



## Design III

### Remise 4

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

17 avril 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
3.0	28 février 2016	Remise 2
4.0	3 avril 2016	Remise 3
5.0	17 avril 2016	Remise 4

# Table des matières

<b>1 Vision</b>	<b>1</b>
1.1 Localisation des îles et des trésors . . . . .	1
1.2 Localisation du robot . . . . .	1
1.3 Phases d'alignement du robot . . . . .	2
1.3.1 Capture du trésor . . . . .	3
1.3.2 Dépôt du trésor . . . . .	3
1.3.3 Recharge du condensateur . . . . .	3
<b>2 Schémas Électroniques</b>	<b>6</b>
2.1 Station de recharge . . . . .	6
2.2 Circuit de charge et de décharge du condensateur . . . . .	7
2.3 Circuit de connexions pour les moteurs . . . . .	8
2.3.1 Connexions moteurs et pont en H . . . . .	8
2.3.2 Connexions sur l'Arduino . . . . .	9
2.4 Circuit d'alimentation . . . . .	10
<b>3 Post-mortem</b>	<b>11</b>
3.1 Points forts du robot . . . . .	11
3.2 Améliorations potentielles . . . . .	12
<b>Bibliographie</b>	<b>15</b>

# Chapitre 1

## Vision

Les sections 1.1 et 1.2 se déroulent évidemment sur la station de base. Il est à noter que lors de la réception d'une nouvelle image, le programme commence par la redimensionner de sorte à ne contenir que la table. Par la suite, un *Gaussian blur* est effectué. C'est à ce stade que les détections sont effectuée.

### 1.1 Localisation des îles et des trésors

Le processus de localisation des îles est lancé au tout début de la routine. Pour localiser les îles, chacune des quatre couleurs est filtrée et placée dans un masque afin d'effectuer un traitement individuel. Ensuite, les formes ayant une aire trop grande ou trop petite sont éliminées. Par la suite, les formes possédant un trou (contour enfant) avec une aire considérable sont éliminées (voir 1.2). Le système compare ensuite les contours restant des formes filtrées avec les formes géométriques en mémoire. De ce fait, celui-ci est en mesure d'identifier la forme qui possède le plus haut taux de compatibilité avec celles en mémoire. Si l'indice de précision est trop faible pour les quatre formes en mémoire, la forme est ignorée. Bien que ce processus soit déjà robuste, ceci est effectuée à dix reprises dans le but de conserver la liste d'île ayant la longueur la plus constante.

Pour ce qui est de la localisation des trésors, un processus similaire est utilisé. L'image est filtrée (à dix reprises) avec la couleur des trésors, puis les formes détectées sont retenues ou non dépendamment de l'aire de celles-ci. Les formes des trésors ne sont donc pas comparées, mais les trésors ne se retrouvent pas sur le long des murs sont négligés. Cette dernière opération s'effectue facilement puisque l'image est redimensionnée de sorte à ne contenir que la table avant la détection.

### 1.2 Localisation du robot

Puisque la détection des îles se déroule avec succès, la même base est appliquée au robot. En effet, sur le dessus de celui-ci se retrouvent deux formes (carré et cercle) de la même

couleur située sur le même axe. Par contre, afin de les différencier des îles, elles possèdent un trou blanc à l'intérieur (voir figure 1.1). L'image redimensionnée de la table est donc filtrée avec la couleur du robot et placée dans un masque. Ensuite (comme pour les îles), les formes ayant une aire trop grande ou trop petite sont éliminées. Par la suite, les formes ne possédant pas un trou (contour enfant) avec une aire considérable sont éliminées. Le système compare ensuite les contours restant avec les formes géométriques en mémoire. De ce fait, celui-ci est en mesure d'identifier la forme qui possède le plus haut taux de compatibilité avec celles en mémoire. Si l'indice de précision est trop faible pour les deux formes en mémoire, le robot ne sera pas détecté à cette itération.



FIGURE 1.1 – Image de la forme sur le robot

Une fois les contours détectés, un peu de géométrie permet d'obtenir les données voulues. Les formes sont positionnées de sorte que le centre des deux formes est situé au centre du robot. L'orientation du robot est simplement l'orientation du vecteur reliant le centre des deux formes (sachant que le cercle est l'avant du robot).

Afin d'ajouter un peu de robustesse à ce procédé, les détections subséquentes considèrent l'ancienne position du robot et une analyse de faisabilité du déplacement est effectuée. Si le déplacement est trop élevé, le robot n'est pas détecté. Dans l'éventualité où le robot n'est pas détecté lors de dix itérations consécutives, le robot est déclaré comme perdu et l'analyse de faisabilité de déplacement est ignorée jusqu'à ce qu'il soit retrouvé.

### 1.3 Phases d'alignement du robot

Tout d'abord, il est important de spécifier que la station de base gère les déplacements principaux du robot alors que le robot gère lui-même les phases d'alignements. Ceci dit, lorsque le robot est dirigé vers une position cible, la station de base lui indiquera quel type d'alignement il doit effectuer. Il y a une phase d'alignement unique pour la capture d'un trésor, le dépôt de celui-ci sur l'île cible ainsi que pour l'amarrage du robot avec la station de recharge.

Les phases d'alignement regroupent plusieurs petites étapes. Tout commence avec le changement de l'orientation de la caméra embarquée. Celle-ci possède plusieurs positions prédefinies, ce qui facilite grandement le déroulement du processus d'alignement.

### 1.3.1 Capture du trésor

Pour la capture du trésor, la caméra est placée en "*position trésor*" afin d'identifier le trésor et évaluer la distance le séparant du robot. La distance est obtenue en comparant la dimension du trésor avec une dimension de référence établie lors de la phase de test. Le trésor est obtenu en filtrant l'image prise par la camera avec un filtre de couleur jaune. On effectue les traitements sur cette forme afin de calculer les ajustements nécessaires.

Ensuite, puisque la position du préhenseur sur le robot est connue, le robot calcule les ajustements nécessaires et les commandes appropriées sont calculées. Une fois la commande de déplacement latéral effectuée, l'électroaimant est activé et le préhenseur est abaissé. Le robot effectue par la suite un déplacement vers l'avant, capture le trésor, remonte le préhenseur et recule afin de valider que le trésor a bel et bien été capturé en effectuant une autre analyse d'image. La phase de préhension de trésor se termine donc et la station de base reprend le contrôle du robot.

### 1.3.2 Dépôt du trésor

Pour le dépôt du trésor, la caméra se positionne face à la surface de jeu et analyse la position de l'île cible pour calculer le déplacement à effectuer. Un filtrage par couleur est effectué afin de valider la couleur de l'île cible. Ensuite, le centre de cette forme est trouvée et le système effectue des calculs afin de déterminer les ajustements. Une fois ces ajustements calculés, des commandes de déplacements sont envoyées. Finalement, une fois l'alignement terminé, le système valide la position de l'île cible par rapport à la zone de dépôt sécuritaire déterminée par des tests. Le trésor est soit déposé ou un autre alignement est effectué. Pour déposer le trésor, l'électroaimant est activé, puis le préhenseur est abaissé en douceur. Une fois le préhenseur abaissé, l'électroaimant est désactivé et quelques secondes plus tard, le préhenseur remonte.

Un exemple de l'analyse effectuée par le système lors de la phase d'alignement avec l'île cible peut être observé à la figure 1.2. Le système calcule les ajustements nécessaires afin que le centre de l'île se trouve dans la zone de dépôt sécuritaire une fois ces ajustements effectués par le robot.

### 1.3.3 Recharge du condensateur

La phase d'alignement avec la station de recharge est une étape cruciale dans le déroulement de la routine. En effet, un mauvais alignement aura comme conséquence un plus long temps de recharge ou aucune recharge dans le pire des cas. Il est donc indispensable que le robot soit parfaitement aligné avec la station de recharge afin que la charge par induction

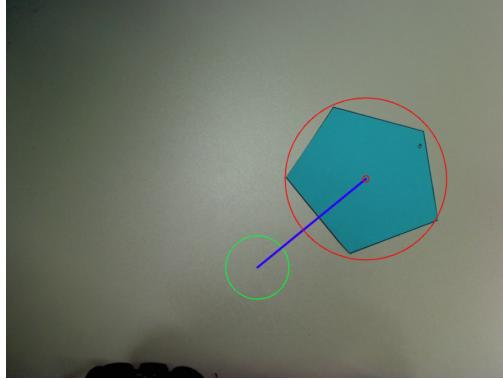


FIGURE 1.2 – Traitement fait lors de l’alignement avec l’île

soit optimale. Nous avons choisis de placer au-dessus de la station de recharge un carré bleu avec un plus petit carré orange à l’intérieur (voir figure 1.3).



FIGURE 1.3 – Image au dessus de la station de recharge

Lorsque le robot est mené à la station de recharge à l'aide de la station de base, celle-ci enverra une commande d'alignement au robot. La caméra se positionne face à la station de recharge. Une photo est prise et est filtrée avec un masque de couleur bleue. Le robot analyse par la suite la position du carré bleu (en éliminant les contours ayant une aire trop grande ou trop petite), qui se trouve au dessus de la bobine d'induction. Suite à l'analyse de l'image, l'aire de cette forme est comparée à une aire de référence établie à une distance connue afin de déterminer la distance séparant les deux objets. Un ajustement latéral est aussi calculé selon la position du centre de la forme cible. L'ajustement latéral est donc effectué suivis de l'ajustement en ligne droite. Par contre, on soustrait quelque centimètre au dernier ajustement. Ceci nous permet de prendre une nouvelle photo très près de la station de recharge. On refait donc les même étape, mais avec un filtre de couleur orange pour détecter le carré orange. Les alignements calculés par ce dernier représentent les alignements finaux effectué par le robot.

Précédant l'exécution des ajustements, la charge du condensateur est activée. La tension

du condensateur est échantillonnée tout au long de la recharge, car dès que celle-ci dépasse 4.7V, on désactive la charge du condensateur et le robot se sépare par la suite de la station de recharge en reculant de quelques centimètres.

# Chapitre 2

## Schémas Électroniques

### 2.1 Station de recharge

La figure 2.1 présente le circuit installé sur la station de recharge. le PIC est utilisé pour générer une horloge nécessaire au Manchester et une autre pour le hachage du signal de charge. Les signaux d'horloge ainsi que le code Manchester sont transmis au module Xbee pour ensuite être transmis au robot.

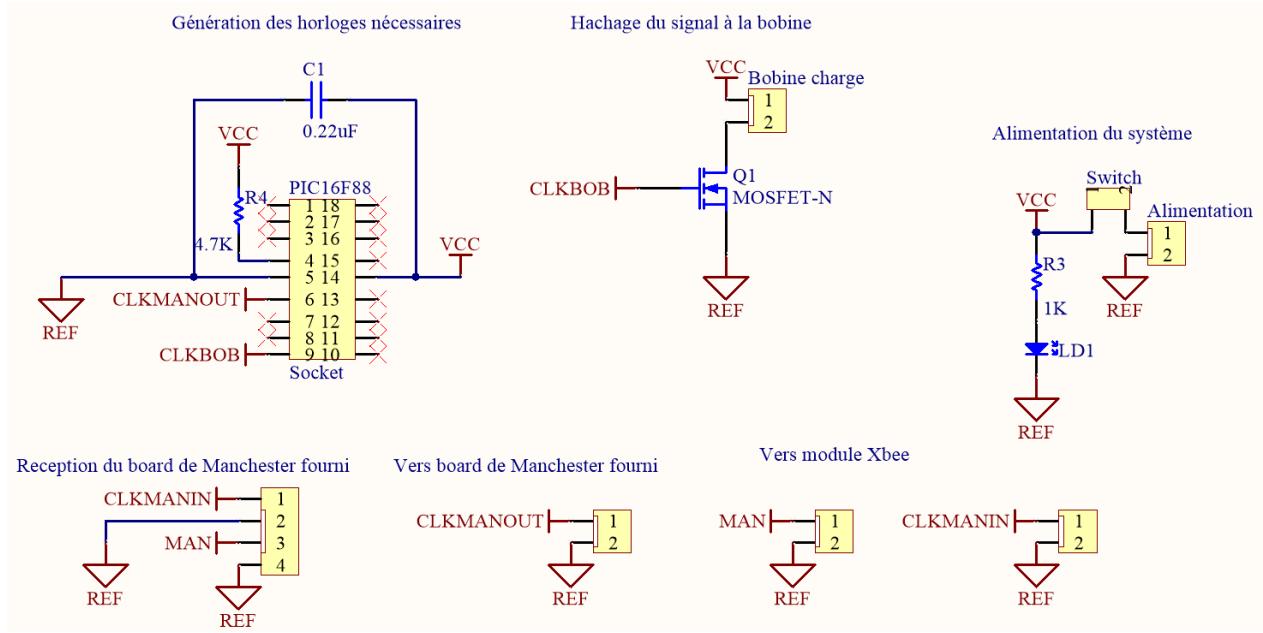


FIGURE 2.1 – Schéma électrique de la station de recharge

## 2.2 Circuit de charge et de décharge du condensateur

La figure 2.2 présente le circuit de charge et de décharge du condensateur. Le MOSFET est utilisé en envoyant un PWM à sa base afin de réduire l'effet de l'induction rémanente à l'électroaimant. Un interrupteur permet aussi de connecter une résistance permettant de décharger les condensateurs.

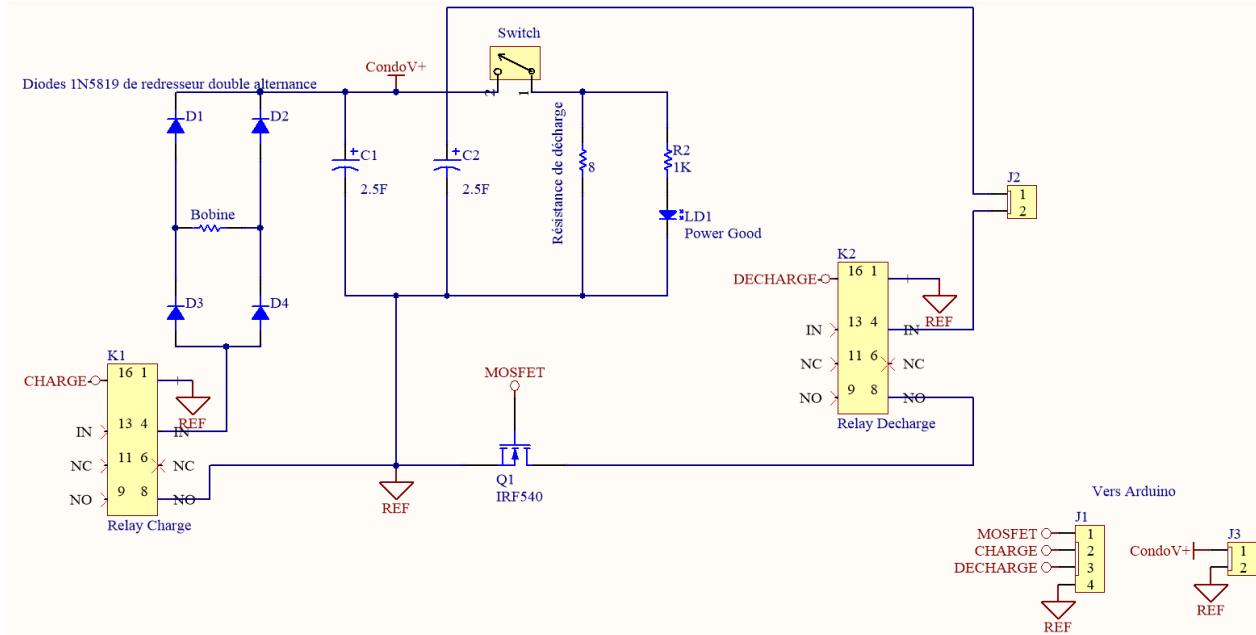


FIGURE 2.2 – Schéma électrique de la charge et la décharge du condensateur

## 2.3 Circuit de connexions pour les moteurs

### 2.3.1 Connexions moteurs et pont en H

La figure 2.3 présente le circuit de connexion utilisé pour les connections des encodeurs des moteurs et du pont en H. On utilise ce circuit situé sous le robot afin d'éviter de rallonger les fils des moteurs et de mieux organiser les fils.

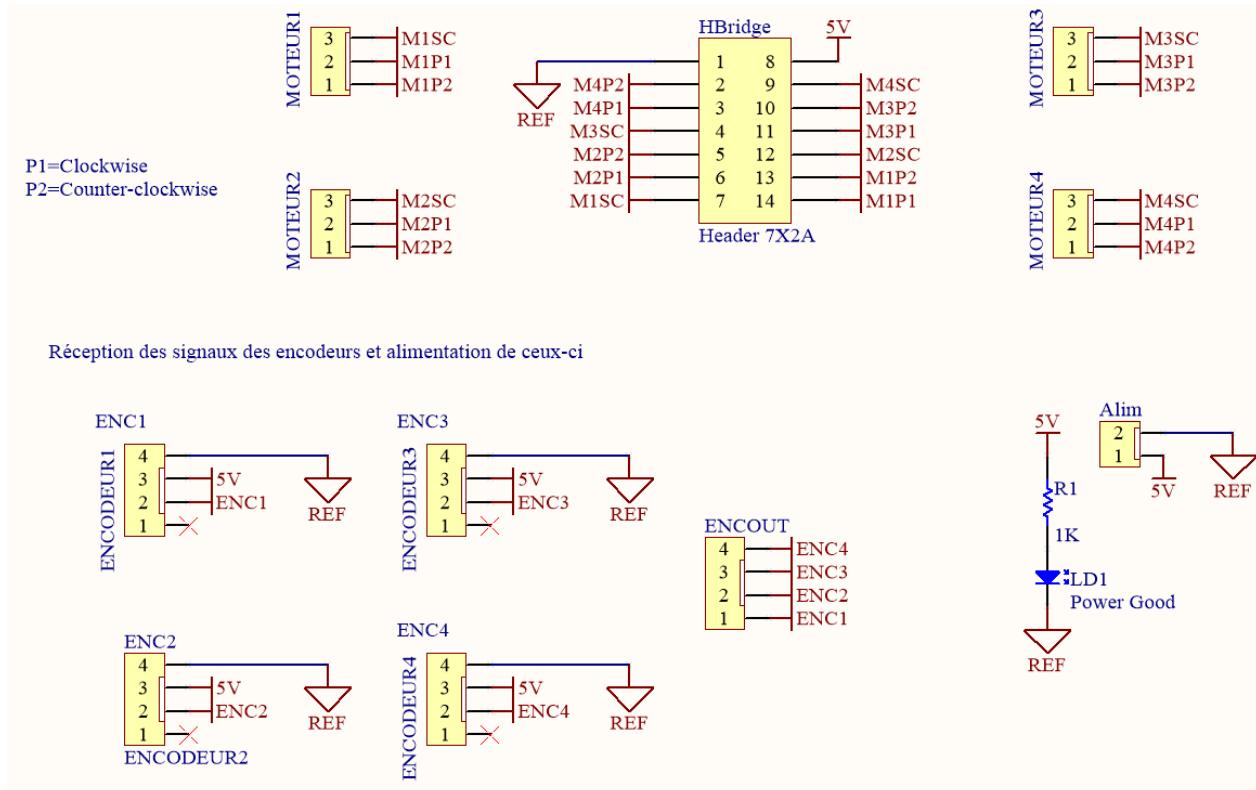


FIGURE 2.3 – Schéma électrique du circuit de connexions des moteurs

### 2.3.2 Connexions sur l'Arduino

La figure 2.4 présente les connexions de tout les systèmes sur l'Arduino.

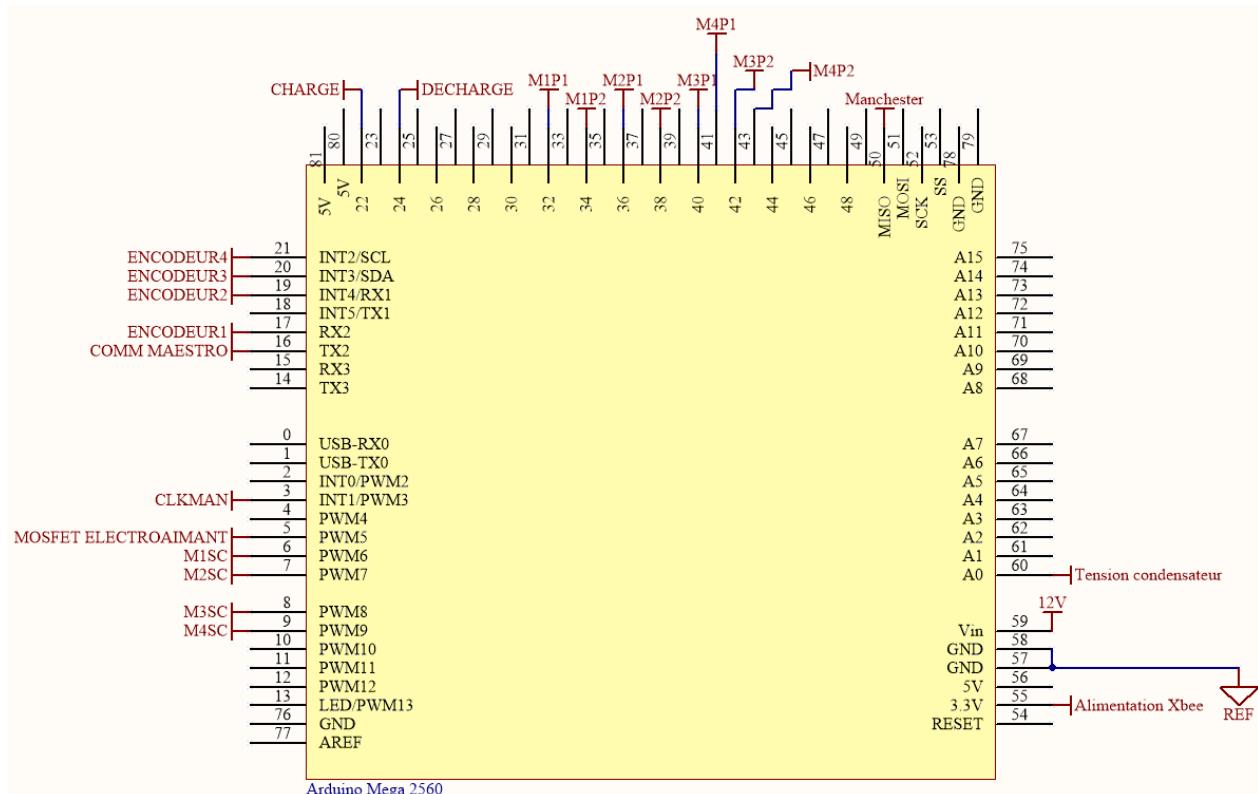


FIGURE 2.4 – Schéma des connexions sur l'Arduino Mega

## 2.4 Circuit d'alimentation

La figure 2.5 présente le circuit d'alimentation utilisé sur le robot.

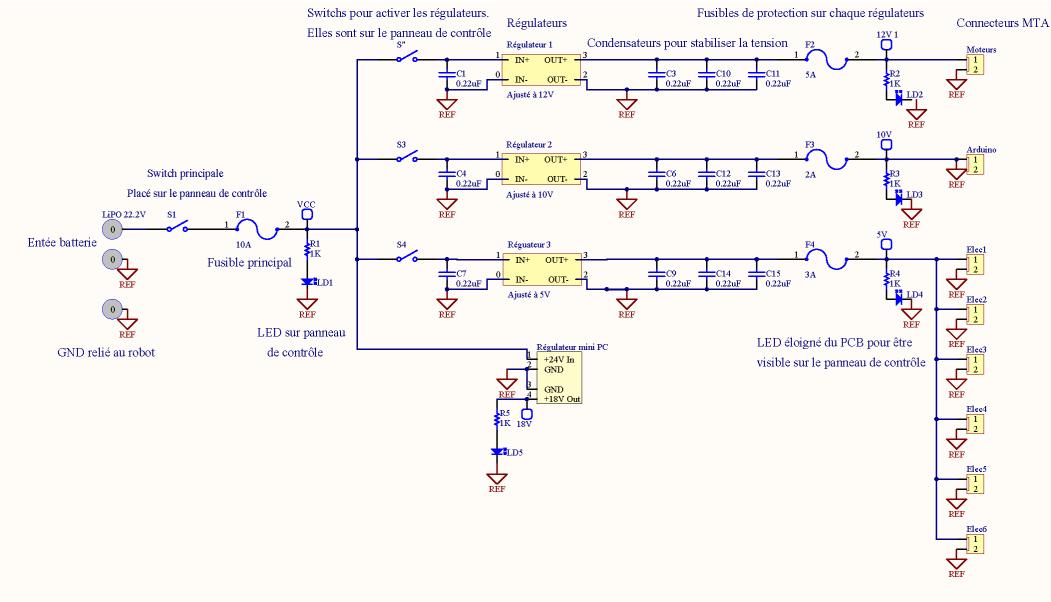


FIGURE 2.5 – Circuit d'alimentation du robot

# Chapitre 3

## Post-mortem

### 3.1 Points forts du robot

Dès le début de la session, l'équipe s'est mise d'accord sur l'importance de simplifier le design du robot le plus possible et donc de minimiser la quantité de problèmes possibles ainsi que de faciliter le débogage. Ceci est démontré par la simplicité d'implantation de la station de recharge, présentée à la section 2.1, en effet le circuit est très simple et est donc facile d'implantation tout en étant très efficace et permettant de recharger le condensateur à la tension voulue en environ 1 minute. Un autre point fort du robot est son préhenseur qui avec son servomoteur permet de soulever le trésor et de la placer dans une petite cavité où celui-ci est coincé. Cette astuce permet donc de couper l'électroaimant durant les déplacements et d'ainsi économiser la charge du condensateur ce qui limite la nécessité de se recharger entre les rondes ainsi que le risque d'échapper le trésor en plein milieu d'une routine.

Un autre des points forts du robot est l'analyse de la table de jeu à partir de plusieurs photos, ce qui permet de s'assurer du nombre et du positionnement des trésors ainsi que la couleur, la forme et la position des îles. La communication du code Manchester par modules *Xbee* ainsi que l'algorithme de décodage assurent un taux de réussite de 100% pour la transmission et le décodage. De plus, l'utilisation du *Xbee* ne demande pas d'alignement pour assurer la réception du code et celui-ci peut donc être reçu dès l'entrée dans la zone de la station de recharge.

Un aspect du robot qui était aussi très bien est son alignement avec les îles et le dépôt du trésor. En effet, une fois rendu devant l'île cible le robot utilise la caméra embarquée pour détecter sa cible et ajuster sa position en fonction de celle-ci. L'ajustement se fait de manière précise sur presque tous les essais et le dépôt du trésor est toujours complètement dans l'île et ne dépasse pas de celle-ci.

## 3.2 Améliorations potentielles

Bien que l'équipe a choisi une démarche simple dans la conception de l'asservissement et l'implémentation des déplacements en général, il y a des ajustements envisageables qui augmenterait la rapidité d'exécution. Pour l'instant, le robot tourne sur lui-même d'abord pour obtenir une orientation et se déplace ensuite vers l'avant, vers l'arrière, vers la gauche et vers la droite. Par contre, il ne peut pas se déplacer en diagonale ou selon une combinaison de déplacements simultanés en x et y. La figure 3.1 illustre le type de déplacement que le robot effectue.

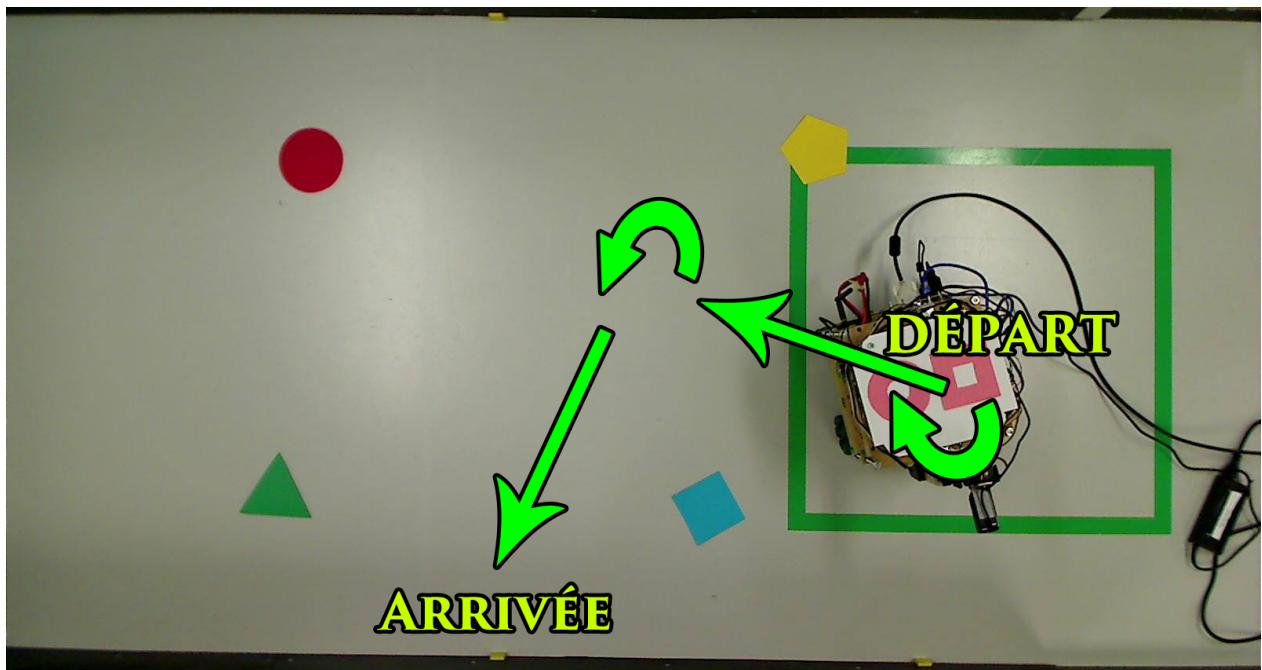


FIGURE 3.1 – Déplacement du robot présentement

Comme amélioration, le robot pourrait se déplacer en diagonale également. Il pourrait se déplacer avec des commandes "x, y" peu importe l'orientation du robot. Par exemple, si le robot se trouve à une position (100, 100), ceci peu importe son orientation et devant se déplacer vers la position (200, 200) sans obstacles sur son chemin, il devrait simplement se déplacer en ligne droite sans avoir à faire face à son objectif. La figure 3.2 suivante présente la trajectoire améliorée envisageable du robot. On remarque que le robot serait plus rapide passant de 4 tâches à 2 dans ce cas ci.

Ce problème force le robot à adopter une stratégie de déplacement sur des courtes distances seulement avant de recalculer sa trajectoire. Ne pas faire d'erreurs sur la trajectoire était l'objectif, c'est pourquoi même sur des lignes droites, le robot avance sur de plus courtes distances afin de s'assurer d'envoyer une correction plus rapidement pour les prochains points à atteindre. Le problème vient du fait que le robot est asservit en vitesse et pas en position, ce qui complique l'implémentation d'un mode de déplacement de ce genre. L'asservissement

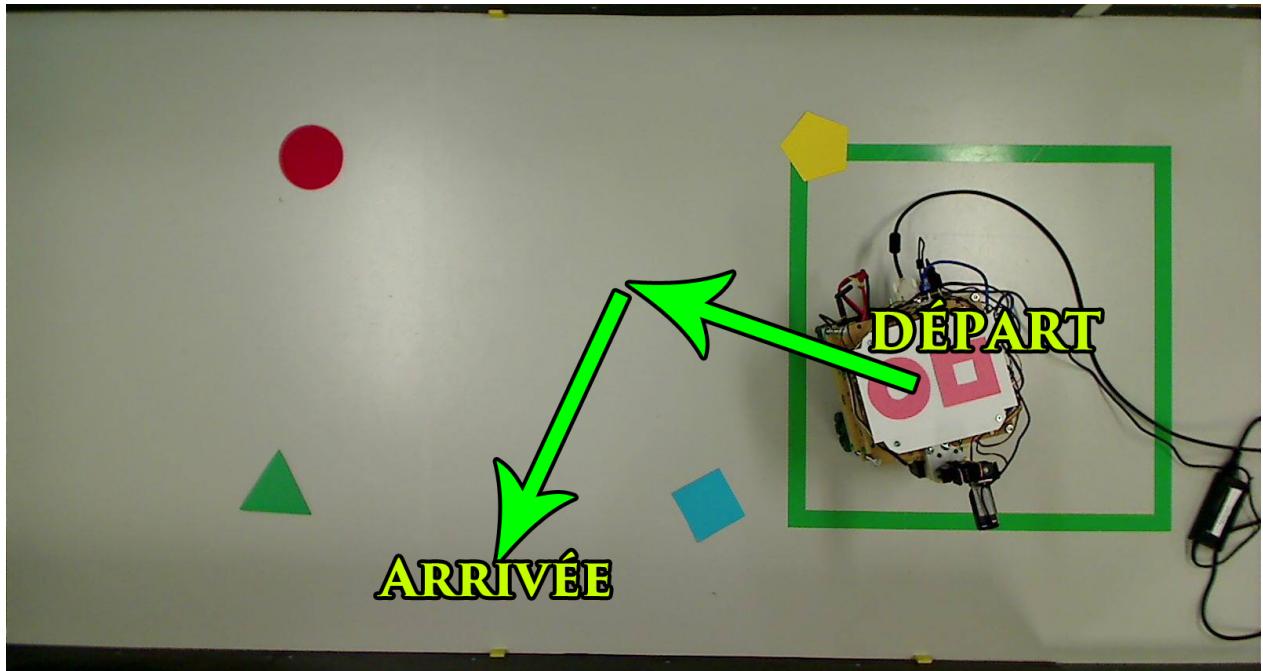


FIGURE 3.2 – Déplacement du robot permettant les directions diagonales

présent permet au robot de parcourir de manière précise la distance demandée par l'algorithme de trajectoire, mais ne permet pas de correction de déviation durant ce déplacement. L'ajustement s'effectue par la suite à l'aide d'une rotation si besoin est. Un des principaux points d'amélioration du robot serait donc d'implanter un véritable asservissement en position qui lui permettrait de rejoindre une position tout en corrigeant ses déviations en cours de route.

Également, l'algorithme de trajectoire trouve le chemin le plus court pour l'instant, mais il pourrait trouver le chemin le plus sécuritaire, c'est à dire passer le plus loin des îles, des murs, des trésors, etc. Il y a déjà un petit ajustement afin de passer entre deux îles de façon sécuritaire, mais il ne fait pas la différence entre un chemin présentant des trajectoires risquées où des trajectoires moins dangereuses.

Un des points à améliorer au niveau électrique serait d'installer des convertisseurs DC-DC plus petits, en effet, ceux installés présentent un affichage 7 segments intéressant pour choisir le niveau de tension de sortie, mais ils sont très gros. Installer des convertisseurs moins imposants permettrait de réduire l'espace occupé par le circuit d'alimentation et donc de placer tous les circuits électriques sur le même étage au lieu d'en avoir un seul situé sur le dessus du robot. Cette amélioration permettrait aussi de mieux disposer le filage du robot et le rendre moins apparent et donc permettre un suivi simplifié des connections.

Le gros point faible du robot est finalement l'alignement avec les trésors. En effet, le robot se déplace vers la position du trésor sélectionné par l'algorithme de déplacement grâce à la vision de la caméra monde. Une fois rendu sur place, la caméra embarquée prend le

relais et prend une photo de l'emplacement supposé du trésor pour détecter avec précision sa position et pouvoir s'ajuster. L'ajustement de position fonctionne très bien et est précis, par contre, la caméra échoue à détecter le trésor dans certain cas, dont lors de la compétition. Le problème est que si le robot ne détecte pas de trésor suite au filtrage de couleur, il passe à l'étape suivante (déposer le trésor sur l'île) au lieu de retenter la détection. Une correction de ce problème a été implémentée quelques jours avant la compétition, mais suite à l'intrusion de problèmes logiciels probablement causés par cet ajout, nous avons préféré retirer cette fonctionnalité. Un message était envoyé par le robot s'il était capable de détecter un trésor ou non. Dans le cas d'un échec, la station redirigeait le robot vers un autre trésor disponible. Il serait plus efficace de changer la technique de détection des trésor, puisque cette technique se basant sur le filtrage de couleur s'est avérée être inconsistante. Il aurait été intéressant d'étudier des techniques de détection alternatives ou utiliser des *threshold* adaptatifs afin de mieux gérer les variations des couleurs.

# Bibliographie

- [1] ETH Zurich, *An LED-to-LED Visible Light Communication System with Software-Based Synchronization*, [En ligne],  
<http://people.inf.ethz.ch/schmied/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] HobbyKing.com, *Turnigy 4500mAh 6S 30C Lipo Pack*, [En ligne],  
[http://www.hobbyking.com/hobbyking/store/\\_10284\\_Turnigy\\_4500mAh\\_6S\\_30C\\_Lipo\\_Pack.html](http://www.hobbyking.com/hobbyking/store/_10284_Turnigy_4500mAh_6S_30C_Lipo_Pack.html), Page consultée le 30 janvier 2016
- [4] ebay.ca, *DC-DC Step-down Buck Converter Adjustable Power 4V-38V to 1.25V-36V 5A Voltmeter*, [En ligne], [http://www.ebay.ca/itm/171445007919?\\_trksid=p2057872.m2749.12649&ssPageName=STRK%3AMEBIDX%3AIT](http://www.ebay.ca/itm/171445007919?_trksid=p2057872.m2749.12649&ssPageName=STRK%3AMEBIDX%3AIT), Page consulté le 30 janvier 2016
- [5] Robotshop.com, *Grove Electromagnet*, [En ligne],  
<http://www.robotshop.com/ca/en/grove-electromagnet.html>, Page consultée le 30 janvier 2016