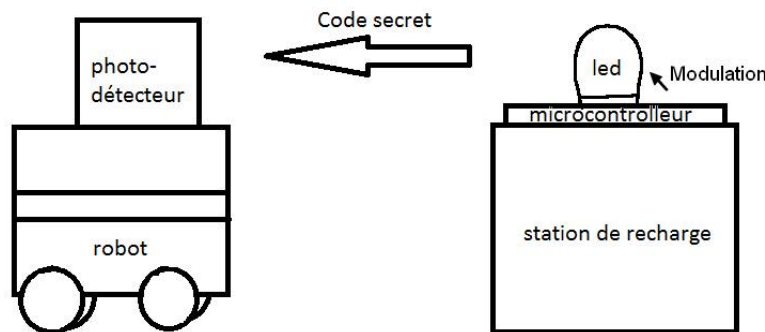


FIGURE 3.7 – Communication sans-fil entre la station de recharge et le robot

3.10 Liste des pièces achetées

Buck (x3) http://www.ebay.ca/itm/171445007919?_trksid=p2050601.m570.l6004&_trkparms=gh1g%3DI171445007919.N41.S2.R2.TR2

Batterie LiPo http://www.hobbyking.com/hobbyking/store/__10284__Turnigy_4500mAh_6S_30C_Lipo_Pack.html

Chargeur de batterie LiPO http://www.hobbyking.com/hobbyking/store/__40270__Hobbyking_8482_T682_AC_6s_10A_90W_Eco_Touch_Balance_Charger_Discharger.html

Alarme batterie http://www.hobbyking.com/hobbyking/store/__74024__HobbyKing_8482_Lipo_Voltage_Checker_2S_8S_.html

Arduino Mega <http://www.robotshop.com/ca/en/arduino-mega-2560-microcontroller-rev3.html>

Servomoteur de contrôle du préhenseur <http://www.robotshop.com/ca/en/hitec-hs-485hb-servo-motor.html>

Shield pour «driver» les moteurs <http://www.robotshop.com/ca/en/motor-shield-kit-arduino-v2.html>

Électroaimant Grove http://www.seeedstudio.com/wiki/Grove_-_Electromagnet

Bibliographie

- [1] ETH Zurich, *An LED-to-LED Visible Light Communication System with Software-Based Synchronization*, [En ligne], <http://people.inf.ethz.ch/schmist/papers/OWC12Slides.pdf>, Page consultée le 27 janvier 2016
- [2] Robotshop, *Arduino Mega 2560 Microcontroller*, [En ligne], <http://www.robotshop.com/ca/en/arduino-mega-2560-microcontroller-rev3.html>, Page consultée le 30 janvier 2016
- [3] Adafruit, *Adafruit Motor/Stepper/Servo Shield for Arduino v2 Kit - v2.3*, [En ligne], <https://www.adafruit.com/products/1438>, Page consultée le 30 janvier 2016