Groupe de projet ingénieur n°70

date : 19/06/2012

Version 1.0

Rapport technique de projet Ingénieur

Rédacteurs :

Emmanuel Caille, Xavier Corbillon,
Léa Castel, Pierre Jacolot,
Jérémy Nadal, Guillaume Recht

Encadrants :

Noël Caillère – Dept Micro-ondes
Magali Le Gall – Dept Electronique
Jean-Pierre Clère – Dept Optique

Client :

Marie-Catherine Mouchot – Directrice de la Communication de Télécom Bretagne

TELECOM Bretagne

Résumé

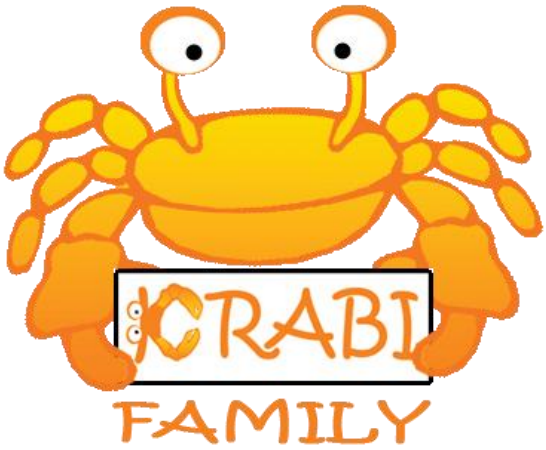
Ce document décrit le travail réalisé par les membres du club de robotique de Télécom Bretagne dans le cadre du projet d'ingénieur n°70. Le but de ce projet est de construire un robot pour participer à la Coupe de France de robotique, robot qui doit respecter le règlement de Planète science.

Nous décrirons tout d'abord le contexte global du projet, puis la gestion des ressources humaine et matérielles et la solution technique retenue sur les plans mécanique, électronique et informatique. Finalement nous évaluerons nos performances vis-à-vis des résultats obtenus lors de la coupe et de la transmission des connaissances techniques aux équipes futures.

Abstract

This document describes the work achieved by the members of the robotics club of Telecom Bretagne within the framework of the engineer project n°7. The aim of this project is to build a robot to participate in the French Robotics Cup, robot that has to follow the rules of Planète science.

We will describe in the part the context of the project, then the management of human and technical resources and the adopted technical solution on a mechanic, electronic and informatics point of view. Finally, we will estimate our performances regarding to our score and the transmission of technical knowledge to future teams.



Rédacteurs

Introduction	Pierre Jacolot
1. Objectifs et Contraintes du projet	Pierre Jacolot
2. Gestion de Projet	Jérémy Nadal
3. Développement technique du robot principal	
3.1. La stratégie	Xavier Corbillon
3.2. Les solutions mécaniques	Emmanuel Caillé
3.3. L'électronique	Jérémy Nadal et Guillaume Recht
4. Développement technique du robot secondaire	Léa Castel
5. Evaluation des Performances	
5.1 Communication	Emmanuel Caillé
5.2 Coupe de France	Guillaume Recht
5.3 Capitalisation des Compétences et Documentation	Guillaume Recht
5.4 Bilan Budgétaire	Pierre Jacolot
Conclusion	Léa Castel
Annexe 1 : Cahier des charges fonctionnelles	Léa Castel
Annexe 2 : Planning du projet	Jérémy Nadal
Annexe 3 : Le simulateur de déplacement	Xavier Corbillon
Annexe 4 : L'asservissement	Xavier Corbillon
Annexe 5 : La gestion de la mémoire du STM	Xavier Corbillon
Annexe 6 : Conception mécanique du robot principal	Emmanuel Caillé
Annexe 7 : Les capteurs	Jérémy Nadal
Annexe 8 : Cartes Electroniques	Jérémy Nadal
Annexe 9 : Diagramme UML	Xavier Corbillon

Relecteurs

Introduction	Léa Castel
1. Objectifs et Contraintes du projet	Léa Castel
2. Gestion de Projet	Pierre Jacolot
3. Développement technique du robot principal	
3.1. La stratégie	Emmanuel Caillé
3.2. Les solutions mécaniques	Léa Castel
3.3. L'électronique	Jérémy Nadal et Guillaume Recht
4. Développement technique du robot secondaire	Pierre Jacolot
5. Evaluation des Performances	
5.1 Communication	Léa Castel
5.2 Coupe de France	Léa Castel
5.3 Capitalisation des Compétences et Documentation	Léa Castel
5.4 Bilan Budgétaire	Léa Castel
Conclusion	Léa Castel
Annexe 1 : Cahier des charges fonctionnelles	Léa Castel
Annexe 2 : Planning du projet	Léa Castel
Annexe 3 : Le simulateur de déplacement	Emmanuel Caillé
Annexe 4 : L'asservissement	Emmanuel Caillé
Annexe 5 : La gestion de la mémoire du STM	Xavier Corbillon
Annexe 6 : Conception mécanique du robot principal	Léa Castel
Annexe 7 : Les capteurs	Guillaume Recht
Annexe 8 : Cartes Electroniques	Jérémy Nadal
Annexe 9 : Diagramme UML	Emmanuel Caillé

Tables des figures

Remerciements

Dans un premier temps, nous souhaiterions remercier nos encadrants techniques, Magali Le Gall, Noël Caillère et Jean-Pierre Clère, qui nous ont beaucoup aidé durant le déroulement de notre projet. Leurs conseils nous ont permis de nous sortir de problèmes que notre formation ne permettait pas toujours d'appréhender. Nous les remercions également d'être venu nous assister à la coupe de France de robotique.

Nous souhaiterions également remercier la Direction de la Communication de l'École et la directrice, Marie-Catherine Mouchot, pour l'aide financière apportées qui nous a permis d'acheter nos divers composants et de mener à terme notre projet. Nous remercions également Chantal Leblond qui nous a financé les t-shirts que nous avont porté durant la coupe. Enfin, nous voulions remercier Cendrine Le Locat pour les manifestions auxquels nous avons assistés, comme la fête de la science.

Ensuite, nous voudrions remercier les personnes qui nous ont aidées à réaliser les différents composants de notre robot. En particulier, Jean-Pierre Clère, qui, comme chaque année, a usiné la base roulante et la structure en aluminium de notre robot, ainsi que les différents éléments mécaniques. Dans ces remerciements nous voulions aussi citer Serge Pinel, grâce à qui nous avons pu percer nos cartes électroniques.

Nous souhaiterions remercier les membres actifs du club robotique qui ne sont pas dans le projet S4 ni dans le projet S2, pour nous avoir aidé à concevoir notre robot : Vivien Dequidt, Alexandre Dely et Ghassane Latfi.

Pour finir, nous remercions nos sponsors pour l'aide financière qu'ils nous ont apportés : sans eux, nous n'aurions jamais pu finaliser notre robot : la Société Général et Brest Métropole Océane.

Sommaire

Introduction

Dans le cadre de notre projet d'ingénieur à Telecom Bretagne nous avons souhaité mettre à profit nos compétences acquises lors des semestres précédents afin de participer à la coupe de France de robotique 2012 qui aura lieu à la Ferté-Bernard du 16 au 19 mai 2012 *[1]*. Pour cela nous avons analysé le besoin de notre client, développé notre produit et enfin participé à la coupe avec notre robot.

Pour ce projet nous avons participé à trois événement majeurs : la pré-coupe du Trégor le 31 mars, qui nous opposa a d'autre écoles de Bretagne, puis le coupe en elle-même et enfin le forum et la présentation de Juin qui clôturerons notre projet.

Nous présenterons dans ce document l'organisation globale du projet robotique et de notre équipe en particulier.

1. OBJECTIF ET CONTRAINTES DU PROJET

1.1 CONTEXTE

Dans le cadre des projets S2 et S4 menés à Télécom Bretagne, et en partenariat avec la Direction de Communication de l'école, deux équipes de projets participent chaque année à la coupe de France de Robotique afin de promouvoir l'image de Télécom Bretagne et afin de permettre aux étudiants membres du club de robotique de l'école de vivre leur passion

Nous avons commencé à construire, dès septembre, deux robots répondants au règlement imposé par Planète Sciences pour participer à la coupe qui s'est tenue cette année du 16 au 19 mai à la Ferté-Bernard (72) ***[1]***. De plus, lors de la réalisation, nous avons rendu compte de notre avancement à notre partenaire et aux organisateurs de la pour mieux préparer cet événement.

La participation à la coupe de robotique se fait en deux étapes : tout d'abord nous avons dû être homologués, c'est-à-dire jugé conforme au règlement de la coupe, puis vinrent les matchs de qualification qui consistent en cinq matchs, les scores de ces cinq matchs étant cumulés pour obtenir le classement final du robot.

A l'issue des qualifications, seules les seize premières équipes du classement peuvent accéder à la phase. Depuis la participation de Télécom Bretagne en 2000, Télécom Bretagne se classe aux alentours de la quarantième place, notre meilleur score, celui de l'année 2008, étant la 9ème place.

1.2 OBJECTIF ET BESOIN

L'objectif principal de ce projet est représenté l'école lors de la Coupe de France de robotique, celle-ci étant un évènement annuel très médiatisé : de nombreux journalistes, notamment pour TF1, viennent filmer et écrire des articles sur les équipes lors de la coupe.

Pour ce faire, nous devons non seulement homologuer notre équipe, pour être vu par les spectateurs, mais aussi faire le meilleur score possible pour promouvoir au mieux l'école. De plus nous participons aussi à de nombreux événements régionaux afin de donner la meilleure image possible de Télécom Bretagne.

Enfin nous devons transmettre au mieux les connaissances acquises pendant cette année et les années précédentes aux équipes suivantes pour permettre la promotion de l'école sur la durée.

Le Cahier des Charges du projet est présenté en ***annexe 1***.

1.3 REGLEMENT DE LA COUPE DE FRANCE

Cette année, exceptionnellement, chaque équipe peut présenter deux robots, un robot principal et un robot secondaire, tous les deux étant autonomes en suivant le règlement de la coupe ***[2]***. Le thème de la compétition de 2012 est « ***Treasure island*** ». Deux équipes s'affrontent lors d'un match de 90 secondes. L'équipe gagnante est celle dont le (ou les) robot(s) auront collecté le maximum de points. La table de jeu, sur laquelle devront évoluer les robots, est présentée sur la *figure 1*.

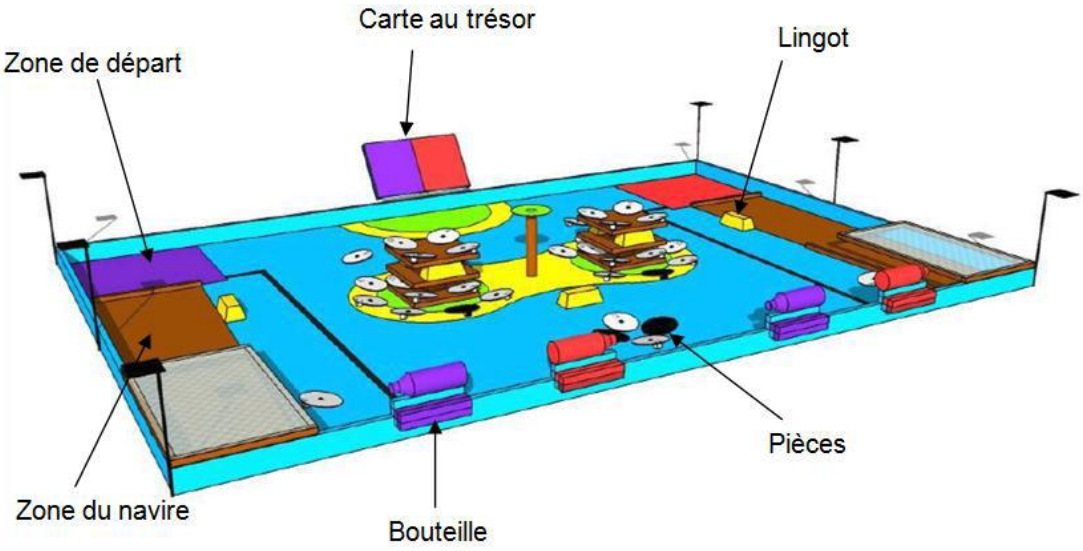


Figure 1: *Table de jeu de la Coupe de France de robotique*

Tout d'abord, nous avons les zones de départ. Chaque équipe se verra attribué à chaque match une couleur, qui sera la couleur de sa zone de départ.
Plusieurs éléments de la table permettent de gagner des points :

- **La carte au trésor**, qui une fois découverte permet de gagner 5 points. Pour la découvrir, il suffit que le robot retire le tissu de la couleur de sa zone de départ, qui la recouvre.

- **Les bouteilles**, qui permettent de gagner 5 points : il faut actionner un bouton poussoir, qui engendre un mécanisme, déroulant ainsi un message.

- **Les pièces** (qui sont des CDs), qui sont positionnées sur la table et sur les totems. Il existe des pièces noires et des pièces blanches, mais seules les pièces blanches permettent de remporter 1 point chacune.

- **Les lingots**, qui sont eux aussi positionnés sur la table et sur les totems. Chaque lingot permet de remporter 3 points.

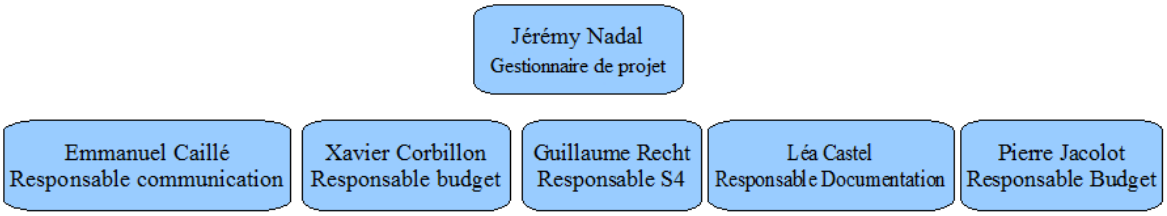
Les pièces et les lingots récoltés doivent être mis dans la zone du navire afin que les points soient attribués.

En outre, les robots doivent pouvoir détecter les robots adverses afin de ne pas entraver leur progression, et afin d'***éviter tout contact.*** Les robots doivent également éviter tous les obstacles de la table (totems, palmiers, etc.).

2. GESTION DE PROJET

2.1 L'EQUIPE

La structure de notre groupe de projet est un peu particulière par rapport aux groupes classiques du projet s4 car elle tient compte non seulement des membres du projet s4 mais aussi de ceux du projet s2 et des membres du club, extérieurs aux projets. Le groupe fonctionne globalement autour de **trois pôles** : informatique, électronique et mécanique. Chacun de ces pôles est encadré par un gestionnaire de projet (*figure 2*). Ceci permet d'avoir une vision globale de chacun des pôles qui travaillent en parallèle. Le président, en parlant avec les chefs de pôle est chargé d'avoir une vision globale du travail du groupe.



Jérémy Nadal Responsable électronique	Xavier Corbillon Responsable informatique	Gaëtan Fayon Responsable mécanique
Guillaume Recht	Samuel Baraud	Léa Castel
Qiao Wang	Alexandre Dely	Vivien Dequidt
Victor Dubois	Pierre Jacolot	Emmanuel Caillé
Ghassen Lafti		

Figure 2: *Organigramme de la gestion de projet*

Membres du projet S4

Jérémy Nadal – Responsable de la gestion de projet

Jérémy Nadal est le président du club Robotique, c'est donc tout naturellement qu'il ait pris la position de chef de projet. L'année dernière, il était membre du projet s2. Son rôle est d'organiser la gestion du projet. Il prépare donc des réunions avec les autres membres et les anime. C'est aussi lui qui s'occupe des formalités administratives pour l'inscription à la coupe de France de Robotique.

Il est également responsable du pôle électronique. Il s'est occupé de la formation des premières années et répartit le travail entre tous les membres de ce pôle. Il s'est chargé de refaire dans leur ensemble une bonne partie des cartes électroniques de cette année.

Xavier Corbillon – Responsable de suivi horaire et de l'avancement

Xavier Corbillon est le responsable du suivit horaire hebdomadaire, de la qualité et de l'avancement. Il comptabilise le nombre d'heure de travail de chacun des membres du groupe et en rend compte à Jérémy Nadal. Il doit aussi vérifier si les différentes tâches du projet sont dans les temps et s'assurer de la qualité des différentes tâches réalisées.

Xavier est aussi le responsable du pôle informatique. Il était déjà membre du pôle informatique l'année précédente mais n'était pas dans le projet s2. Tout comme les autres responsables de pôle, il a assuré la formation des premières années et réparti le travail entre tous les membres du pôle.

Emmanuel Caillé – Responsable communication

Emmanuel Caillé est le responsable communication du club. Il est l'interface entre le groupe de projet et le monde extérieur. Il est chargé de prendre contact avec le client, d'organiser les rendez-vous avec les contacts extérieurs et doit discuter avec les institutions en relation avec le projet. Il est également membre du pôle mécanique mais il intervient ponctuellement dans le pôle informatique. Il était déjà présent l'année dernière dans le pôle informatique mais n'a pas participé au projet s2.

Guillaume Recht – Responsable Forum S4

Guillaume Recht est le responsable Forum S4. Il doit s'assurer du bon déroulement du forum de présentation de notre projet S4 qui aura lieu fin juin. Notre projet nécessite beaucoup de matériels et une bonne organisation de son déplacement est nécessaire. Guillaume sera donc en charge de cette organisation et devra s'assurer que tout se déroule comme il le faut. Guillaume n'était pas dans le club robotique l'année dernière car il a été admis sur titre cette année. Il est membre du pôle électronique et s'est chargé de la fabrication de plusieurs des cartes électroniques du robot.

Léa Castel – Responsable Documentation

Léa Castel est la responsable documentation du projet. Elle doit prendre en charge la rédaction des comptes-rendus de chaque réunion avec les encadrant et des documents de travail. C'est aussi elle qui doit s'assurer que les documents présentés dans le cadre du projet respectent les différentes règles exigées. Léa a rejoint le club cette année et est membre du pôle mécanique. Elle est entre autre en charge de la partie mécanique du robot secondaire.

Pierre Jacolot – Responsable budget et trésorerie

Pierre Jacolot est le trésorier du club robotique, c'est donc tout naturellement qu'il a pris le poste de responsable budget du projet. Il doit s'assurer que les dépenses du club rentrent dans notre budget. Il valide les achats effectués par les différents pôles et rembourse les achats avancés par des membres. Il fait également partie du pôle informatique et était présent dans le projet s2 l'année dernière. Il s'occupe entre autre des différents algorithmes de déplacement du robot.

Membres du Projet S2

Cette année, le projet S2 est constitué de quatre membres répartis dans les trois pôles du club :

- Gaëtan Fayon responsable du pôle mécanique
- Samuel Baraud membre du pôle informatique
- Qiao Wang membre du pôle électronique
- Victor Dubois membre du pôle électronique qui s'occupe majoritairement du robot secondaire

Membres du club

En plus des différents membres des projets, quelques autres étudiants participent activement au bon fonctionnement du club. Nous pouvons principalement citer Vivien Dequidt, membre du pôle mécanique, Alexandre Dely membre du pôle informatique et, Hadi Tiab et Ghassane Latfi, membres du pôle électronique.

2.2 LA PLANIFICATION DU PROJET

Le projet robotique est un peu particulier par rapport aux autres projets S4 car il se déroule sur 8 mois. Il s'agit d'un projet long et qui peut s'avérer complexe, il est donc indispensable de planifier les différentes tâches du projet. Les exigences du projet S4 nous ont permis de formaliser cela : nous utilisons un diagramme de Gantt pour notre planification. Cette partie expliquera la manière dont le diagramme de Gantt a été conçu.

Le projet a commencé en septembre 2011, lorsque nous avons reçu le règlement de la coupe. Avant de réellement planifié notre projet, nous avons étudié le règlement et réfléchie à diverse solution pour la conception de notre robot. Après 2 semaines de brainstorming, nous avons aboutie à une solution.

Pour le ***pôle mécanique***, la première étape est de concevoir les plans mécaniques de la base roulante du robot, pour la faire construire le plus rapidement possible. En effet, sans la partie mobile du robot, il n'est pas possible de commencer les tests globaux en électronique et informatique. Un retard sur cette tâche peut s'avérer problématique car c'est du temps en moins pour les débogages. Après envoie des plans au mécanicien, le plan complet du robot doit être achevé. La dernière étape consiste alors à finaliser le montage des derniers éléments mécanique sur la base roulante.

Pour le ***pôle électronique***, la planification suit un schéma relativement linéaire : il faut dans un premier temps concevoir les schémas des cartes avant de les imprimer, souder et percer, puis les tester. Ces tâches sont répétées pour les versions suivantes des cartes, qui sont des améliorations des précédentes. Ainsi, nous avons un cycle en « V » pour chaque version des cartes. Il est important d'avoir des premières versions de carte fonctionnelle rapidement, pour détecter rapidement des bugs gênant, et être testé sur la base roulante du robot.

En ***informatique***, les différentes tâches peuvent être fait en parallèle, mais il y toujours une étape de conception, de réalisation puis de test/débogage, puis d'amélioration. Contrairement à la planification de la partie électronique, nous sommes plus dans un cycle en spiral. Ainsi, la plupart des tâches se déroulent sur une période de plus de 4 à 6 mois, comme par exemple l'amélioration de la stratégie qui doit s'adapter à l'avancé mécanique et électronique du robot.

2.3 LE PLAN DE MANAGEMENT ET LES OUTILS DE GESTION

Nous avons rédigé un plan de management pour formaliser l'organisation de la gestion du projet, qui est plus complet que cette partie du rapport. En plus du diagramme de Gantt, il s'y trouve d'autre outil de planification tel que les diagrammes WBS pour la définition de nos tâches et les fiches de lots. Il présente également le rapport d'avancement que nous délivrons chaque semaine à notre cliente, nos outils de communication comme notre site internet.

Le planning du projet est présenté en ***annexe 2***.

3. DÉVELOPPEMENT TECHNIQUE DU ROBOT PRINCIPAL

3.1 LA STRATÉGIE

3.1.1 La Stratégie du robot

La définition de la stratégie globale du robot est essentielle pour comprendre l'ensemble des choix qui ont été effectués dans les différentes étapes de conception du robot. La stratégie permet de définir les objectifs que l'on souhaite atteindre avec le robot et oriente donc sa conception informatique, mécanique et électronique.

Nous avons fixé trois objectifs principaux pour le robot principal. Le **premier objectif**, le plus important, est de pouvoir récupérer et déplacer jusqu'à la zone d'embarquement le plus d'éléments de jeux possibles. La réalisation de cet objectif permet de garantir un nombre minimal de points.

Le **deuxième objectif** est de permettre au robot de détecter son adversaire. En effet, pour avoir le droit de participer à la coupe, celui-ci doit être capable d'éviter son adversaire.

Le **troisième objectif** est de pouvoir vider au minimum un demi-totem afin d'augmenter de manière conséquente le nombre d'éléments de jeu que l'on peut récupérer. Il est à noter que ce troisième objectif est conditionné par la réalisation du premier.

Lors de la coupe, ce qui distingue les robots du haut du classement et ceux du bas du classement est souvent leur fiabilité. C'est pour cela que nous avons décidé de mettre en place plusieurs niveaux de difficulté dans la stratégie; on peut alors choisir avant chaque match la stratégie la plus adaptée. Tout d'abord, nous avons choisi de réaliser des déplacements simples, afin de récupérer seulement une pièce et un lingot (*figure 3*).

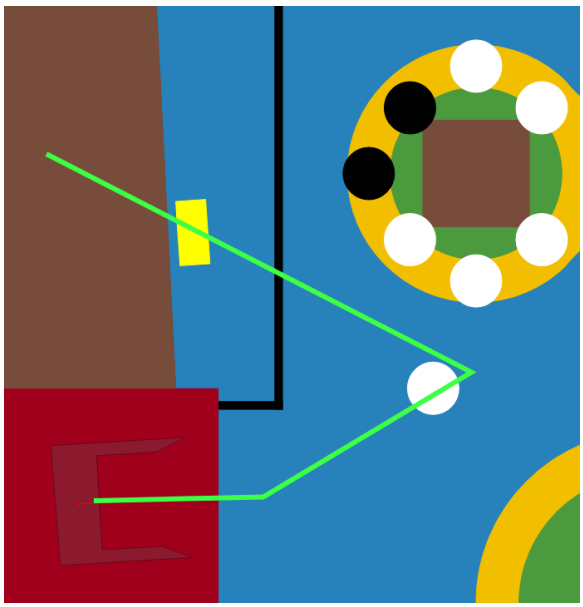


Figure 3: *1ère version de la stratégie du robot*

Cet objectif simple a l'avantage d'être extrêmement fiable. Le robot marque alors toujours quatre points et ne gêne en aucun cas le robot secondaire.

La deuxième stratégie du grand robot est de vider le demi-totem situé à proximité de son aire de départ (*figure 4*). Cette deuxième stratégie est un peu plus complexe car elle nécessite de faire fonctionner en même temps plus d'éléments du robot et d'avoir une meilleur précision sur les

déplacements. Elle est moins fiable car il y a beaucoup plus de chances d'échec (et donc de ne pas rapporter de points), mais en cas de réussite cette stratégie rapporte entre 10 et 15 points. Un échec de cette stratégie peu aussi bloquer le robot secondaire.

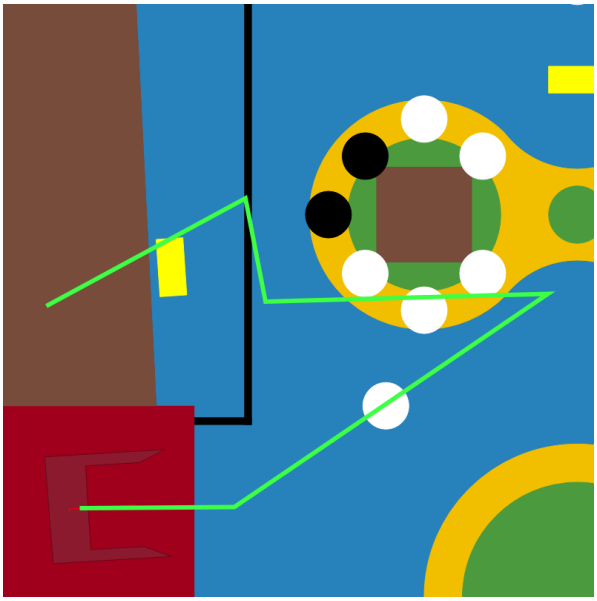


Figure 4: *2ème version de la stratégie du robot*

Le dernier niveau de la stratégie consiste à vider le premier demi-totem (comme dans la deuxième stratégie), puis de ressortir de l'aire de départ pour essayer de récupérer d'autres éléments de jeu au sol ou sur les totems. Il faut alors bien veiller à ne pas trop déranger le robot secondaire pour ne pas le faire échouer (*figure 5*).

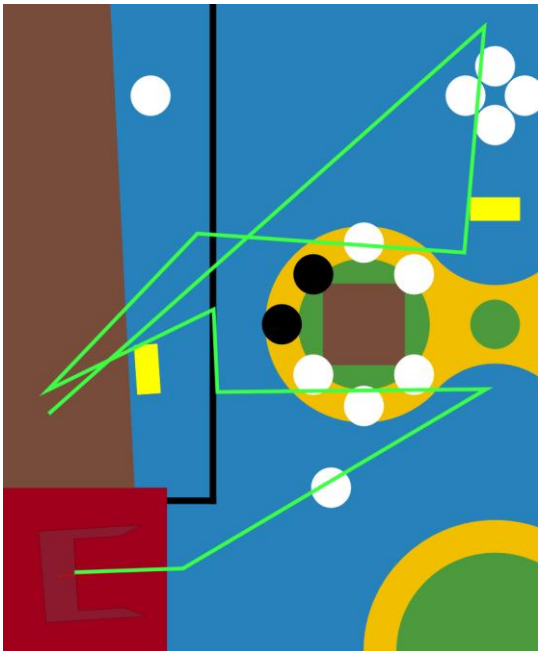


Figure 5: *3ème version de la stratégie du robot*

La stratégie d'évitement du robot adverse est elle aussi prévu avec différents niveaux de complexité. Tout d'abord, l'évitement basique qui consiste à s'arrêter dès que l'on détecte un adversaire et attendre que celui-ci s'éloigne pour repartir sur l'action que l'on était en train de réaliser.

Cet évitement garanti l'homologation mais peut s'avérer fatal si le robot adverse implémente la même stratégie ou si un élément du décor est confondu avec le robot adverse. En effet, puisque le robot attend que d'adversaire s'éloigne, si l'obstacle ne bouge pas, le robot va rester bloqué pendant tout le match. La stratégie d'évitement plus évoluée consiste à s'arrêter dès qu'on détecte un obstacle, attendre quelques secondes pour voir si l'obstacle s'éloigne ou pas, et si l'obstacle ne bouge pas, essayer de le contourner. Si le contournement échoue, on essaye de modifier la stratégie pour effectuer une action réalisable malgré l'obstacle. Cette stratégie d'évitement est bien plus complexe, elle a donc beaucoup plus de possibilités d'erreur mais permet, si elle réussit, de continuer le match malgré l'obstacle.

3.1.2 Architecture informatique

L'architecture globale du code informatique reste assez similaire à celle des années précédentes. En effet, il existe toujours une couche haut niveau (la stratégie) qui prend des décisions en fonction des informations qu'elle reçoit des couches bas niveau (les capteurs, l'odométrie) et qui envoie des consignes à l'asservissement et aux différents actionneurs du robot. L'architecture globale informatique est présentée sur la *figure 6*.

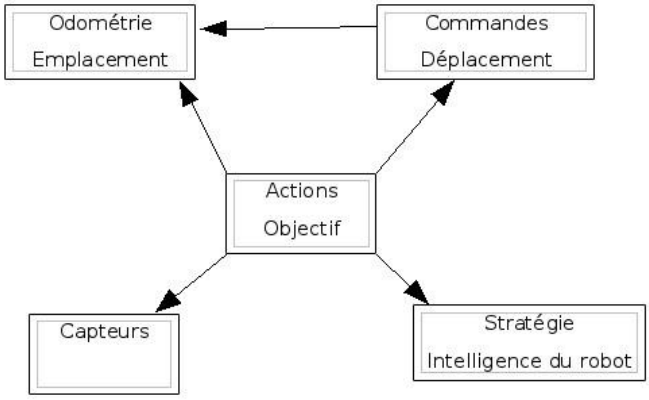


Figure 6: *Architecture globale informatique*

Néanmoins, nous avons modifié en profondeur quatre grandes parties du code. Tout d'abord la **stratégie**. L'ancienne stratégie était trop restrictive, pas assez dynamique et trop peut flexible pour permettre au robot de s'adapter réellement à l'environnement changeant de la coupe. Nous avons décidé d'ajouter un niveau d'abstraction supplémentaire : les actions.

Une **action** est un ensemble ordonné d'instructions cohérentes que doit effectuer le robot. L'ajout de ce niveau d'abstraction permet au robot de décider de changer d'action assez facilement au cours du temps. Par exemple, s'il avait pour action de vider un totem et de ramener la récolte dans le bateau alors, en cas d'obstacle, le robot peut facilement passer à l'action cohérente suivante et décider de finir de vider le totem plus tard, ce qui était difficilement faisable avant. L'ajout des actions facilite aussi la mise en place de procédures d'évitement des obstacles. Finalement, nous avons modifié en profondeur la gestion de la mémoire. En effet, il n'était auparavant pas possible de supprimer un objet créé en mémoire en cour de l'exécution du programme, nous avons donc modifié la manière d'allouer la mémoire afin de rendre ceci possible. Pour plus de précision vous pouvez vous référer à l'***annexe 5***.

Nous avons aussi modifié totalement la gestion des capteurs. L'année dernière, nous ne pouvions facilement gérer qu'un seul type de capteur à la fois. Avec cette nouvelle structure nous pouvons facilement gérer et traiter les réponses d'un grand nombre de capteurs à la fois grâce à un seul objet. Nous avons fait en sorte que ce code soit facilement adaptable pour les années futures. Globalement, l'ajout d'un nouveau type de capteurs ce fait simplement par l'ajout d'un nouveau module de gestion.

La dernière modification profonde a été effectuée sur la partie de gestion des ordres donnés au robot. Ces modifications ont surtout eu comme objectif de simplifier le code précédent en séparant les différentes fonctionnalités dans des classes différentes, afin de faciliter la gestion.

Il est à noter que tout le code informatique est exécuté par un microcontrôleur (cf glossaire). Ce microcontrôleur est un STM32-H103 (cf glossaire). Nous avons commencé la migration vers la gamme supérieure de ce microcontrôleur, le STM32-H107 qui possède deux fois plus de mémoire flash et 3 fois plus de RAM, mais par manque de temps et après un problème de reconnaissance des roues codeuses (cf glossaire), nous sommes restés sur la gamme H103.

Le code peut être découpé en différentes entités distinctes qui ont chacune des fonctionnalités différentes et qui communiquent entre elles (*figure 6*).

L'**odométrie** est l'objet qui s'occupe de calculer à chaque instant la position du robot sur la table ainsi que son orientation. La mise à jour de la position n'est pas faite en continue mais à chaque instant de calcul, qui correspondent aux interruptions générées par le microcontrôleur. Ces interruptions ont lieu toutes les 10ms. Pour calculer sa position sur la table, le robot utilise les informations reçues par les roues codeuses et utilise une approximation des petits angles. Cette approximation nécessite que les instants de mise à jour soient assez proches les uns des autres sans quoi une erreur d'approximation s'ajoute à chaque mise à jour de la position.

Les **capteurs** sont cette année gérés par un seul objet qui permet de récupérer facilement l'ensemble des informations des trois types de capteurs disposés sur le robot : cinq télémètres infrarouges (SHARP), trois « fin de course » et un télémètre ultrason. Ces différents capteurs sont principalement utilisés pour l'évitement des obstacles.

Les **commandes** permettent de demander au robot d'aller à une position donnée sur la table. La commande s'arrange alors pour que la courbe de vitesse du robot suive une courbe trapézoïdale. En effet, la commande génère une courbe idéale de vitesse que devrait atteindre le robot et transmet ses informations directement à l'asservissement qui se charge de corriger les erreurs de déplacement du robot. Le fonctionnement de l'asservissement ainsi qu'un exemple sont fournis en ***annexe 4***.

Pour plus de détails sur la structure du code, n'hésitez pas à aller consulter le diagramme UML en ***annexe 9***.

3.1.3 Les différents capteurs et leur utilité

La position des différents capteurs sont représentés sur la *figure 7*.

Les **capteurs SHARP** sont représentés en bleu avec une flèche indiquant la direction de vue. Ils servent à la détection des robots adverses. En effet, afin de respecter l'homologation dynamique, il nous faut éviter de rentrer en contact avec ces derniers

.

Selon l'action en cours d'exécution, certain capteurs SHARP seront inhibés et les informations qu'ils renvoient ignorées. C'est le cas à proximité des totems (pour que le robot ne les confonde pas avec l'ennemie).

Les **capteurs « Fin de course »** sont représentés en vert. Les deux à l'arrière servent à positionner le robot avant un match ou à se calibrer en bord de table si le robot est perdu sur la zone de jeu. Celui à l'intérieur permet de savoir si des éléments de jeu de type lingot ou pièces sont rentrés dans la zone de stockage du robot.

Le **capteur ultrason** est représenté en rouge. Il possède une vision supérieure à celle des SHARP. Il sert à connaître la distance à laquelle se trouve le premier objet le plus proche du robot, situé dans le cône de vision du capteur, et d'en déduire une action optimale. Des informations supplémentaires sur ces capteurs sont fournies en ***annexe 7*** de ce rapport.

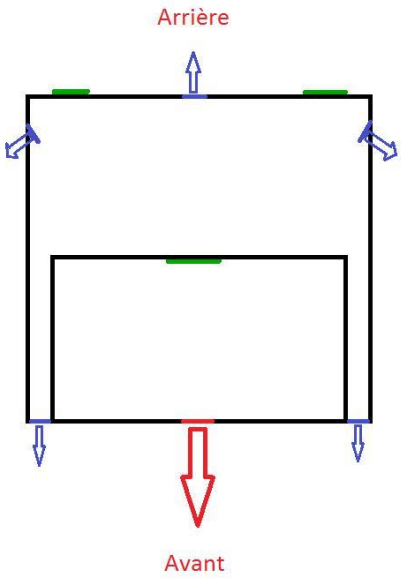


Figure 7: *Emplacement des capteurs sur le robot*

3.1.4 Les tests

Les tests et le débogage du code sont une étape essentielle de la conception du robot. C'est lors de ces tests que l'on découvre toutes les erreurs dans le code et tous les comportements anormaux du robot. Mais réaliser ces tests n'est pas une chose facile. Il existe différents niveaux dans le code.

Pour tout le code bas-niveau, qui nécessite de fortes interactions avec l'environnement électronique, il faut souvent que les cartes électroniques fonctionnent correctement avant de pouvoir faire les tests. Le code haut-niveau, comme la stratégie, nécessite que l'ensemble du code bas-niveau soit fonctionnel ainsi que la plupart des cartes électroniques et la mécanique du robot. Ceci est difficilement possible longtemps avant la coupe. Afin de pouvoir tester le plus rapidement possible le code haut-niveau, nous avons donc mis en place un simulateur qui permet de simuler le comportement du robot. Le simulateur génère le comportement d'un robot avec un asservissement parfait et une odométrie parfaite. Plus d'informations sont fournies en ***annexe 4***.

Nous essayons le plus possible de travailler en parallèle pour optimiser le temps de travail. Ceci est facilité par l'utilisation de notre gestionnaire de version (git).

3.2 LES SOLUTIONS MÉCANIQUES

Pour répondre aux besoins exprimés par la stratégie tout en respectant les contraintes du règlement de la Coupe, le pôle mécanique du projet a dû créer les différents organes du robot. Cette année, nous avons misé sur la fiabilité et donc la simplicité.

Vous trouverez en ***annexe 6*** quelques mises en plan détaillées du robot, ces mises en plan ont été réalisés grâce au logiciel SolidWorks (cf glossaire) et il en est de même pour toutes les modélisations 3D présentées ici.

3.2.1 La base roulante

Nous avons commencé par la « base roulante ». Cette base doit permettre au robot de se déplacer et de se repérer sur la table, c'est sur elle que reposera toute la structure du robot. Elle doit également permettre de stocker et transporter les pièces et lingots, il faut qu'elle puisse en contenir un maximum pour limiter les allers retours. Il faut en outre qu'elle respecte la contrainte de périmètre imposée par le règlement.

À partir de toutes ces contraintes nous avons imaginé, modélisé puis construit la base roulante (*figure 8*). Elle se compose de plusieurs parties :

- un grand espace à l'avant pour stocker les éléments de jeu,
- deux roues codeuses (cf glossaire) sur les côté pour que le robot connaisse sa position (plus elles sont à l'extérieur meilleur sera la précision de l'angle mesuré), ces roues sont montées sur glissières verticales pour adhérer à la table en toute situation,
- deux roues motrices
- deux moteurs reliés aux roues par une courroie crantée : cette courroie permet d'avoir un peu plus de souplesse dans le positionnement des moteurs tout en assurant une bonne transmission mécanique. Ces moteurs ont été placés sur un étage intermédiaire pour maximiser l'espace de stockage.

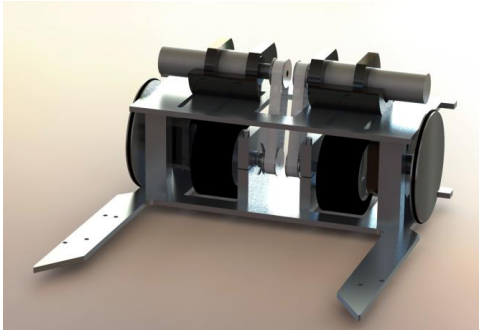


Figure 8: *Modélisation 3D de la base roulante*

Une fois les pièces modélisées nous faisons une mise en plan que nous envoyons au mécanicien du département Optique, J.P. Clerc, puis quand elles sont usinées, nous les assemblons. Le résultat final est représenté sur la *figure 9* et la *figure 10*.

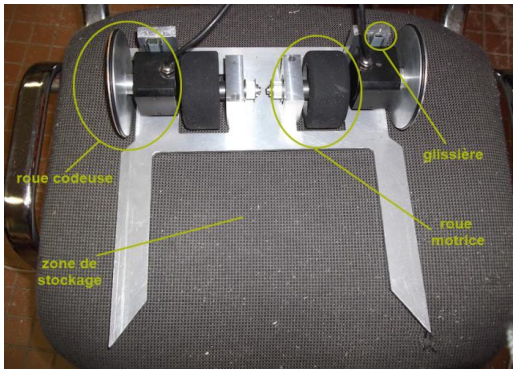


Figure 9: *La base roulante sans les moteurs*

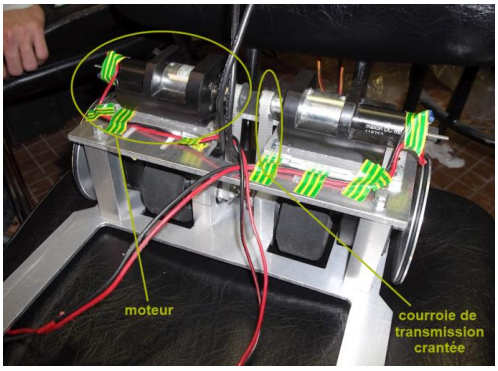


Figure 10: *La base roulante au complet*

Les premiers tests ont révélé un petit problème : la base roulante étant légèrement surélevée car elle repose sur les roues motrices et sur deux patins à l'avant, il y a un risque que le robot roule sur les pièces, qui sont faites à partir de CDs, ce qui peut le coincer. Aussi, les roues codeuse ne reposant plus sur le sol, nous donnent des informations faussées sur son emplacement. Pour bloquer les pièces et palier ce problème nous avons disposé du polystyrène tout autour de la base roulante du robot, comme le montre la *figure 11* qui suit :

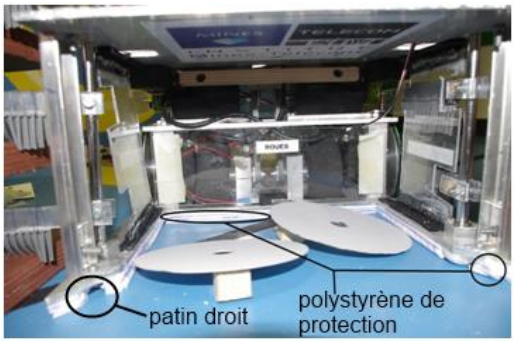


Figure 11: *La base roulante du robot équipé de ses polystyrènes de protection et de ses patins avant*

Les moteurs utilisés pour le robot principal sont très puissants (ils délivrent chacun un coupe 150N.m). Cependant, nous ne les utilisons qu'à 25% de leur puissance maximal. En effet, si nous ne faisions pas ceci, le robot serait entraîné et deviendrait « incontrôlable ».

3.2.2 L'étage supérieur

Nous avons également dû installer les cartes électroniques et les batteries sur le robot. Pour ceci, nous avons conçu un nouvel étage, sur lequel reposent les capteurs SHARP. Ces derniers ont été installés de telle sorte à pouvoir détecter les totems et les autres robots. L'étage supérieur est donc installé à 14,5 cm au-dessus de l'aire de jeu et les SHARP à 15 cm.

Cet étage est soutenu par 4 montants en forme de U, cette forme permet d'avoir une bonne résistance tout en ayant une masse et un encombrement faible. Au niveau modélisation et réalisation, nous obtenons la *figure 12* et la *figure 13*:

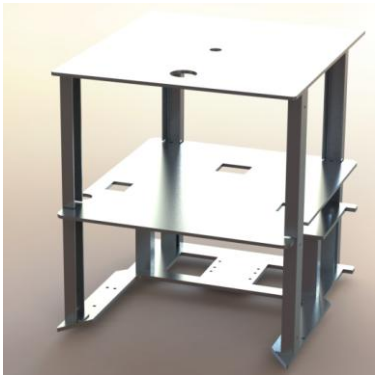


Figure 12: *Le squelette de Krabi 2012, avec son étage supérieur et son toit*

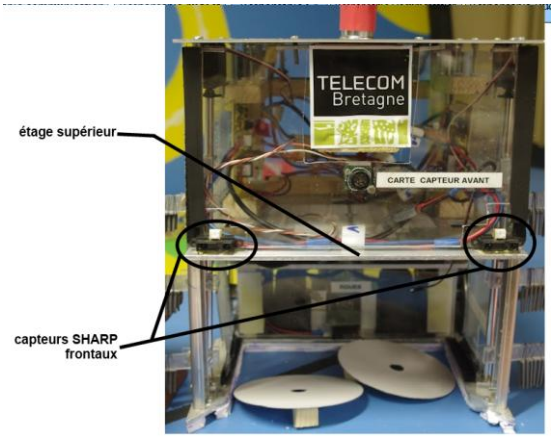


Figure 13: *Vu sur l'étage supérieur du robot avec les capteurs SHARP frontaux visibles*

En outre, afin de garder un marge suffisante sur les contraintes de hauteur imposées par le règlement (350 mm), nous avons positionné le « toit » du robot se situe à 31 cm au-dessus du sol.

Un ingénieux système permet de fixer les cartes au robot, et de les démonter très rapidement en cas de problème électronique (cf ***annexe 6***).

3.2.3 Les capteurs et leurs cartes

Les différents capteurs du robot (capteur SHARP, capteur « fin de course » et ultrason) ont été positionnés sur le robot afin de correspondre aux éléments de stratégie élaborés par le pôle informatique.

3.2.4 Les balais latéraux

Pour avoir plus de points il faut récupérer les éléments disposé sur les trois étages des deux totems de la table. Nous avons pour cela conçu des balais latéraux, qui sont composés chacun de trois peigne, pouvant « balayer » les différents étages du totem. Ces deux balais sont actionnés par un servomoteur (cf glossaire) qui permet de les sortir à l'approche de totem et de les rentrer après. Leur conception mécanique est modélisée sur la *figure 14* et la *figure 15*.

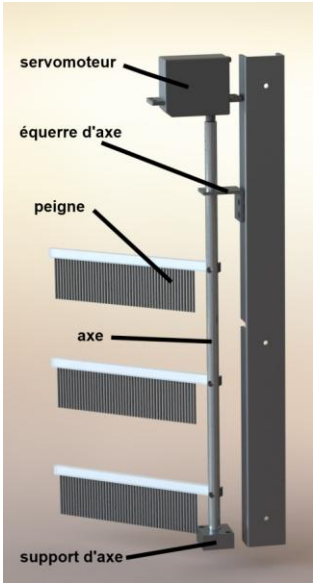


Figure 14: *Modélisation des balais du robot*

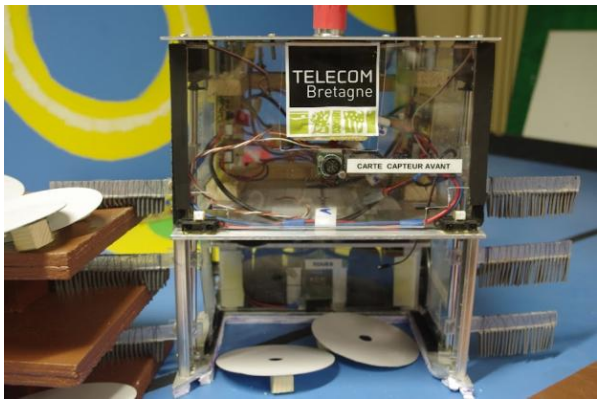


Figure 15: *Les balais latéraux déployés sur le robot longeant un des totems*

3.2.5 Le râteau frontal

Nous espérons pouvoir emmagasiner des éléments de jeu dans l'espace prévus à cet effet dans la base roulante. Le problème survient lorsque le robot recule. En effet, il risque de perdre son « butin ». Aussi, nous avons conçu un système de râteau frontal (*figure 16*), qui devra s'abaisser lorsque le robot recule. Ce mécanisme est actionné par un servomoteur (cf glossaire) fixé sous l'étage supérieur, qui est commandé par le code pour s'abaisser dès que le robot fait une marche arrière.

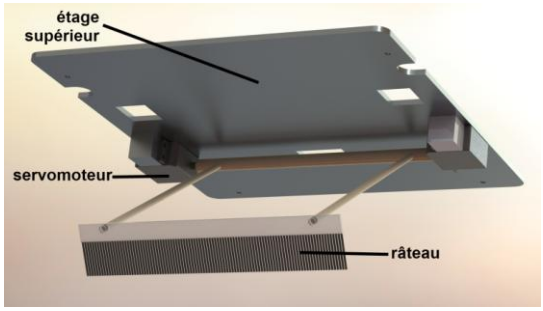


Figure 16: *Modélisation du râteau du robot principal*

3.2.6 Derniers éléments et homologation statique

Il reste encore quelques éléments indispensables à l'homologation : le bouton d'arrêt d'urgence, le support de balise et un système de « tirette » permettant d'indiquer au robot le top départ des matchs. L'année dernière la tirette avait été défaillante et nous avait couté une défaite, nous avons donc cette année installé une deuxième tirette en cas de défaillance de la première.

Ces éléments ont été installés sur le toit du robot présenté en *figure 17* :

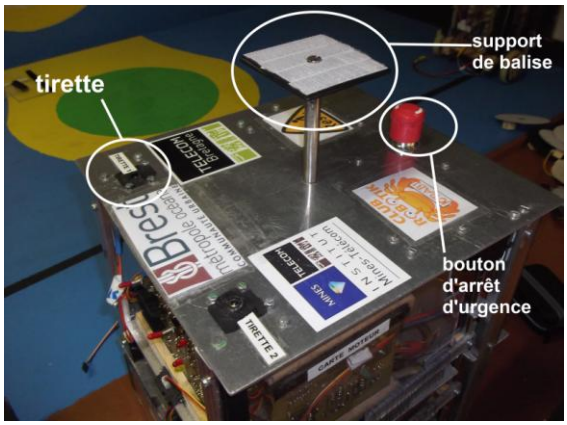


Figure 17: *Le toit du robot avec les derniers éléments*

Une fois le robot terminé il ne reste plus qu'à vérifier les mesures des périmètres du robot. Le périmètre non-déployé ne doit pas dépassé 1200 mm et le périmètre déployé 1600 mm. Lors de la phase de conception, nous avons pris en compte ces contraintes, mais nul n'est à l'abri d'une erreur. Au final, le robot non-déployé possède un périmètre de 1190 mm non-déployé et de 1440mm déployé.

Les critères d'homologations statiques et dynamiques ont donc bien été respectés. En outre, les éléments concernant la stratégie ont bien été implémentés au niveau mécanique.

3.3 L'ÉLECTRONIQUE : FAIRE LE LIEN ENTRE LA STRATÉGIE ET LES SOLUTIONS MÉCANIQUES

3.3.1 Architecture fonctionnelle de l'électronique

Architecture globale

Cette année, nous avons remodelé la structure des cartes pour avoir une électronique plus fiable et mieux protégée, compte tenu des problèmes rencontrés les années précédentes. Nous voulions également avoir une structure simple à comprendre, même pour une personne inexpérimentée dans ce domaine, afin de simplifier la transmission de connaissance dans les années futures. Les cartes devaient donc avoir des fonctions claire et précise, en évitant une trop grosse dépendance entre elles, ce qui permet également de simplifier les tests unitaires, c'est à dire lorsque les cartes sont testées indépendamment.

Intuitivement, nous pouvons identifier 2 parties distinctes dans l'électronique d'un robot : une partie qui s'occupe de fournir la puissance aux actionneurs, comme les moteurs ou servomoteurs (cf glossaire), et une partie correspondante à l'intelligence du robot, qui commande la partie puissance via des signaux logiques. Nous avons défini notre structure en partant de ce constat. Du côté de la partie logique, il y a naturellement une carte qui accueillera le cerveau de notre robot, le microprocesseur. Du côté de la partie puissance, nous avons identifié deux fonctions, et donc deux cartes : une pour alimenter les actionneurs, et une pour commander les moteurs. Suites aux problèmes des années précédents, nous voulions une séparation électrique entre ces 2 parties, pour éviter que les problèmes sur les cartes de puissance se propage sur la partie logique, et endommage le microcontrôleur (cf glossaire).

Ainsi, nous avons retenu une structure avec 3 cartes principales et 2 cartes secondaires qui servent juste d'interface (*figure 18*). La carte alimentation permet de délivrer les niveaux de tensions et la puissance aux différents actionneurs. Cette carte s'occupe aussi d'alimenter et de commander les servomoteurs (cf glossaire), grâce aux signaux provenant de la carte STM, l'intelligence du robot. La carte STM récupère les informations des roues codeuses (cf glossaire) pour connaitre la position du robot, puis les traitent et envoie des signaux de commande PWM (cf glossaire) (vitesse) et SENS (sens de rotation) à la carte moteur. Celle-ci va faire la transition entre la logique et la puissance pour

commander les deux moteurs, alimentés par la carte alimentation. Nous insistons sur le fait que cette transition se fait de manière isoler pour séparer totalement ces 2 parties.

Les deux cartes restantes ont pour rôle de récupérer tous les signaux des capteurs pour directement les envoyer à la carte STM. Elles n'ont pas de fonction particulière, mise à part de servir d'interface en récupérant seulement les signaux utiles. Cela évite de mettre plusieurs rallonges sur les câbles des capteurs, car les cartes sont placées dans des endroits stratégiques, et surtout simplifie le schéma de la carte STM.

L'architecture électronique globale du robot principal est présentée sur la *figure 18*.

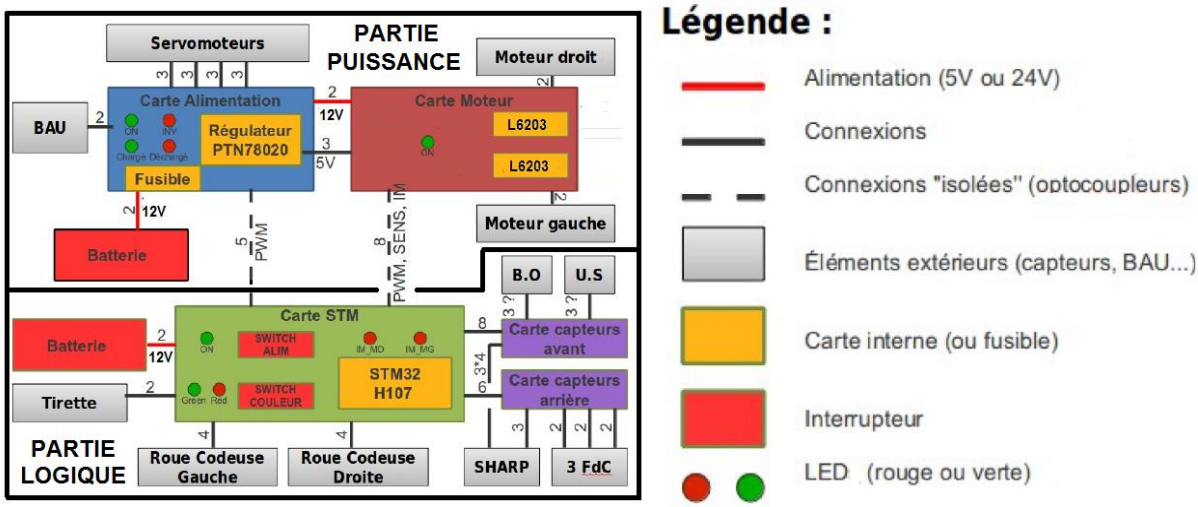


Figure 18: *Structure électronique globale*

Position des éléments sur le robot

La position des cartes sur le robot doit être réfléchie avant même de les avoir imprimés et soudés. C'est durant la conception des schémas des cartes, et plus particulièrement des typons, que cette question doit être soulevée. En effet, des cartes mal placés peuvent être plus difficile à câbler et à monter, ce qui peut être source d'erreur et une perte de temps. De plus, il faut que les cartes soient accessibles dans le robot, pour rapidement les réparer ou changer en cas de problème.

Notre idée était de placer les cartes sur les côtés du robot, protégé par du plexiglas facilement retirable (*figure 19*). La plupart des composants sont placés du coté intérieur du robot. Les diodes électroluminescente (LED) et interrupteurs sont placés du coté extérieur pour être facilement accessible par les membres de l'équipe. La carte STM se trouve à l'arrière du robot pour que la carte se retrouve face aux membres du pôle informatique souhaitant débuguer le robot.

Les capteurs sont disposés dans des endroits stratégiques afin de n'avoir aucun angle mort lors de la détection. Les cartes capteurs sont placées de manière à être les plus proche possibles des capteurs.

Ce positionnement des cartes s'est avéré très efficace : leur branchement s'est avéré relativement simple à faire, et le débogage des cartes pouvaient se faire directement sur le robot sans les démonter, ce qui est un gain de temps énorme. Nous conseillons aux années futures de reprendre cette idée. Le seul point négatif, ce sont les batteries qui étaient difficilement accessible, mais ce problème peut être corrigé avec un toit en plexiglas retirable.

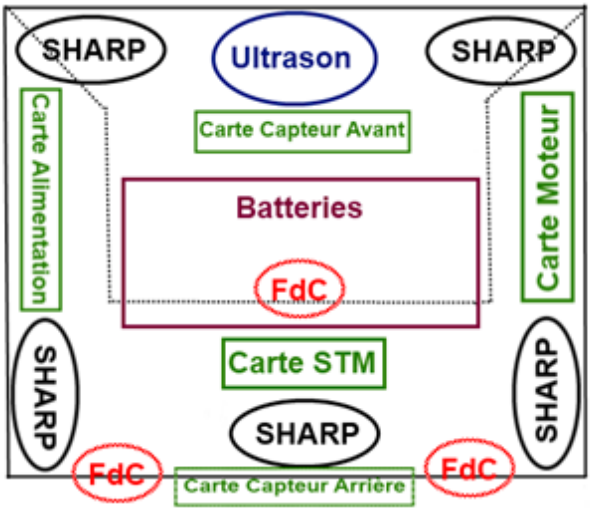


Figure 19: *Positions des éléments électroniques sur le robot*

Ainsi, nous avons vu l'architecture globale de l'électronique du robot principal. Nous pouvons maintenant nous atteler à décrire plus en détail le fonctionnement de chaque carte. Dans ces parties, l'analyse sera seulement fonctionnelle. L'***annexe 8*** traite des choix technologiques effectués pour chaque carte.

3.3.2 La carte STM

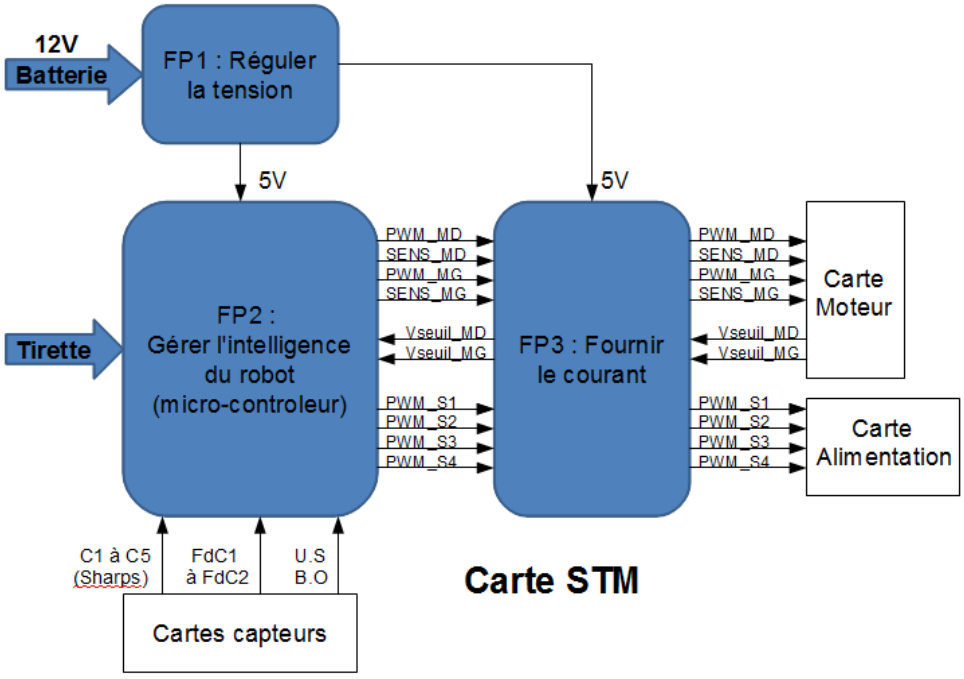


Figure 20: *Fonctions principales de la carte STM*

La carte STM représente l'intelligence du robot. C'est sur cette carte où se trouve notre microcontrôleur (cf glossaire), le STM-H107 (cf glossaire). Cette carte reçoit toutes les informations provenant des différents capteurs et des roues codeuses (cf glossaire) pour être envoyé au STM et envoie des signaux de commande à la carte moteur et alimentation. On peut décomposer la carte STM en 3 fonctions principales (*figure 20*), notées FP sur le schéma :

- **FP1** : Réguler la tension pour obtenir 5V à partir de l'alimentation à 12V, afin d'alimenter la partie logique (STM, capteurs, etc.).
- **FP2** : Gérer l'intelligence du robot. Le robot est sous le contrôle d'un microcontrôleur, le STM32-H107, placé sur la carte STM.

- **FP3** : Fournie le courant nécessaire pour que les signaux puissent être traité sur les autres cartes, car le STM seul ne peut pas fournir le courant nécessaire sur ses sorties pour alimenter les entrées des optocoupleurs (cf glossaire).

3.3.3 Les cartes capteur

Il existe deux type de cartes capteurs : la carte capteur arrière, qui contrôle le capteur SHARP et les capteurs « Fin de course » situés à l'arrière du robot, et, la carte capteur avant, qui contrôle les quatre autres capteurs SHARPS et le capteur ultrason à l'avant du robot. Ces cartes permettent d'alimenter les capteurs mais également de renvoyer au microcontrôleur les signaux des capteurs.

3.3.4 La carte alimentation

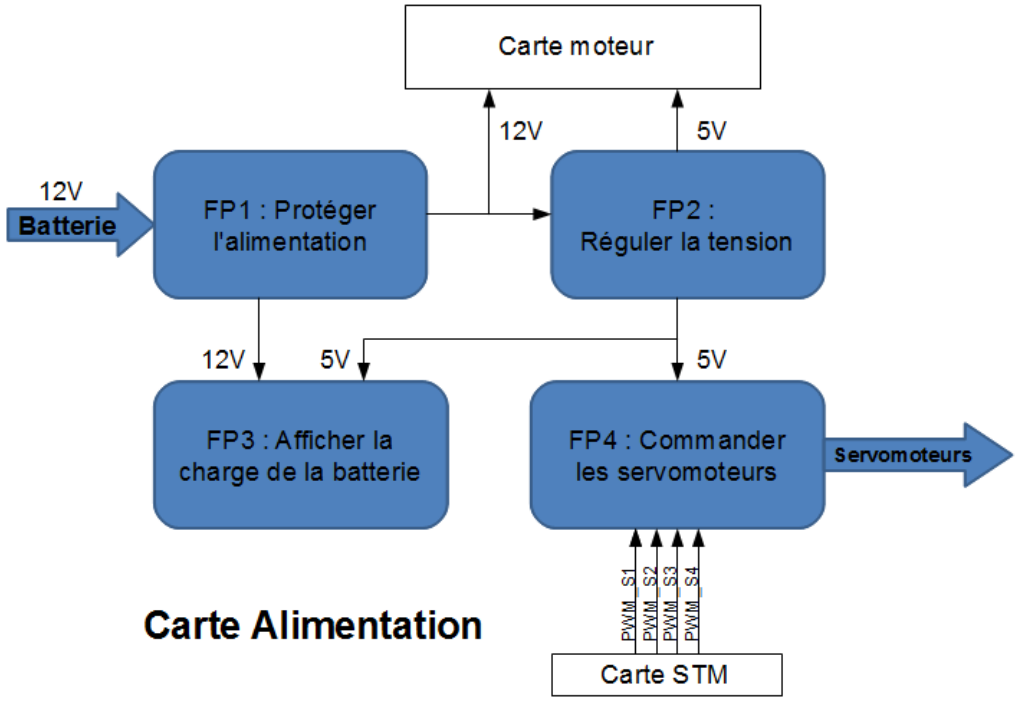


Figure 21: *Fonctions principales de la carte alimentation*

Cette carte permet de fournir les différents niveaux de tension aux différents composants et la puissance nécessaire aux actionneurs, tel que les moteurs et les servomoteurs (cf glossaire). Elle est découpée en 4 fonctions principales (*figure 21*) :

- **FP1** : Protège l'alimentation en cas de court-circuit, ou de chute de tension. Le bouton d'arrêt d'urgence (BAU) agit sur cette fonction et permet de couper l'alimenter en cas de problème, ou pour des raisons de tests. A noter que cette fonction ne protège pas totalement la carte contre les erreurs de branchement de l'alimentation. Ce n'est pas un oubli de notre part : un tel système risque de poser des problèmes à cause du retour de courant des moteurs.
- **FP2** : Régule la tension pour obtenir du 5V. Elle est composée d'un régulateur (cf glossaire) à découpage PTN78020 qui délivre une tension régulée pour les servomoteurs et pour la partie logique des cartes alimentation et moteur ;
- **FP3** : Détecte si la batterie est chargée ou déchargée. La LED verte s'allume lorsque la batterie à une tension supérieure à 12.2V, et la LED rouge s'allume dans le cas contraire.
- **FP4** : Isole électriquement les signaux provenant de la partie logique de notre structure, c'est à dire la carte STM. Les signaux reçus sont alors délivrés aux servomoteurs, alimenté grâce au 5V obtenu grâce à FP2.

3.3.5 La carte moteur

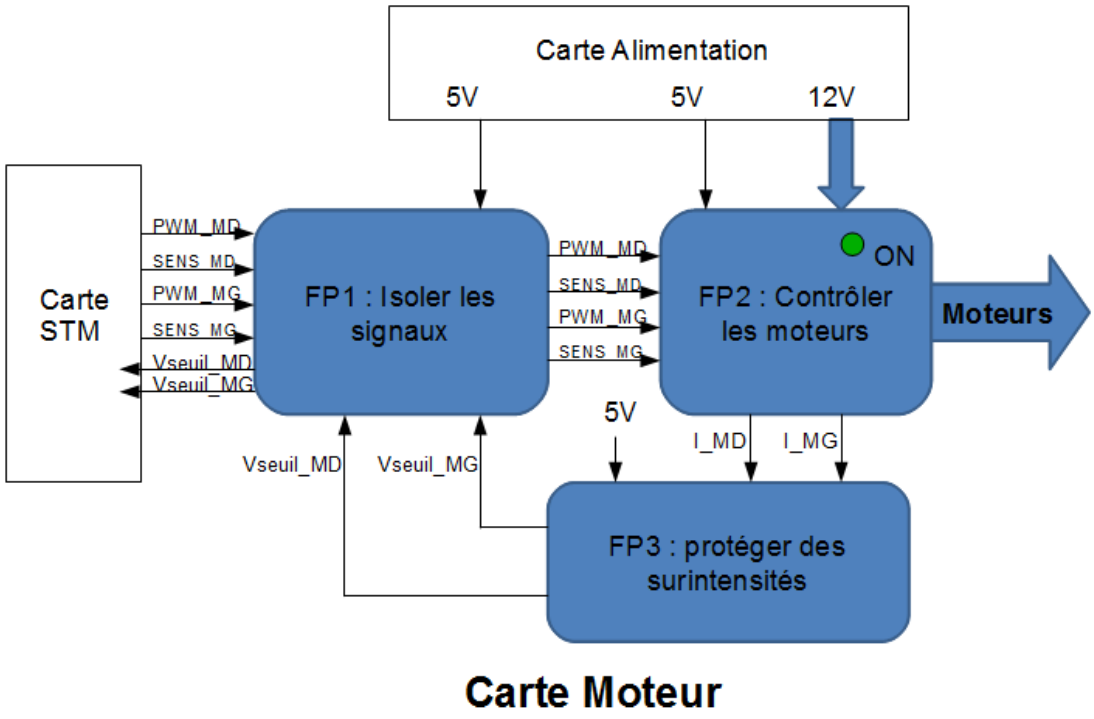


Figure 22: *Fonctions principales de la carte « Moteur »*

La carte « Moteur » a pour rôle de fournir la puissance nécessaire pour contrôler les moteurs. Nous pouvons distinguer 3 fonctions (*figure 22*) :

- **FP1** : Isoler les signaux. Cette fonction a pour but de protéger la carte STM en cas de retour de courant et simplifier le réglage de l'asservissement. En effet, auparavant, le bouton d'arrêt d'urgence arrêtait à la fois le microcontrôleur (cf glossaire) et les moteurs. Aussi, cela rendait peut pratique le réglage de l'asservissement : lors de celui-ci, nous ne pouvions arrêter le robot sans perdre toutes les courbes calculées par le STM (cf glossaire).
- **FP2** : Contrôler les moteurs. Cette fonction est réalisée à l'aide d'un composant, le pont en H (cf glossaire), qui permet de changer le sens et la vitesse des moteurs grâce aux signaux PWM (cf glossaire) et SENS provenant du microcontrôleur (cf glossaire). Une LED verte permet d'informer si la carte moteur est bien alimentée.
- **FP3** : Protéger des surintensités. Cette fonction mesure le courant qui traverse les moteurs (et donc les ponts en H) grâce à un capteur à effet Hall, puis compare la mesure obtenue à un seuil. Si l'image du courant dépasse ce seuil, une alerte est envoyée au microcontrôleur (cf glossaire).

3.3.6 L'alimentation du robot

Batterie

Cette année, nous utilisons 2 batteries au plomb de 12V pour alimenter notre robot. Une batterie s'occupe d'alimenter la partie logique, et la deuxième la partie puissance. Cela peut paraître étrange d'utiliser 2 batteries et non une pour tout alimenter, mais nous y sommes contraints si nous voulons isoler correctement ces 2 parties, et donc gagner en fiabilité. De plus, il s'est avéré pendant les tests qu'il est avantageux de séparer les batteries. En effet, contrairement à ce que l'on pourrait penser, ce n'est pas la partie puissance qui consomme le plus, mais belle et bien la partie logique, pour une raison simple : durant les tests, le robot est 90% du temps à l'arrêt, et donc moteurs coupés, alors que la carte STM est tout le temps alimentée pour pouvoir insérer un programme sur le microcontrôleur (cf glossaire) à chaque modification. La carte STM consomme environ 400 mA en continu, et la carte moteur 3A pour 10% du temps, soit une moyenne de 300 mA. Nous avons presque quadruplé les temps de test grâce à cette configuration de batterie.

Contrairement aux trois dernières années, nous n'avons pas utilisé du 24V pour nos moteurs, pour deux raisons. La première, c'est la nécessité de mettre une 3e batterie, mais cela aurait été trop encombrant pour le robot, aussi bien au niveau du poids que de la place. La deuxième raison, c'est que nous n'avons jamais utilisé nos moteurs jusqu'à leur pleine puissance : le robot est difficilement contrôlable à pleine vitesse, et les risques de collisions sont non négligeables car l'inertie est du coup beaucoup plus grande. Nous nous sommes toujours limités à un PWM (cf glossaire) de 50%, soit 12V. Cela ne change rien non plus au niveau de la consommation, car les batteries auraient de toute façon été mises en série.

Nos batteries sont surdimensionnées par rapport aux besoins exprimés dans le cahier des charges : elles peuvent facilement tenir 2 matchs de 90s successifs. En effet, leur charge étant de 5Ah, et la consommation maximale du robot de 5A, le robot peut fonctionner 1h en continu. Ce surdimensionnement est cependant nécessaire pour faire des tests prolongés.

Système de démarrage

Comme nous l'avons dit précédemment, le règlement de la coupe nous impose un système de démarrage avec une tirette, qui agit comme un interrupteur. Au niveau électronique, lorsque la tirette n'est pas encore retirée, le STM (cf glossaire) détecte un 0 sur un de ses pins (cf glossaire). En effet, la tirette est reliée à la masse. Lorsque cette dernière est retirée, le STM reçoit un signal qui lui permet de démarrer.

En début de match, nous devons également signaler au robot la zone de départ sur laquelle il sera placé. Pour ce faire, nous avons installé un interrupteur sur la carte STM, qui se traduit par la délivrance d'une tension de 5V ou 0V sur un pin du STM, suivant la couleur : pour le côté rouge, on impose une tension de 5V et pour le côté violet, on impose une tension de 0V.

4. DÉVELOPPEMENT TECHNIQUE DU ROBOT SECONDAIRE

Comme nous l'avons dit précédemment, nous pouvons cette année et exceptionnellement, présenter deux robots qui évolueront sur la même table de jeu. Aussi, nous avons décidé de produire un robot secondaire qui épaulera le robot principal afin de gagner le maximum de points.

4.1 STRATEGIE ET INFORMATIQUE

La stratégie initialement prévue pour le robot secondaire était la suivante : il devait actionner les mécanismes des deux bouteilles. Pour ce faire, à l'aide de capteurs de couleur que nous détaillerons ultérieurement, il devait suivre la ligne noire pour atteindre la première bouteille. Le deuxième mécanisme étant plus difficile à atteindre, nous avons opté pour une solution de « temporisation » : suite à de nombreux tests, nous avons pu estimer le temps moyen que mettait le robot à accéder à hauteur de la deuxième bouteille. Ensuite, il lui suffisait de faire une rotation de 90° pour gagner cinq points de plus. Cette stratégie est présentée en *figure 23*.

Cependant, nous avons pu remarquer lors de nos nombreux tests que les capteurs de couleur placés initialement sur le devant du robot, n'était pas d'une grande fiabilité. En effet, il s'est avéré qu'ils ne fonctionnaient pas totalement à tous les matchs. Aussi, avant de partir à la Coupe de France, nous avons modifié la stratégie du robot concernant la première bouteille : pour l'atteindre, il devait réaliser la même méthode que pour la seconde bouteille.

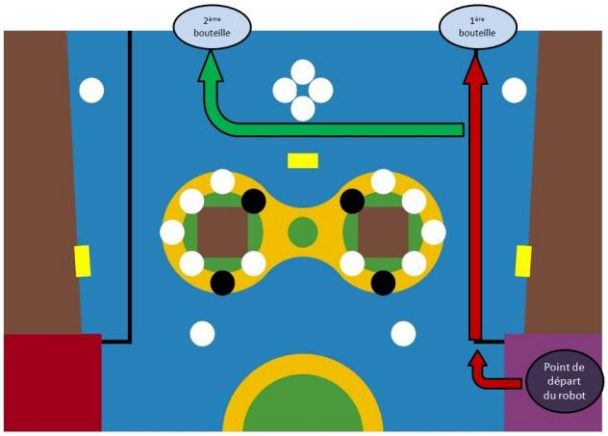


Figure 23: *Stratégie du robot*

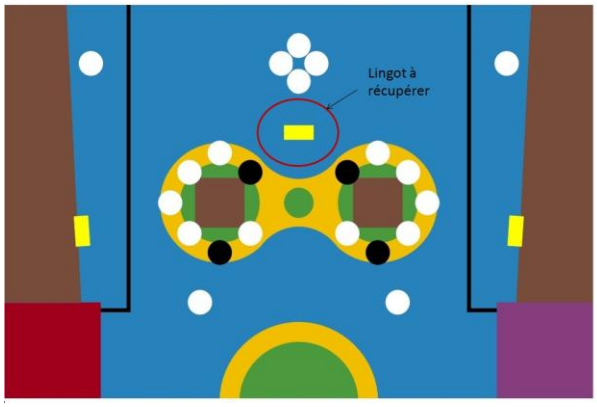


Figure 24: *Lingot à récupérer*

Lors de la coupe, voyant que les 10 points donnés par les bouteilles étaient assurés, et les performances des autres équipes lors de l'homologation, nous avons voulu rajouter un élément dans la stratégie du petit robot, à savoir prendre le lingot présent sur la *figure 24*.

Bien que nous ayons réussi à gagner un match grâce à cette nouvelle stratégie, étant donné que le robot secondaire n'est pas doté de roues codeuses (cf glossaire) comme le robot principal, elle n'est pas très stable.

Pour la partie informatique du robot secondaire, nous avons choisi d'utiliser la technologie Arduino (cf glossaire) en raison de sa simplicité d'implémentation : le langage de programmation utilisé est le C/C++.

4.2 MECANIQUE

Pour la structure mécanique du robot, nous avons choisi une structure basique en bois, pour plus de simplicité. Nous avons réalisé deux versions mécaniques du robot. La première a servi pour les tests électroniques et informatiques du robot. Cependant elle ne respectait pas les contraintes de tailles imposées par le règlement pour le robot secondaire (son périmètre était trop important). Aussi, nous avons réalisé une deuxième version, qui est plus compact mais qui présente les mêmes éléments que la première. Ici, nous décrirons donc la deuxième version mécanique du robot secondaire.

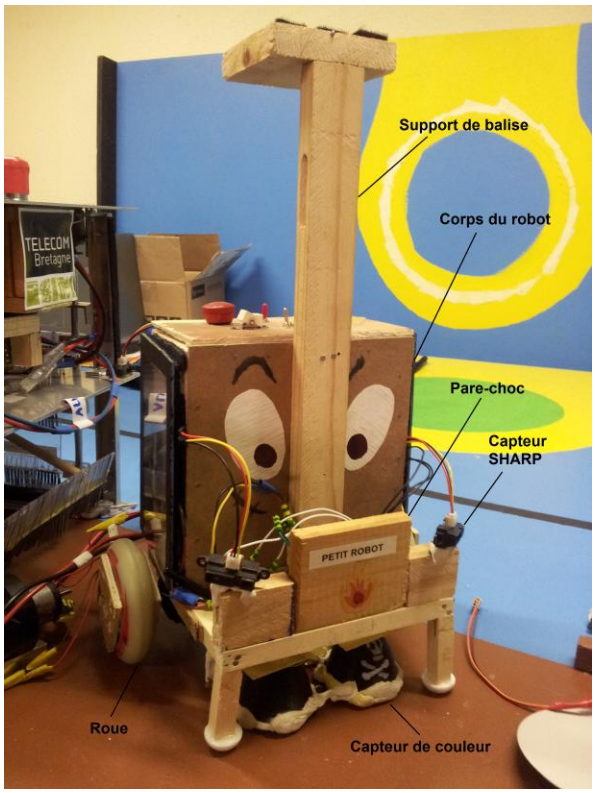


Figure 25: *Le robot secondaire vu de face*

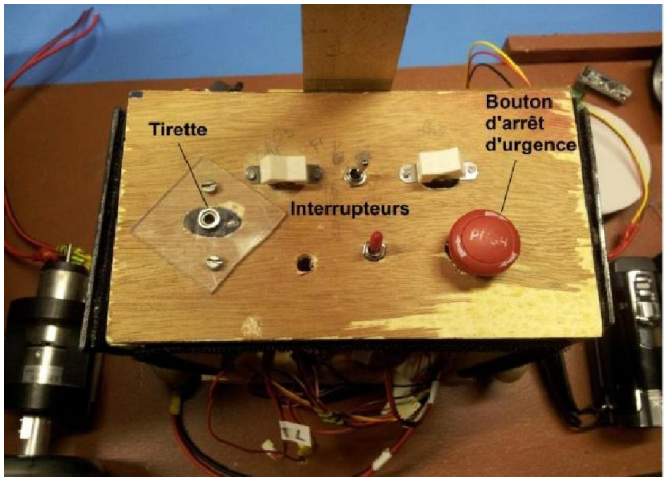


Figure 26: *Le robot secondaire vu de dessus*

La *figure 25* présente le robot secondaire de face. Voici les éléments mécaniques qui le constituent :

- Le **support de balise** est indispensable pour la participation du robot à certains matchs. En effet, certaines équipes utilisent des balises pour pouvoir connaître leur position mais également la position des robots adverses sur la table. Ces supports de balises sont soumis à des règlementations dans les dimensions : il doit être placé à 43cm en hauteur (±5mm) et doit présenter un support 8cm*8cm recouvert de scratch.

- Le **corps du robot** renferme les cartes électroniques, ainsi que l'alimentation.

- Le **pare-choc** sert à actionner le mécanisme des bouteilles. En effet, nous avons pu remarquer suite à plusieurs tests qu'il fallait une surface plane d'une certaine largeur pour pouvoir être sûr d'actionner correctement les mécanismes.

- On trouve des **capteurs SHARP** à l'avant et à l'arrière du robot. Pour pouvoir passer l'homologation dynamique et notamment la partie concernant l'évitement des autres robots, nous utilisons cinq capteurs SHARP. Deux sont positionnés en bais aux extrémités du pare-choc pour prévenir les collisions lorsqu'il tourne. Un capteur est positionné à l'arrière afin d'éviter les obstacles lorsque le robot recule. Ensuite, nous trouvons deux étages de capteurs sur l'avant du robot. En effet, après quelques tests, nous avons pu remarquer que, les bouteilles étant placées assez hautes par rapport à la table, le robot les détectait comme des obstacles et donc empêchait d'actionner le mécanisme. Aussi, si les deux capteurs détectent un objet, il faut s'arrêter, en revanche, si seulement le capteur du bas détecte quelque chose, le robot peut continuer sa route.

- Les **capteurs de couleur** sont placés entre le corps du robot et le pare-choc, afin de suivre au mieux la ligne noire sur la table de jeu. Etant donné que nous ne les utilisions plus lors de la Coupe de France et comme nous avons rencontré des difficultés avec les fixations des capteurs, nous avons les avons retirés pour les matchs.

- La **batterie** est positionnée au centre du robot afin d'éviter tout basculement vers l'arrière de ce dernier.

- Pour faire avancer le robot, nous utilisons des **servomoteurs** (cf glossaire) à rotations continus

Dans le corps du robot, nous y trouvons trois cartes, que nous détaillerons dans la partie électronique, **une carte de connecteur** où est placés l'Arduino Nano (cf glossaire) et **deux cartes « régulateur 6V » et « régulateur 5V »**. La première carte est fixée à l'alimentation grâce à du scratch, ainsi elle est facilement amovible et accessible en cas de problème. Les deux autres cartes sont fixées sur le plexiglas qui habille la structure en bois du robot.

Le dessus du robot (*figure 26*) est constitué de **quatre interrupteurs** : un interrupteur logique (ON/OFF de l'Arduino), un interrupteur qui détermine la zone de départ de l'équipe (rouge ou violet), un interrupteur pour calibrer les capteurs de couleur et un autre pour calibrer les capteurs SHARP. Nous y trouvons également le **bouton d'arrêt d'urgence**, obligatoire pour l'homologation statique et dynamique : il permet de stopper toute progression du robot. Nous pouvons également voir la **tirette de démarrage** qui permet de lancer le robot en début de match. Cette dernière est également indispensable pour l'homologation statique et dynamique.

4.3 ELECTRONIQUE

4.3.1 Alimentation

Nous avions initialement choisi d'alimenter le robot secondaire avec des piles 9V. Cependant, nous avons opté pour une batterie 12V, car les piles étaient trop longues à charger et ne présentaient pas une autonomie très grande, notamment lors de la phase de test. Grâce à la nouvelle alimentation installée, nous obtenons une autonomie d'en moyenne 3h, ce qui est amplement suffisant pour les exigences de la coupe. En effet, il est demandé que le robot ait une autonomie pour au moins deux matchs soit 3 minutes.

4.3.2 La carte Arduino

La carte Arduino (*figure 27*) comporte, comme son nom l'indique, l'Arduino Nano. Cette carte sert essentiellement à connecter les pins (cf glossaire) du microcontrôleur (cf glossaire) aux divers capteurs qui constitue le robot. Nous avons dû réaliser plusieurs versions de cette carte, en raison d'ajout de capteurs.

Figure 27: *La carte Arduino*

4.3.3 Les cartes « régulateur 6V » et « régulateur 5V »

Les cartes « régulateurs » servent ici à réguler la tension appliquée en entrée des servomoteurs (cf glossaire) et des cartes électroniques. En effet, le régulateur (cf glossaire) déjà intégré à l'Arduino Nano délivre un courant qui n'est pas suffisant pour alimenter les servomoteurs. Aussi, nous avons dû ajouter une carte « régulateur 6V » : elle délivre donc une tension stable à 6V.

La carte « régulateur 5V » a quant à elle été créée et finalement placé dans l'architecture électronique du robot pour des raisons de sécurité. En effet, nous avions prévu que l'Arduino (cf glossaire) et les capteurs pourraient ne pas supporter les pics de courant induits par l'alimentation : celle-ci délivre une tension 9V -12V non régulée. Aussi, nous avons placé cette carte, qui délivre donc une tension stable à 5V.

4.3.4 Les capteurs de lumière

Comme nous l'avons dit précédemment, nous avions initialement prévu d'utiliser des capteurs de lumière, que nous avons réalisé nous-mêmes et ce pour trois raisons.

Tout d'abord, les capteurs de lumière que nous pouvons trouver dans le commerce fonctionnent dans l'infrarouge. Or, la ligne noire est peinte sur un fond bleu, sur la table des matchs. Aussi, nous avons préféré utiliser des capteurs dans le vert-turquoise, afin de mieux discerner la ligne.

Ensuite, l'un des principaux problèmes avec les robots suiveurs de ligne est l'éclairage ambiant. En effet, lors de la Coupe, les tables sont exposées aux spots lumineux. Comme nous avons pu le constater pour d'autres équipes, qui avaient la même stratégie que nous, les capteurs exposés à tant de lumière voient leurs mesures faussées. Nous avons donc créé des cônes, réalisés à l'imprimante 3D du FalbLab, quasiment imperméables à la lumière ambiante.

Enfin, en raison de la luminosité ambiante, nous avons choisi d'avoir des capteurs calibrables. Nous pouvions alors nous adapter à tout type d'éclairage.

Chaque capteur de luminosité est constitué d'un coupe diode / photodiode fonctionnant donc dans le vert-turquoise. Le capteur est alimenté en 5V et renvoie une tension aux bornes de la photodiode proportionnelle à la luminosité mesurée. Ils sont rattachés à des cartes capteurs qui sont elles-mêmes connectées à la carte Arduino.

4.3.5 Les capteurs SHARP

Comme nous l'avons dit précédemment, les capteurs SHARP utilisés sont du même modèle que ceux utilisé pour le robot principal. La stratégie d'évitement est la suivante : si l'un des capteurs détecte un obstacle, il s'arrête, jusqu'à ce que l'obstacle en question ne soit plus détecté.

4.3.6 Le câblage

Les connecteurs utilisés pour le câblage du robot ont été choisi afin que tout membre du club puisse brancher et débrancher ses circuits électriques. En effet, les connecteurs sont chacun muni d'un détrompeur, qui empêche le branchement des câbles dans le mauvais sens. En outre, nous avons établi un code de couleur qui permet de faire correspondre des câbles avec des emplacements sur les cartes électroniques.

5. EVALUATION DES PERFORMANCES

5.1 COMMUNICATION

La communication a été un aspect très important de notre projet puisque c'était la demande principale de notre cliente M.C. Mouchot. Elle s'est faite sous différents aspects et à travers différents médias.

5.1.1 Internet

Nous disposons d'un site web (*figure 28*) **[3]** qui nous permet d'une part de partager nos connaissances avec les autres équipes grâce à un wiki (cf glossaire) et d'autre part de permettre à tout le monde de suivre nos aventures via le flux RSS des nouvelles (aussi affichées en page d'accueil).

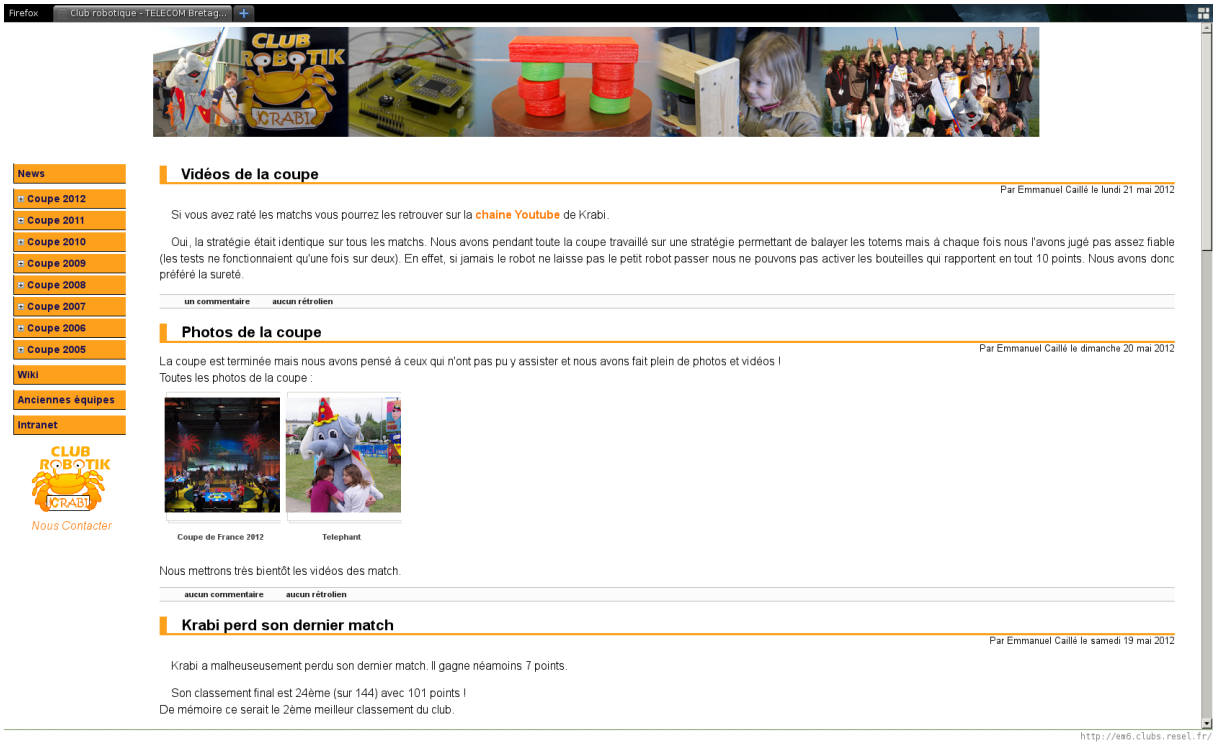


Figure 28: *Le site web du club*

Nous avons ainsi, pendant toute la coupe, publiée en temps réel les résultats de Krabi et son classement. Le site contient également le résumé de la coupe, la composition de l'équipe et plein d'autres informations utiles.

Pour illustrer tout cela nous avons aussi publié sur YouTube des vidéos de démonstration et les vidéos de tous les matches. Il y a également plusieurs albums photos sur Picasa : on y retrouve la construction progressive du robot, les photos des évènements .... Ces publications sont annoncées sur le site et sur le réseau social Google + où le club a une page **[4]**.

5.1.2 Presse

Un reportage sur le club a été fait par un journaliste du journal local *Sept Jours À Brest*, il a été publié le 09 Mai 2012 **[5]**.

5.1.3 Interventions

Certains membres du projet interviennent au Collège Kerallan pour assister des collégiens lors de leur construction de robot Lego pour leur participation à la coupe RoboFesta, une coupe inter collèges et lycées dans le Finistère.

5.1.4 Manifestations et démonstrations

Le club participe à plusieurs évènements où il représente fièrement Telecom Bretagne :

La fête de la science *[6]* :

Lors de cette manifestation nationale nous tenons un stand à l'UBO où nous présentons au publique tous nos robots : du gros robot au petit robot Lego. Le public très varié rend cette manifestation très enrichissante et nous permet de montrer tous les aspects de la robotique, du général aux détails techniques (*figure 29*).

Figure 29: *Le robot Lego présenté à la fête de la science*

Les portes ouvertes de Telecom Bretagne :

Cette journée est l'occasion pour nous de montrer aux futurs élèves ingénieur ce que peut être la vie associative d'un campus.

La pré-coupe du Trégor :

Organisée par l'ENSSAT à Lannion, c'est une forme de répétition pour la Coupe de France ; elle permet également de voir où on en est par rapport aux autres équipes et de comparer nos solutions technologiques.

Les cordées de la réussite *[7]* :

Un des groupes des Cordées sud-Finistère (cf glossaire) avait pour thème la robotique et était tutorés par Léa, membre du projet. Nous avons également présenté notre club lors de la visite de l'ensemble des cordées à Telecom.

Figure 30: *La Coupe de France de Robotique*

La Coupe de France de Robotique *[1]* :

Cet événement national (international cette année puisqu'Eurobot avait lieu au même endroit) est le point culminant du projet. Pendant 72h nous travaillons et faisons concourir nos robots face à plus d'une centaine d'autres équipes qui sont autant de concurrents que d'allié prêt à nous aider ou nous conseiller (*figure 30*). Vous trouverez plus de détails dans la partie suivante.

Le Forum S4 :

Marquant la fin du projet, ce forum nous permet de faire la démonstration de notre robot devant de nombreux industriels.

Ces manifestations, toutes aussi différentes les unes des autres, nous permettent de découvrir différents publics et différentes attentes, il nous faut adapter la communication à chacune d'entre elles pour toujours satisfaire nos interlocuteurs.

5.2 COUPE DE FRANCE

Après vous avoir présenté les différentes étapes qui ont permis de mener à bien notre projet, nous allons à présent vous montrer, de manière plus concrète, les capacités et les performances que notre robot a pu dévoiler au cours de la Coupe de France de Robotique. Pour cela, nous allons tout d'abord expliquer le déroulement de la coupe et l'organisation que nous avons mise en place au sein du groupe afin d'obtenir les meilleurs résultats possibles. Nous allons ensuite décrire les phases d'homologation ainsi que nos différents matchs de qualification. Enfin, nous ferons un bilan de cette Coupe et nous discuterons des résultats obtenus.

5.2.1 Déroulement

La Coupe de France était organisée sur quatre jours. Le premier était principalement destiné à la mise en place du matériel sur les stands mais aussi et surtout à l'homologation, c'est-à-dire à la vérification officielle des robots, afin de s'assurer que les normes imposées par le règlement étaient bien respectées. Les trois autres jours ont été consacrés aux matchs de qualification durant lesquels les différentes équipes se sont affrontées. L'objectif étant de marquer le plus de points possible au cours de cinq matchs. Les 16 équipes ayant obtenu les scores les plus élevés se sont qualifiés pour les phases finales. A ce stade, des 8^{e} de finale à la finale, les matchs éliminatoires ont permis de trouver la meilleure équipe du tournoi, à savoir RCVA, le club de robotique de l'IUT de Ville d'Avray.

Par ailleurs, en ce qui concerne plus particulièrement notre équipe, l'organisation générale de la coupe nous a permis de travailler dans un environnement plutôt adapté aux différentes tâches qu'il nous fallait accomplir. Trois tables de jeu ont été mises à la disposition des différentes équipes afin de réaliser des tests personnels ainsi qu'un atelier disposant d'un grand nombre d'outils et de composants électroniques pour satisfaire aux besoins de chacun. De notre côté, nous avions emmené la quasi-totalité de notre matériel ainsi que notre propre table pour être à même d'améliorer ou de corriger certaines parties de notre robot, sur un plan mécanique, informatique ou électronique. Afin de travailler de manière efficace et optimale, nous nous sommes organisés en deux sous-groupes pour permettre un roulement au cours la nuit. Chaque sous-groupe (de 21h à 4h pour le premier et de 4h à 11h pour le deuxième) était composé d'au moins un membre de chaque pôle pour palier toute éventualité. Bien évidemment, les deux sous-groupes étaient présents toute la journée mais ce système nous a permis de travailler à temps plein sur le robot tout en nous assurant un repos minimum.

5.2.2 L'homologation

L'homologation a été pour nous l'étape clef de cette Coupe de France car c'est elle qui nous a ouvert l'accès aux phases de qualification et qui nous a ainsi donné une chance de prétendre au titre. C'était bien là de l'objectif principal de notre projet et nous étions fins prêts bien avant notre arrivée à La Ferté Bernard. En effet, grâce à l'avance que nous avions prise, nous avons pu effectuer tous les tests nécessaires afin de garantir notre homologation et de mettre en place des stratégies pour remporter nos matches. Cela a été grandement facilité par la mise en place d'un simulateur informatique, la qualité de nos cartes électroniques et notre mécanique irréprochable.

Pour toutes ces raisons, nous nous sommes présentés pour l'homologation de notre robot dès le premier jour. Celle-ci s'est effectuée en deux temps, obligatoires et nécessaires : une phase statique et une phase dynamique. La partie statique, qui consiste à vérifier les bonnes dimensions de nos robots (chacun individuellement et les deux ensemble afin de respecter la surface de la zone de départ), leur respect des normes en matière de cordon de démarrage, de bouton d'arrêt d'urgence et de support balise, a été très rapidement couronnée de succès. En revanche, la partie dynamique, qui implique de remporter un match contre un adversaire fictif tout en évitant toute collision avec celui-ci ainsi qu'avec la table de jeu, a été un peu plus ardue. Étant donné que notre propre table de jeu n'était pas exactement conforme à la table de jeu officielle, nous avons dû corriger certaines parties du code informatique pour adapter les mouvements de nos robots à cette nouvelle table. Néanmoins, nous avons très vite corrigé les erreurs et nous avons validé la partie dynamique le soir même. Une fois l'homologation obtenue, nous avons pu nous préparer pour les premiers matchs de qualification.

5.2.3 Les qualifications

Le premier match a eu lieu le deuxième jour. Après une nuit passée à améliorer notre stratégie, nous nous sommes présentés avec une stratégie simple mais fiable pour le robot principal. Le robot secondaire, quant à lui, était entièrement codé en temporisations, c'est-à-dire qu'il effectuait des mouvements en un temps fixe et précis, étant donné que les capteurs couleurs permettant de suivre les lignes ont cessé de fonctionner une semaine avant la coupe. Son objectif étant de dévoiler les deux bouteilles à la mer et ainsi d'accumuler un grand nombre de points. Nous avons gagné le premier match avec un total de 24 points (14 points marqués et 10 points de victoire). Nous avons d'ailleurs largement partagé cette victoire avec nos tuteurs Magali LE GALL, Noël CAILLERE et Jean-Pierre CLERE qui étaient présents lors de ce premier match.

Trois autres matchs étaient prévus pour le troisième jour. La stratégie du robot principal n'étant pas encore tout à fait aboutie, nous nous en sommes tenus à notre stratégie plus simple de base. Par contre, nous avons élaboré une nouvelle stratégie pour le robot secondaire afin de lui permettre de rapporter un lingot jusqu'à notre navire, en plus des deux bouteilles. Nous sommes également sortis victorieux de ces trois matchs et nous étions classés 11^{e} (sur les 140 équipes homologuées) juste avant le cinquième et dernier match. Nous étions donc très bien parti pour figurer parmi les 16 finalistes.

Le cinquième match s'est déroulé le dernier jour. Après avoir réussi à améliorer la stratégie du robot principal, nous disposions alors d'une stratégie fiable mais uniquement du côté rouge de la table. Cela étant lié à des problèmes d'asymétrie des roues codeuses (cf glossaire) que nous connaissions bien, mais que nous n'avons malheureusement pas eu la possibilité de régler du fait d'un manque de moyens financiers. De plus, le hasard nous a conduits à commencer du côté violet de la table lors de ce dernier match. Nous nous en sommes donc tenus à la stratégie de base pour le robot principal et nous avons perdu ce match. Nous avons tout de même réalisé un très bon score qui nous a valu la 24 e place dans le classement final.

5.2.4 Bilan de la Coupe 2012

A l'issue de cette Coupe de France, nous sommes plutôt satisfaits des résultats que nous avons obtenus et, comme l'objectif d'homologation de nos robots a été facilement atteint, nous pouvons dire que cette compétition a été pour nous un franc succès. C'est le résultat de notre organisation en amont de cette coupe qui nous a permis de tester nos robots et de les confronter à la table de jeu et ses divers éléments, mais aussi de notre organisation pendant l'épreuve qui nous a permis de corriger certaines erreurs et d'affiner nos stratégies. Enfin, les architectures mécanique, informatique et électronique se sont révélés plutôt fiables et nous ont permis d'obtenir une place très honorable.

5.3 CAPITALISATION DES COMPETENCES ET DOCUMENTATION

Un des points essentiels pour mener à bien notre projet a consisté à pouvoir bien réutiliser l'expérience et les connaissances des équipes précédentes. En effet, au sein du club robotique nous avons à notre disposition un certain nombre d'outils permettant de garder une trace de nos travaux actuels ainsi que des travaux des années précédentes. On peut notamment y trouver des schémas électroniques, des codes informatiques, des plans mécaniques mais également l'évocation de certains problèmes rencontrés et parfois même leurs solutions.

Parmi ces différents outils, on trouve tout d'abord les rapports des années précédentes qui nous offrent une documentation très détaillée des anciens robots et de leur conception. Ils nous permettent de reprendre certaines solutions techniques et ainsi éviter de perdre du temps pour des erreurs récurrentes ou habituelles.

D'autre part, nous pouvons compter sur l'actualisation du Wiki du Club (cf glossaire) qui retrace l'avancée de nos travaux ainsi qu'un certain nombre d'explications sur notre code informatique, sur l'architecture électronique et sur des parties mécaniques. Cette page internet étant accessible à tous depuis le site de notre club.

Par ailleurs, nous possédons une mailing-liste qui regroupe l'ensemble des membres du club ainsi qu'un grand nombre d'anciens étudiants qui nous donnent régulièrement de très bons conseils et des informations pertinentes relatives à la robotique.

Enfin, comme le partage des connaissances avec les étudiants de première année est une des missions principales de ceux de deuxième année, nous avons formé des étudiants au sein de chaque pôle et donc assuré notre relève.

En conclusion, nous pouvons dire que tous ces outils et ces démarches pour assimiler et transmettre les compétences accumulées depuis la création du club ont été déterminants dans notre réussite lors de la Coupe de France de Robotique. En effet, nous avons ainsi pu disposer d'une bonne base technique qu'il nous a simplement fallu perfectionner et adapter au nouveau règlement et donc aux nouveaux objectifs. De plus, certaines erreurs passées, comme les surtensions créées dans les ponts en H, ont été corrigées. C'est pourquoi, étant donné la motivation et le bon déroulé des formations des premières années, nous sommes être très confiants pour le club et les années à venir.

5.4 BILAN BUDGETAIRE

5.4.1 Démarchage

Notre politique de démarche fut cette année de contacter nos précédents sponsors : Brest Métropole Océane, la Société Générale et Télécom Bretagne. Nous avons aussi tenter d'établir sans succès des partenariat avec d'autre organisation, notamment Thales qui soutient chaque année plusieurs équipes participants à la coupe.

5.4.2 Bilan de Trésorerie

Les dépenses et les recettes du club Robotique sont présentées *figure 31* et *figure 32*.

Electronique	
Composants	750,00 €
batteries	150,00 €
sous total	900,00 €
Informatique	
Microcontrôleurs	200,00 €
sous total	200,00 €
Mécanique	
Outils et matériaux	300,00 €
Table	200,00 €
sous total	500,00 €
Frais lors de la Coupe de France	
Déplacement	350,00 €
Logement	300,00 €
Nourriture	250,00 €
sous total	900,00 €
Divers	110,00 €
Total	**2 610,00 €**

Figure 31: *Dépenses*

Recettes	
BMO	610
Télécom Bretagne	1000
Société Générale	1000
Total	**2610**

Figure 32: *Recettes*

Conclusion

Cette année, afin d'améliorer nos performances à la Coupe de France, nous avons opté pour une stratégie et une architecture certes plus simple mais fiable. Aussi, nous sommes très satisfaits des résultats obtenus qui se placent bien au dessus de nos espérances. Cette expérience nous a permis d'observer qu'il suffit de réaliser un produit simple pour avoir des performances tout a fait satisfaisantes.

Nous avons pleinement rempli le cahier des charges et nous sommes confiants quant à l'avenir de ce projet. En effet, étant donné nos aboutissements cette année, nous pouvons réaliser une bonne passation des connaissances. Les années futures vont pouvoir utiliser nos travaux afin de pouvoir innover sur une base sûre.

GLOSSAIRE

Arduino : microcontrôleur programmable pouvant générer et recevoir pour analyse de signaux électriques.

Buffer : montage électrique spécifique permettant de diminuer le courant de sortie d'un circuit, en adaptant l'impédance.

Microcontrôleur : circuit intégré qui comporte un microprocesseur, de la mémoire vive (RAM), de la mémoire morte (ROM) et des pins Entrée/Sortie. Un microcontrôleur est programmable et permet de concevoir des systèmes embarqués autonomes tels que le robot.

STM32 (ou simplement STM) : microcontrôleur qui constitue l'intelligence du robot. Il permet de gérer entre autre les différents capteurs du robot.

Optocoupleur : composant électronique permettant de transmettre un signal d'un circuit électrique à un autre sans qu'il y ait de contact direct entre eux.

Pin : Il s'agit des broches (ou pattes) des différents composants électroniques utilisés dans le robot.

Pont en H : composant constitué de 4 transistors qui doivent gérer la puissante : il permet de choisir la valeur et le signe de la tension de sortie suivant les signaux d'entrées.

PWM : (*Pulse Width Modulation*) signal périodique, en créneau. La fréquence du signal émis est fixée et on peut le moduler en modifiant tout simplement la largeur des créneaux.

Régulateur (linéaire) : composant électronique qui permet de fournir une tension régulée : par exemple un régulateur 6V permet de fournir une tension 6V stable, indépendamment de la tension d'entrée.

Roue codeuse : roue équipée d'un dispositif électronique permettant de mesurer sa rotation.

Servomoteur : moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure. Il s'agit d'un système asservi.

Krabi : Nom du robot principal présenté cette année.

Krabi Junior : Nom du robot secondaire présenté cette année.

Solidworks : outils de CAO, utilisé pour les modélisations 3D et les mises en plan du robot principal.

Wiki : site dont les pages peuvent facilement être modifiées par les visiteurs ***[8]***.

Cordées de la réussite : label créé par le ministère de l'enseignement supérieur pour encourager les projets visant à faire découvrir l'enseignement supérieur à des lycéens de milieux « défavorisés ». Plus d'infos sur le projet Finistère ***[7]***.

BIBLIOGRAPHIE

[1] http://www.planete-sciences.org/robot/index.php?section=pages&pageid=108
Site officiel de la coupe de France de Robotique édition 2012, réalisé par les organisateurs bénévoles de la coupe de France. Dernière consultation le 19/06/2012.

[2] http://www.planete-sciences.org/robot/data/file/coupe/2012/E2012_Reglement_FR_final.pdf
Règlement en français de la coupe de France de Robotique édition 2012, publié en octobre 2011 et disponible sur le site officiel de la coupe de France de Robotique. Dernière consultation le 19/06/2012.

[3] http://em6.clubs.resel.fr
Site du club robotique de Telecom Bretagne. Dernière consultation le 19/06/2012.

[4] https://plus.google.com/u/0/111085881977205266492
Page google + contenant certaines photographies et vidéos prises par les membres du club pendant la coupe de France de Robotique.

[5] http://www.septjoursabrest.fr/2012/05/09/robots-krabi-a-labordage-de-la-coupe-de-france/
Article de presse publié dans le journal 7 jours à Brest, écrit par Batiste Kolenc, le 09/05/2012. Dernière consultation le 19/06/2012.

[6] http://www.brest.fr/culture/tous-les-ans-a-brest/la-fete-de-la-science.html
Article publié sur le site officiel de BMO, expliquant ce qu'est la fête de la science. Dernière consultation le 19/06/2012.

[7] http://www.cpge-brizeux.fr/cordees/
Site de la CPGE de Brizeux, page web concernant les Cordées de la Réussite du Finistère. Dernière consultation le 19/06/2012.

[8] http://em6.clubs.resel.fr/dokuwiki/doku.php
Wiki du club de robotique de Telecom Bretagne contenant les informations nécessaires aux années qui vont suivre pour perpétuer les travaux du club. Dernière consultation le 19/06/2012.

[9] www.arm.com/files/pdf/introToCortex-M3.pdf
Document de présentation de la structure processeur ARM Cortex M3. Dernière consultation le 10/06/2012.

[10] http://www.ti.com/lit/ds/symlink/ptn78020w.pdf
Document constructeur du régulateur PTN 78020W. Dernière consultation le 10/06/2012.

[11] http://www.datasheetcatalog.org/datasheet/siemens/ILD620.pdf
Document constructeur de l'optocoupleur ILD620. Dernière consultation le 10/06/2012.

[12] http://pdf1.alldatasheet.com/datasheet-pdf/view/22621/STMICROELECTRONICS/L78S05C.html
Document constructeur du régulateur de tension 78S05. Dernière consultation le 10/06/2012.

[13] http://pdf1.alldatasheet.fr/datasheet-pdf/view/27517/TI/SN54ABT827.html
Document constructeur du buffer SN54ABT827. Dernière consultation le 10/06/2012.

[14] http://www.datasheetcatalog.org/datasheet/stmicroelectronics/1373.pdf
Document constructeur du pont en H L6203.

Annexes

Les annexes ont été réalisées conjointement avec les membres du projet s2 de robotique.

ANNEXE 1 - CAHIER DES CHARGES FONCTIONNELLES

A1.1. LE PROJET ROBOTIQUE

A1.1.1. FINALITES

Le but de ce projet est, avant toute chose, de concevoir deux robots autonomes, pouvant toutefois communiquer entre eux, devant être tous deux homologués à la Coupe de France de Robotique 2012 (se déroulant du 16 au 19 mai à la Ferte-Bernard).

Afin d'être homologués, ces derniers doivent respecter plusieurs conditions. D'une part, ils doivent satisfaire les normes officielles de la compétition, détaillées dans le règlement de la Coupe *[2]*. Ceci constitue la pré-homologation (ou homologation statique). D'autre part, les deux robots doivent valider trois compétences chacun (ceci constitue l'homologation dynamique):

- Etre capable de remporter un match sans adversaire
- Etre dans la mesure d'éviter le (ou les) robot(s) adverse(s)
- Pouvoir être arrêté en cas de fonctionnement anormal, à l'aide d'un bouton d'arrêt d'urgence.

A1.1.2. ESPERANCE DE RETOUR SUR INVESTISSEMENT

Nous espérons obtenir le meilleur résultat possible à la Coupe. Ceci signifie non seulement que l'on compte avoir un meilleur résultat que l'an dernier (ce qui, nous l'espérons, se réalisera, bien sûr) et d'autres part finir parmi les cinquante premières équipes, jusqu'à même faire partir du tableau final de la coupe (ce qui correspond aux seize premières équipes).

Par ailleurs, le fait de réaliser un bon classement contribue directement à l'amélioration du prestige de l'École.

A1.2. CONTEXTE

A1.2.1. SITUATION DU PROJET PAR RAPPORT AUX AUTRES PROJETS DE L'ECOLE

Le Club Robotique de Télécom Bretagne prépare chaque année la Coupe de France de Robotique et y participe. Notre projet s'inscrit donc logiquement dans le cadre de cette préparation et de cette participation.

Par ailleurs, il existe une collaboration active avec le projet s2, qui est constitué de Gaëtan FAYON, Samuel BARAUD, Qiao WANG et Victor DUBOIS.

A1.2.2. ETUDES PREALABLEMENT EFFECTUEES

Les participants aux projets robotiques précédents ont préalablement travaillé sur des sujets semblables aux nôtres. Nous bénéficions ainsi des connaissances acquises lors des précédentes coupes et ainsi les études sur ces sujets seront prises en compte dans la réalisation de notre projet. Par exemple, nous nous basons sur les cartes électroniques des années précédentes (que nous adapterons et améliorerons). Quant à la base roulante, son fonctionnement et son architecture restent également basés sur le principe des années précédentes, de la même manière qu'une partie du code informatique (sur la stratégie d'évitement ou sur la façon d'avancer notamment) est repris années après années compte tenu du fait que certains principes de la Coupe restent inchangés. Bien entendu, nous y adapterons les concepts nouveaux mis en place par les organisateurs cette année.

A1.2.3. NATURE DES PRESTATIONS DEMANDEES

Il est demandé aux membres du Club Robotique, et à fortiori notre groupe travaillant sur ce projet, de concevoir et de réaliser les deux robots dans leur intégralité (nous avons en effet décidé cette année d'utiliser deux robots, comme l'autorise le règlement pour la première fois depuis 2005) à la fois d'un point de vue informatique, électronique et mécanique.

Par ailleurs, en plus de devoir assurer l'homologation des robots et leur participation à la Coupe, un plan de Management, des rapports d'avancement hebdomadaires, une plaquette de promotion, des posters et un rapport technique, seront également à fournir.

A1.2.4. PARTIES CONCERNEES PAR LE DEROULEMENT DU PROJET ET SES RESULTATS (DEMANDEURS, UTILISATEURS)

Notre projet concerne dans un premier temps les deux groupes travaillant sur le Projet Robotique (le groupe en semestre 2 et celui en semestre 4). Il concerne de manière plus générale le Club Robotique de l'École et l'ensemble de ses membres, les robots adverses (et les clubs qu'ils représentent) qui participeront à la Coupe de France de Robotique 2012, sans oublier les organisateurs de cette Coupe, en charge de l'homologation du robot. Enfin, de manière encore plus générale, il concerne également Télécom Bretagne, via notamment la Direction de la Communication de l'École (notre client), et nos partenaires (entreprises, banques, etc.).

A1.3. ENONCE DU BESOIN

Cette partie s'intéresse aux finalités premières du produit, c'est-à-dire aux objectifs que les deux robots doivent accomplir pour le futur utilisateur, tel que prévu par l'utilisateur.

Chaque année, un concours de robotique est organisé par l'association Planète Sciences et le Conseil Municipal de La Ferte-Bernard à la fin du printemps. Depuis maintenant plus d'une dizaine d'années, le Club Robotique de l'École se présente à ce concours prestigieux où le thème change tous les ans. Cette année, le thème retenu est « *Treasure Island* », et le but du jeu est de récupérer un maximum de pièces et de lingots éparpillés sur la table et de les ramener dans des zones spécifiques nous appartenant. Un autre objectif réside dans l'actionnement de mécanismes libérant des messages contenu dans des bouteilles, et de dévoiler une carte au trésor en retirant le voile qui la cache. L'aire de jeu ressemble au final à ceci :

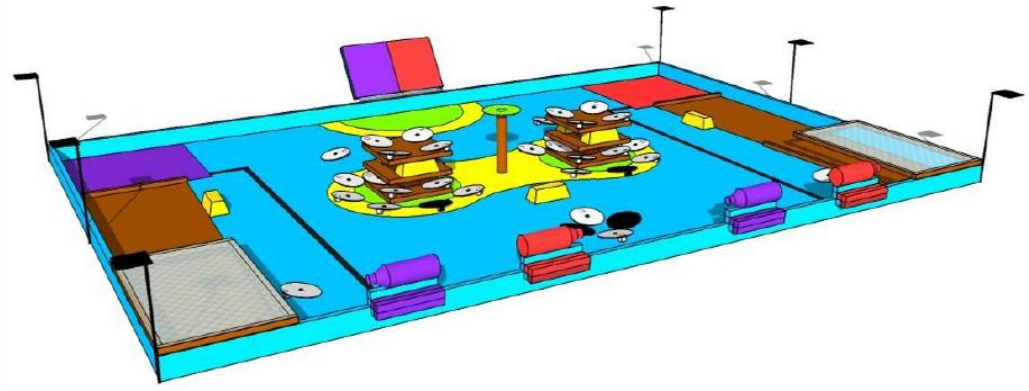


Figure 33: *Table de jeu*

Notre besoin est assez clair : concevoir deux robots autonomes, homologables, compétitifs et qui respectent l'intégralité du règlement imposé par les organisateurs de la Coupe de France de Robotique 2012, ainsi que de réaliser la table de jeu et l'ensemble des éléments qui la compose.

De cette manière, notre robot principal doit être adapté au règlement en place cette année en étant notamment capable de ramasser des pièces et des lingots sur des endroits précis de la table de jeu en les amenant ensuite sur une zone qui nous est réservé. Notre robot secondaire doit être notamment capable de déclencher le mécanisme de libération des bouteilles situées en bord de table.

A chaque action réalisée correspond un nombre précis de points, le total concernant notre robot devant être supérieur à celui concernant le robot adverse. Le tout doit être réalisé en moins de 90 secondes (durée d'un match).

A1.4. ENVIRONNEMENT DU PRODUIT RECHERCHE

L'environnement du robot est composé de :

- **Coupe de Robotique** : celle-ci se déroule du 16 au 19 mai 2012 à La Ferte-Bernard. Les robots ayant réalisés les meilleurs performances gagneront la coupe, et pourront ainsi participer à Eurobot (Coupe d'Europe de Robotique).

- **Jury** : c'est la personne qui vérifie si le (ou les) robot(s) respecte(nt) les conditions d'homologation, et qui est de ce fait capable de participer à la Coupe. Lors de l'homologation, le(s) robot(s) devra(ont) passer deux examens pratiques : l'homologation statique et l'homologation dynamique.

- **Règlement** : c'est l'ensemble des règles qui définissent les caractéristiques du robot, les dimensions de l'aire de jeu et le déroulement de la compétition.

- **Equipe** : c'est l'ensemble des membres de l'équipe du Projet Ingénieur s4 qui participera à la Coupe de France de la Robotique 2012 (accompagné de l'équipe du Projet Robotique du semestre 2 et d'autres membres du Club Robotique).

- **Budget** : c'est la somme d'argent qui nous est consacrée pour le projet.

- **Alimentation** : cela concerne la source d'énergie principale du robot, permettant de l'alimenter. Les sources d'énergies prohibées sont indiquées à la page 21 du règlement ***[2]***.

- **Environnement** : Il s'agit ici de l'ensemble des éléments physiques qui interagissent avec le robot lors d'un match. Cela concerne principalement la table de jeu et les éléments qui la constituent (totems, lingots et pièces par exemple). La table est de forme rectangulaire, d'une longueur totale de 3 mètres, et d'une largeur de 2 mètres. Le thème de cette année – *Treasure Island* – possède une influence directe sur la forme de l'aire de jeu globale, comme nous pouvons le voir sur la modélisation de la table ci-dessous :

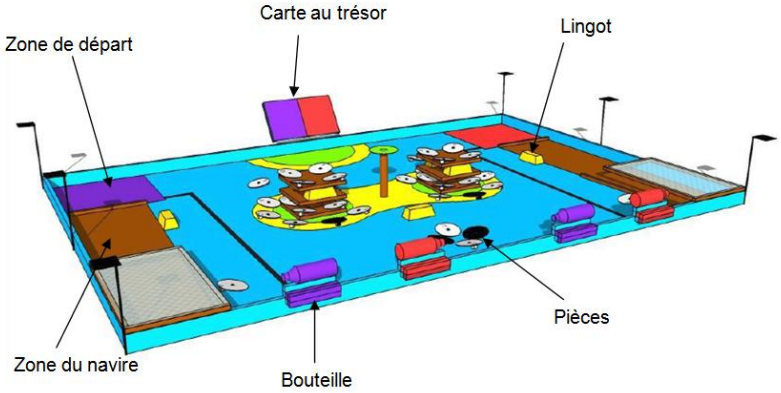


Figure 34: *Table de jeu*

Les éléments à mouvoir sur l'aire de jeu peuvent être divisée en deux catégories :

- **Les pièces**, fabriquées à partir de CD-ROM de taille standard (12 cm de diamètre). De couleur blanche, chacune rapport 1 point si elles sont mises dans les cales qui correspondent à notre bateau (sur les bordures de la table). A contrario, de couleur noire, elle ne nous rapporte aucun point.

- **Les lingots**, de couleur jaune, nous rapportent 3 points lorsque qu'ils sont placés dans une zone qui nous est réservée.

Les éléments interactifs du jeu sont de deux sortes :

- **Les bouteilles** : le but est d'ouvrir les bouteilles en poussant une sorte de bouton poussoir sur une des bordures de la table. Chaque équipe possède deux bouteilles à valider et chaque bouteille validée remporte 5 points.

- **La carte** : le but est ici de « découvrir » la partie de la carte correspondant à la couleur de son équipe. Au début du match, cette carte est complètement couverte par deux morceaux de tissus scratchés sur le support de la carte. Un morceau de carte découvert vaut 5 points.

- **Robot adverse** : La compétition se déroule en même temps contre un autre robot qui doit lui aussi marquer des points. Le but du jeu est alors de collecter plus de points que le robot adverse.

A1.5. FONCTIONS DE SERVICE ET DE CONTRAINTES

A1.5.1. FONCTIONS DE SERVICE PRINCIPALES

Il s'agit ici des fonctions qui sont la raison d'être du produit. Elles découlent ainsi directement de notre but premier, c'est-à-dire être homologué à la Coupe de France de Robotique 2012.

FP1 : Déplacer les pièces et les lingots

Les robots doivent être capables de déplacer les pièces d'un point précis jusqu'à notre zone de marquage. Cela sous entend directement que le robot doit au moins capable de se déplacer. Cette fonction permet de s'assurer la validation de deux points importants de la phase d'homologation dynamique :

- Les robots peuvent quitter leur zone de départ car ils peuvent se mouvoir.

- Ils peuvent gagner un match sans aucun adversaire s'ils sont capables de déplacer les pièces et les lingots (seuls éléments du jeu que nous sommes autorisés à déplacer). Evidemment, les robots doivent pouvoir établir une stratégie de déplacement afin de ramener les pièces et les lingots aux emplacements qui nous sont réservées et qui nous rapportent des points (les cales longeant les zones de départ), le tout en un temps raisonnable (90 secondes, c'est-à-dire la durée d'un match). Dans notre cas, cette tâche incombera en principe majoritairement au robot principal.

FP2 : Actionner les éléments interactifs (bouteilles et carte)

Les robots doivent être capables d'actionner les éléments interactifs de l'aire de jeu, à savoir actionner le mécanisme des bouteilles en bordure de table et/ou dévoiler la carte afin de marquer 5 points par éléments actionnés. Des deux robots, c'est le robot secondaire qui s'occupera en principe de cette tâche (il marquera alors les points nécessaires à son homologation de cette façon).

FP3 : Repérer et éviter un obstacle

Chaque robot doit être capable de repérer un obstacle (les robots adverses, les deux totems, les bordures, l'arbre situé au milieu de l'aire de jeu, etc.) dont la position est dans le premier cas indéterminés et dans les autres cas entièrement connue. Une stratégie de déplacement doit alors être envisagée pour éviter ces obstacles. L'homologation exige que chaque robot soit en mesure d'éviter un cylindre de diamètre 20 cm et d'une hauteur de 30 cm (taille moyenne des robots principaux) : il est alors indispensable que les robots puissent identifier une telle forme pour le repérer.

A1.5.2. FONCTIONS DE SERVICE COMPLEMENTAIRES

Il s'agit maintenant d'étudier les fonctions qui améliorent, facilitent ou complètent le service rendu.

FS1 : Vaincre le robot adverse

Les deux robots doivent être capable de marquer plus de point que le(s) robot(s) de l'équipe adverse, afin d'assurer la victoire. Bien que la raison d'être de notre produit dans le cadre de ce projet est l'homologation, gagner la Coupe de France de Robotique 2012 (ou arriver le plus loin possible dans la compétition) est, de ce point de vue, plus secondaire, mais cela reste tout de même appréciable de finir bien classé dans la compétition. Cette fonction est très difficile à évaluer, car le nombre de points que les robots adverses vont gagner est très difficile à prévoir. Cependant, le but est d'optimiser la stratégie du robot afin d'assurer un maximum de points de notre côté.

FS2 : Repérer son environnement

Le robot peut utiliser des moyens lui permettant de repérer sa position sur la table ou les éléments (amovibles ou non) de l'aire de jeu, ceci dans le but d'améliorer ses déplacements et d'identifier l'emplacement des éléments ou leurs natures (pièces, lingots, etc.). Cette fonction est secondaire dans le sens où elle n'est guère indispensable pour assurer l'homologation : la détection est en effet seulement obligatoire dans le cas d'un obstacle, et il n'est pas non plus indispensable de repérer les éléments afin de les déplacer, compte tenu que nous connaissons l'emplacement des éléments à mouvoir au début du match.

A1.5.3. FONCTIONS DE CONTRAINTES

Etudions désormais les fonctions contraintes, c'est-à-dire les fonctions qui limitent la liberté du concepteur du produit.

FC1 : Respecter les échéances

Les robots doivent être homologués avant le 1er tour de qualifications et pouvoir participer à la Coupe du 16 au 19 mai 2012. Les différents livrables, comme les posters, le dossier technique, etc. doivent être déposés eux aussi à temps (avant l'homologation).

FC2 : Respecter le règlement

Les deux robots doivent respecter à la fois les règles et les conditions définies dans le règlement officiel de la compétition. Que ce soit au niveau des caractéristiques des robots ou des règles du jeu, chaque point doit être vérifié et respecté avec la plus grande attention, sous peine de ne plus pouvoir être homologable le jour de la compétition, ou de perdre des points au cours des différents matchs.

FC3 : Etre autonome

Après le début du match, et jusqu'à la fin, les robots doit travailler en parfaite autonomie, c'est-à-dire ne pas subir d'intervention extérieurs, aussi bien humaine que matérielle. La seule intervention humaine autorisée est celle induite par l'arbitre lorsqu'il actionne le bouton d'arrêt d'urgence d'un des robots sur la table. Ceux-ci doivent de plus s'arrêter automatiquement lorsque le match est fini, soit au bout de 90 secondes.

FC4 : Respecter le budget

Le budget utilisé pour concevoir et fabriquer les deux robots ne doit pas dépasser le budget de 1500€ mis à notre disposition. En réalité, ce budget inclut également les frais de déplacement et de logement nécessaire pour participer à la Coupe.

FC5 : Avoir une alimentation adaptée

L'énergie que fournit l'alimentation doit être suffisante pour que les robots puissent fonctionner durant les 90 secondes du match, plus un autre match, soit 180 secondes au total. L'alimentation doit également être rapidement rechargeable, c'est-à-dire que celle-ci doit être opérationnelle avant un match. La source d'énergie utilisée doit être autorisée par le règlement. Les sources utilisant des procédés chimiques ou pyrotechniques sont par exemple interdites.

Afin de résumer cette analyse fonctionnelle, voici le diagramme pieuvre du projet Robotique :

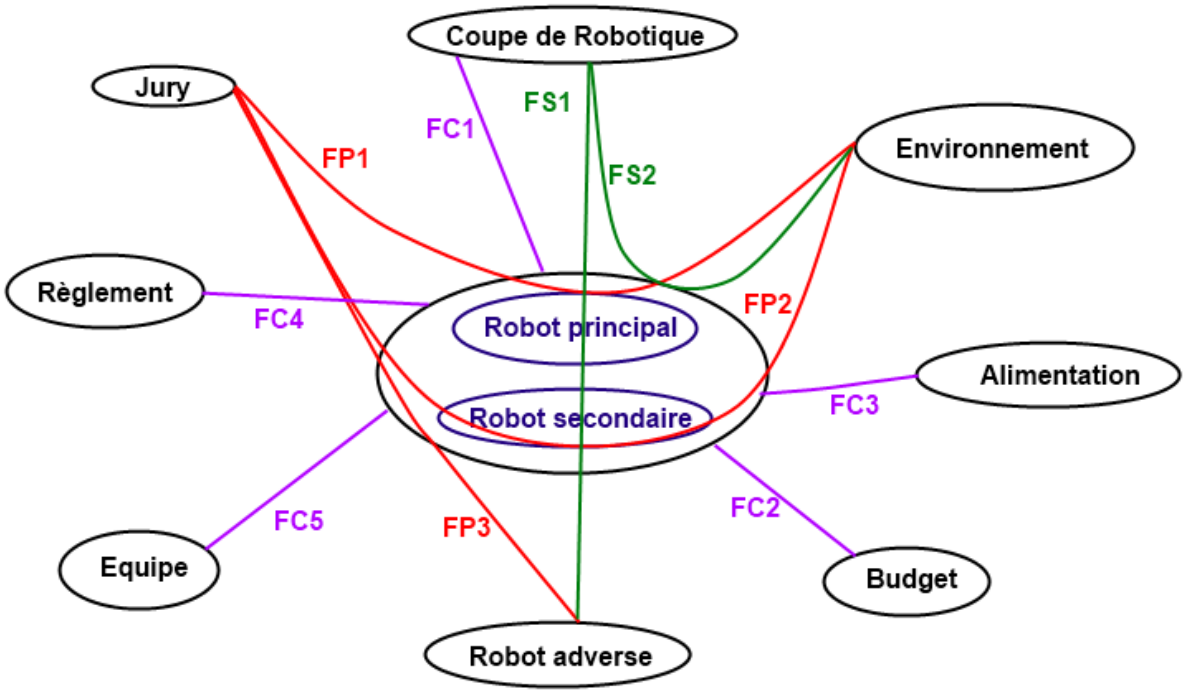


Figure 35: *Diagramme pieuvre du projet*

A1.6. CRITERES D'APPRECIATION

Le tableau qui suit présente dans son intégralité les critères d'appréciation relatifs aux fonctions précédemment énoncées. Sont aussi présents les différents niveaux de flexibilité (F0 = Impératif, F1 = Peu Négociable, F2 = Négociable et F3 = Très Négociable) pour chacun de ces critères d'appréciation, ainsi que les moyens de contrôle.

Fonctions	***Critère(s)***	***Niveau(x)***	***Flexibilité(s)***	***Moyen de contrôle***
FP1 Déplacer les pièces et les lingots	Distance de déplacement	0 à 4 mètres (ceci dépend de la position du robot et de l'élément à mouvoir)	F2	Outil de simulation ou mesure de distance avec un appareil adapté
	Position final de la pièce	Zones dédiées à nos dépôts d'éléments amovibles (les cales)	F0	Observation du comportement des robots en phase de test
	Temps de déplacement	1 à 90 secondes	F0	Montre ou chronomètre
FP2 Actionner les éléments interactifs	Force suffisante pour pousser le mécanisme d'ouverture des bouteilles	Entre 5 et 20 Newtons	F0	Observation du comportement des robots lors de phases de test, et vérification lors de la coupe
	Force suffisante pour dévoiler la carte	Entre 5 et 20 Newtons	F2	

FP3 Repérer et éviter un obstacle	Stratégie d'évitement	Aucun contact avec l'obstacle (typiquement un cylindre de 30 cm de diamètre et de 20 cm de hauteur)	F0	Observation du comportement des robots lors de phases de test
FS1 Vaincre le robot adverse	Points marqués	30 points pour espérer gagner	F3	Observation du comportement des robots lors de phases de test, et vérification lors de la coupe
FS2 Repérer son environnement	Position sur la table	Précision : 2 à 3 cm	F1	Observation du comportement des robots lors de phases de test
	Position des éléments	0 à 4 m par rapport au robot	F1	
	Nature des éléments	Forme globale différent pour chaque élément	F2	
FC1 Respecter les échéances	Date de la Coupe	Du 16 au 19 mai 2012	F0	Calendrier
	Date de l'homologation officielle	Le 16 mai 2012	F0	
FC2 Respecter le règlement	Caractéristiques des robots	Règlement page 17 à 32	F0	Lire le règlement
	Règles du jeu	Règlement page 7 à 16	F0	
FC3 Etre autonome	Autonomie	Aucune intervention extérieure	F0	Observation du comportement des robots lors de phases de test
	Durée	90 secondes	F0	
FC4 Respecter le budget	Budget	1500€	F2	Vérifier les comptes
FC5 Avoir une alimentation adaptée	Temps de recharge	Inférieur à 3 heures	F2	Temps affiché sur le chargeur
	Temps de décharge	Supérieur à 180 secondes	F0	Observation du comportement du robot lors des phases de test
	Source d'énergie réglementaire	Voir règlement à la page 21	F0	Lire le règlement

Figure 36: *Critères d'appréciation des fonctionnalités et contraintes du projet robotique*

ANNEXE 2 - PLANNING DU PROJET

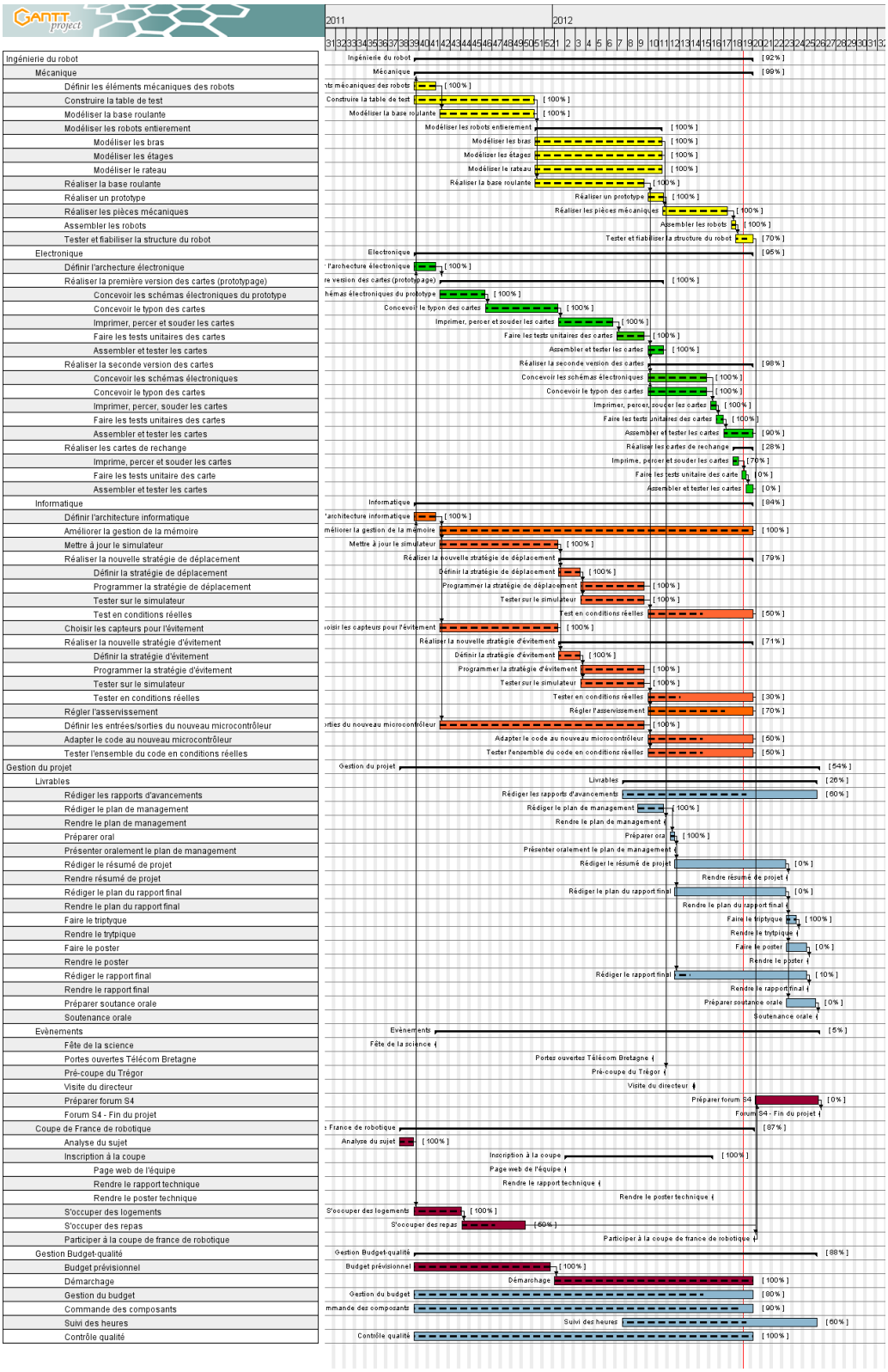


Figure 37: *Diagramme de Gant du projet*

ANNEXE 3 - LE SIMULATEUR DE DEPLACEMENT

Le débogage d'un programme devant tourner sur un microcontrôleur (cf glossaire) n'est la plupart du temps pas très simple. En effet, contrairement à un programme plus classique, on ne peut pas faire tourner le code sur nos ordinateurs personnels car celui-ci est compilé pour l'infrastructure processeur du STM32 (cf glossaire) (un processeur ARM-Cortex M3) mais surtout parce que le programme attend de recevoir des informations spécifiques de l'environnement de la carte électronique. Il s'offre alors à nous deux solutions.

On peut communiquer directement avec le microcontrôleur (cf glossaire) pour déboguer le code qui est exécuté dessus (obtenir les informations sur les registres, l'état de la mémoire, etc.). Cette méthode est indispensable pour débogue les parties bas niveau du code. Celles qui interagissent directement avec l'environnement électronique externe ou interne à la carte STM (comme l'asservissement).
La deuxième solution qui s'offre à nous est de simuler le comportement du microcontrôleur (cf glossaire) et d'exécuter le code du robot sur ce simulateur. Cette solution est très efficace pour tester le code haut-niveau, comme la stratégie, car, si le simulateur est bien codé, celui-ci reproduit parfaitement le comportement idéal que devrait avoir le robot en réponse à un ordre.

Le fonctionnement est assez simple. Il est codé en C++ tout comme le reste du code du robot. Toutes les parties faisant référence au STM (cf glossaire) dans le code sont remplacé par du code simulant le comportement réel du robot. Ceci est "facilement" réalisable grâce à l'utilisation du paradigme objet. Les parties du code échangées par une autre dans le simulateur utilisent les mêmes entrées/sorties dans le code du simulateur. De ce fait, le code haut-niveau n'a pas besoin d'être modifié pour pouvoir s'adapter au simulateur.

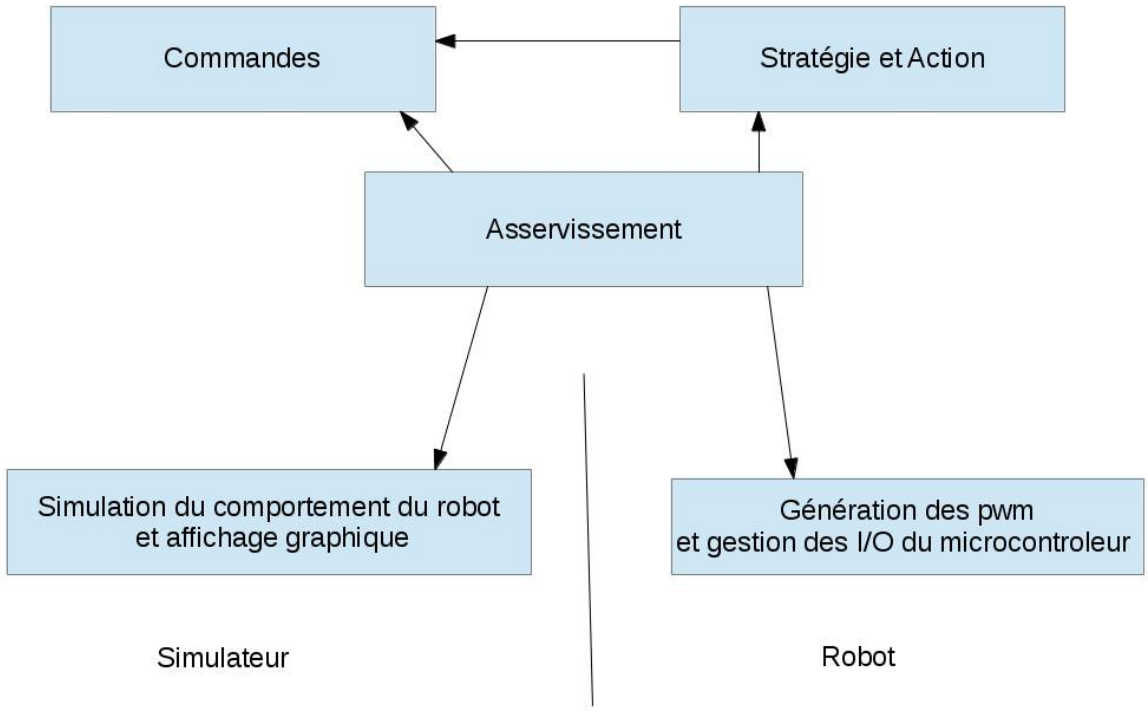


Figure 38: *Comportement du robot et simulateur*

Le simulateur est donc très pratique pour tester sans trop de contraintes les parties haut-niveau du code, mais ne permet en aucun cas de tester le code bas-niveau qui interagit avec le STM (cf glossaire) (*figure 37*). Il faut quand même faire attention au fait que certaines erreurs peuvent apparaître dans le simulateur du fait de quelques approximations physiques.

Afin de mieux représenter l'état actuel d'exécution du code sur le simulateur, nous avons implémenté une interface graphique avec QT qui permet de dessiner sur l'écran la position du robot sur la table ainsi que sa trajectoire passée. Nous utilisons aussi un simulateur de physique (à deux dimensions) box2D afin de mieux simuler interaction entre le robot physique et les éléments de jeux. Vous pouvez voir ci-dessous un aperçu de l'interface graphique du simulateur.

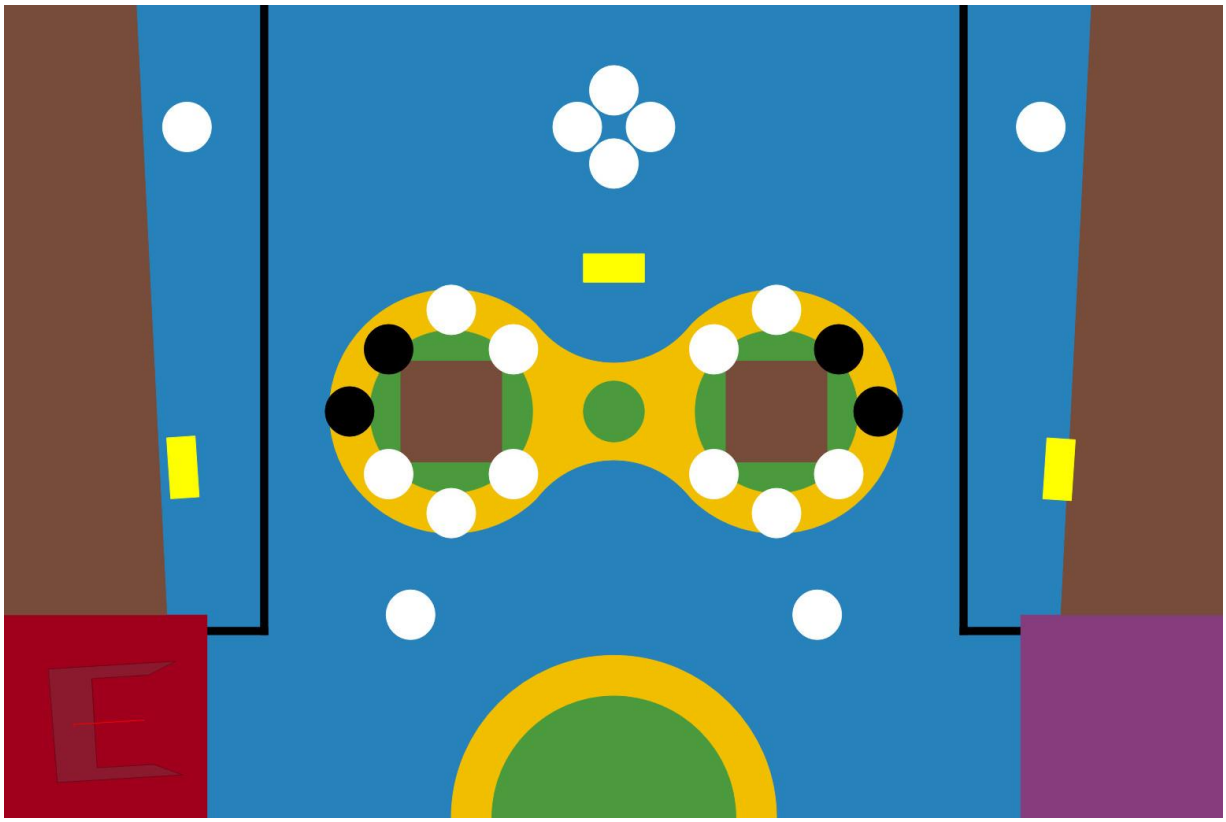


Figure 39: *La table vue depuis le simulateur*

ANNEXE 4 - L'ASSERVISSEMENT

La boucle d'asservissement est un point essentiel dans le fonctionnement du robot. En effet, c'est seulement grâce à lui que le robot est capable de suivre à la lettre les ordres que la stratégie a décidé de lui donner. Sans asservissement, le robot n'irait par exemple pas en ligne droite si on lui demandait car de nombreux paramètre comme les forces de frottement ou l'asymétrie des réponses des moteurs le feraient dériver.

La stratégie base une grande partie de son raisonnement sur la position du robot. Sa maîtrise est donc un facteur essentiel.

L'asservissement que nous avons implémenté est le même que celui des deux années précédente. C'est un asservissement en vitesse contrôlé par la position du robot. Lorsque la stratégie demande à une commande de générer un ordre, pour que le robot se déplace à une position donnée, la commande génère un ordre en vitesse qui est transmis après traitement au filtre PID (Proportionnel Intégral Dérivé). L'ordre en vitesse est décomposé en deux ordres distincts : l'ordre en vitesse linéaire et en vitesse angulaire. Chacune de ces deux consignes ont globalement une forme de trapézoïde.

En effet, la consigne est générée de telle sorte qu'il y ait tout d'abord une phase d'accélération constante, puis une phase où la vitesse du robot reste constante puis finalement une phase de décélération constante. Ce qui est ensuite transmis au filtre PID c'est la valeur de l'erreur entre la vitesse réelle du robot et la valeur de vitesse théorique qu'il devrait avoir (celle générée par la commande). En sortie du filtre PID, on a donc deux valeurs de PWM qu'il faut envoyer aux deux roues, afin que le robot corrige l'erreur entre la consigne et sa vitesse réelle. Vous pouvez voir ci-dessous une représentation de cette correction. En vert, on peut voir la consigne en vitesse passée par la couche commande et en rouge la vitesse réelle du robot, qui tente de suivre la consigne grâce à l'asservissement.

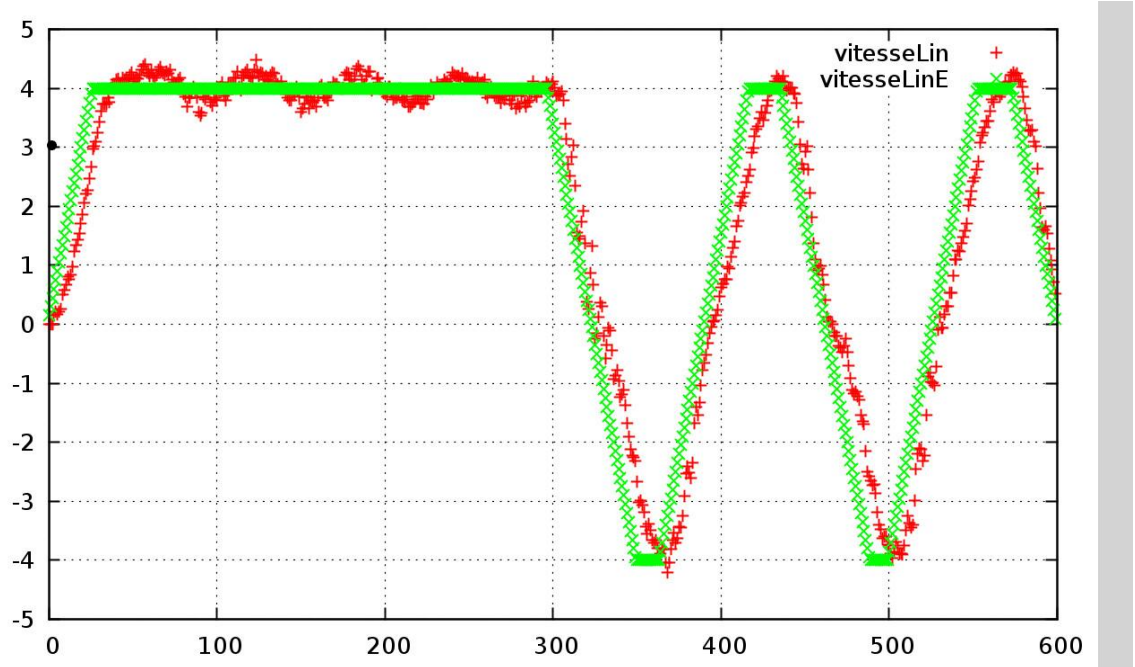


Figure 40: *Les courbes d'asservissement idéale et réelle*

Le filtrage PID qui nous utilisons est une boucle d'asservissement assez classique. Comme évoqué ci-dessus, la sortie (la vitesse réelle du robot) et bouclé sur l'entrée (la consigne en vitesse envoyé par la commande) afin de calculer l'erreur entre ces deux valeurs. Lorsque l'erreur est nulle, le robot se déplace exactement à la vitesse demandée. Voici schématiquement à quoi ressemble un filtre PID (Proportionnel Intégral Dérivé) :

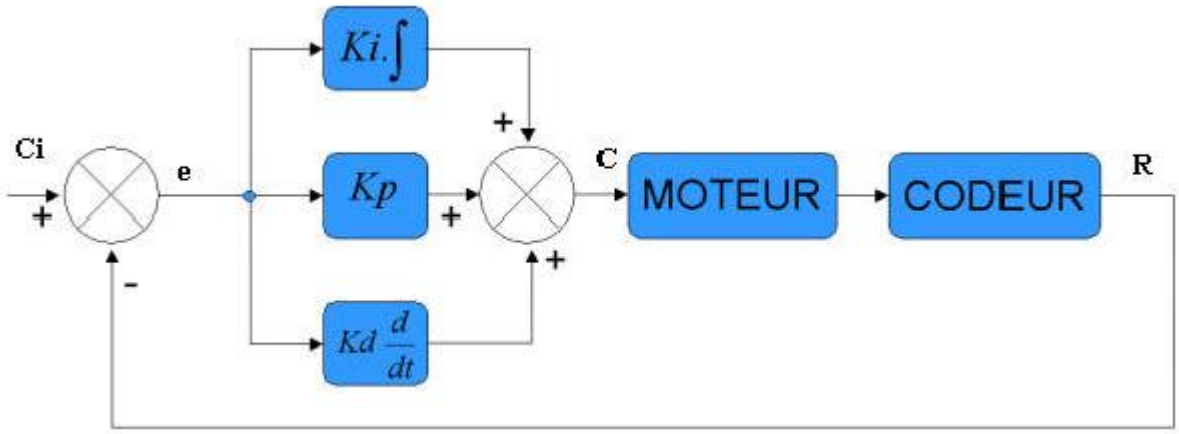


Figure 41: *Fonctionnement du filtre PID*

Légende :
1. Ci : Consigne initiale (ce qu'on veut que le robot fasse) ;
2. e : erreur entre la consigne initiale et la réalité ;
3. C : Consigne appliquée au moteur ;
4. R : Vitesse réelle mesurée (réalité).

On a alors :

$$C = Kp*e + Ki*\int e + Kd*de/dt$$

Pour effectuer le réglage de l'asservissement, il est donc nécessaire de régler trois coefficients (qui correspondent au nom de la régulation) :

- **'Kp' comme proportionnel** : permet d'augmenter la vitesse de montée (pour atteindre la consigne le plus rapidement possible), mais rend le système moins stable ;
- **'Ki' comme intégrale** : réduit l'erreur statique, mais crée beaucoup d'oscillation ;
- **'Kd' comme dérivée** : réduit le dépassement, ralentit la réponse mais atténue les oscillations et rend donc le système plus stable.

Ce réglage se fait de manière empirique. En effet, du fait de l'imprécision de tous les composants du robot, il est très difficile voire impossible d'avoir un modèle théorique suffisamment précis pour trouver les paramètres optimaux du filtre PID. Pour ce faire, on procède en plusieurs étapes :

1) On fixe toutes les valeurs des constantes K à zéro et on fixe l'accélération maximale autorisée pour le robot à une valeur très grande (l'infinie), afin d'obtenir une consigne sous forme d'un créneau. Ensuite, on règle le coefficient Kp, de telle sorte que la consigne soit dépassée d'à peu près 20% >>> tout en restant stable (pas de divergence) ?

2) Ensuite, on règle la valeur du coefficient Kd (et pas Ki comme on pourrait le croire) afin d'obtenir un système stable.

3) Finalement on règle le coefficient Ki afin d'augmenter la vitesse de monté de la réponse du robot, tout en s'assurant de garder un système stable (peu d'oscillation). Finalement, on règle l'accélération maximale autorisée pour le robot. Il faut effectuer ces réglages pour les deux filtres PID : pour la vitesse linéaire et la vitesse angulaire.

ANNEXE 5 - LA GESTION DE LA MEMOIRE DU STM

La gestion de l'allocation de la mémoire est quelque chose de rarement pris en compte dans la programmation "classique" car tout est déjà fait. En effet, lorsque l'on fait un "new" pour instancier un objet en mémoire sur un ordinateur classique, le compilateur du C++ utilise du code déjà implémenté qui fait appel à des fonctions système pour communiquer avec le système d'exploitation lancé sur la machine. Ce dernier indique alors au programme des zones de mémoire qu'il a le droit d'utiliser; le programme gère alors cette zone mémoire automatiquement. Or, sur notre microcontrôleur (cf glossaire), il n'y a pas de système d'exploitation. Le code "standard" n'est donc pas exploitable. Il faut gérer l'allocation mémoire nous-même, et ce n'est pas forcement chose simple.

Les STM (cf glossaire) que nous programmons utilisent la structure mémoire des processeurs ARM Cortex M3. La figure ci-dessous, issue de ***[9]***, présente l'adressage de la mémoire du STM (cf glossaire). C'est un processeur 32 bits, les adresses sont donc codées sur 4 octets. On voit que le code stocké dans la mémoire flash est adressé avec des adresses comprises entre 0x00000000 et 0x1FFFFFFF et les objets instanciés pendant l'exécution du code ont des adresses comprises entre 0x20000000 et 0x2FFFFFFF. Le reste des adresses n'est pas utile pour la compréhension de l'allocation mémoire

.

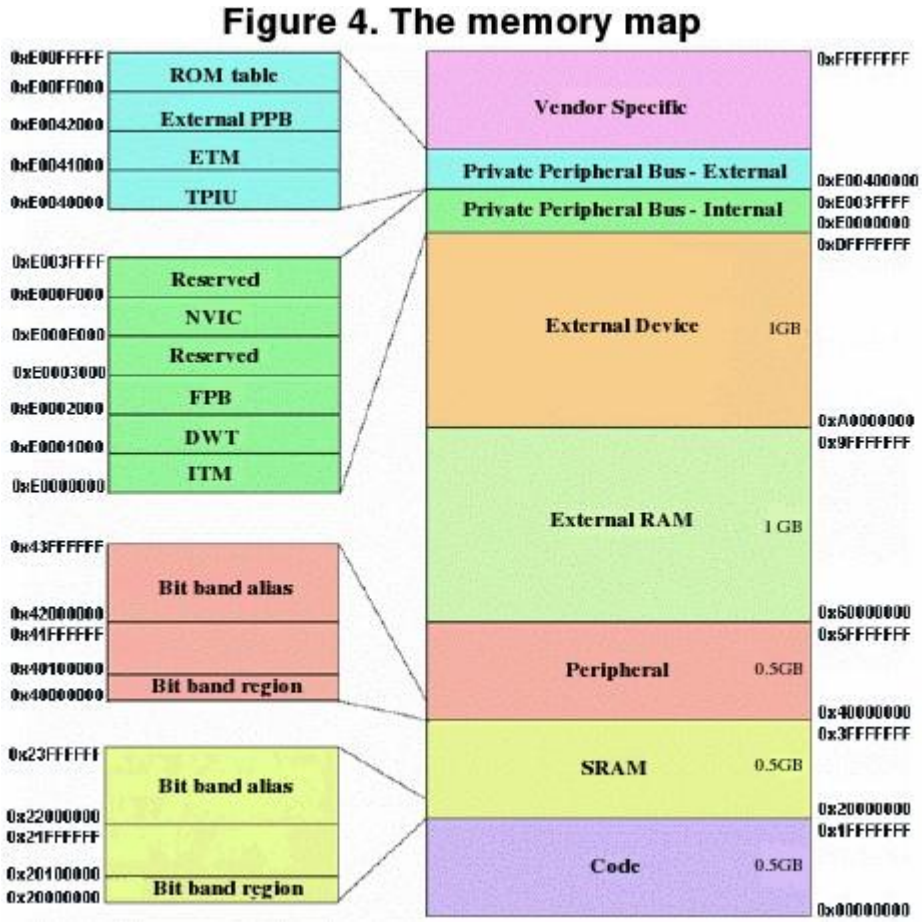


Figure 42: *Les adresses mémoires dans le STM*

Une gestion basique de la mémoire avait déjà été implémentée, mais celle-ci ne permettait pas de libéré la mémoire déjà utilisée. Elle était assez simple. On récupérait au tout début du code

l'adresse de la toute première zone mémoire libre dans la RAM. Si un "new" est effectué dans le code, alors on retourne l'adresse actuelle de la première zone mémoire libre. Le pointeur qui garde en mémoire l'adresse de la première zone mémoire libre est alors incrémenté du nombre d'octets que l'on vient d'allouer pour que ce pointeur pointe toujours vers la première zone mémoire libre pour la prochaine allocation. On peut constater qu'avec cette méthode d'allocation, il n'est pas possible de faire de véritable "delete" qui libère la mémoire utilisée, car, si on déplace le pointeur qui indique la première zone mémoire libre pour qu'il pointe vers la zone mémoire que l'on vient de libérer, alors tout objet alloué après risque de se faire écraser à la prochaine allocation. Un petit schéma pour illustrer tout ça :

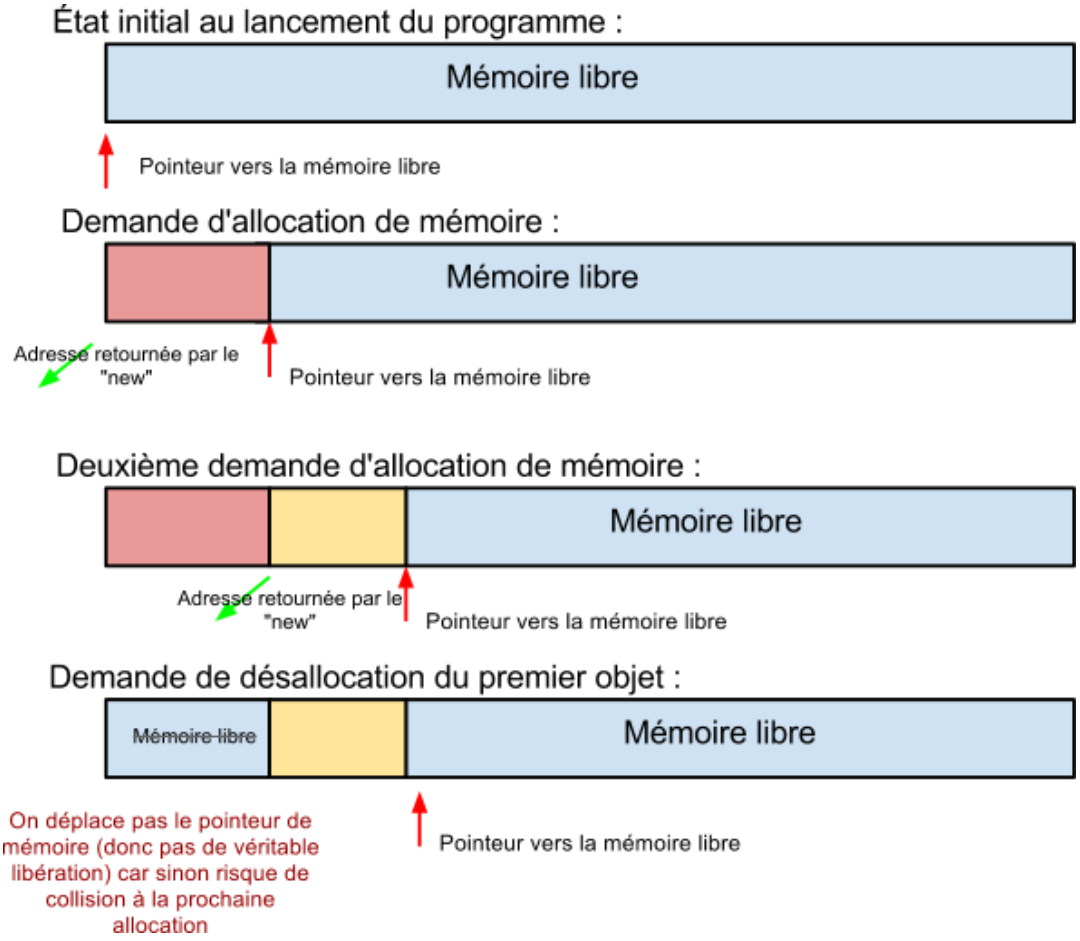


Figure 43: *L'ancienne allocation de mémoire*

Pour ajouter la possibilité de supprimer un objet précédemment alloué dans la mémoire, nous avons décidé de changer totalement la méthode d'allocation de la mémoire. Nous avons décidé de structurer les zones mémoire libre et occupé en ajoutant des headers à chaque début de zones. Ces header contiennent deux informations principales : la taille en multiple de la taille d'un header de la zone mémoire suivante et l'adresse de la prochaine zone mémoire libre.

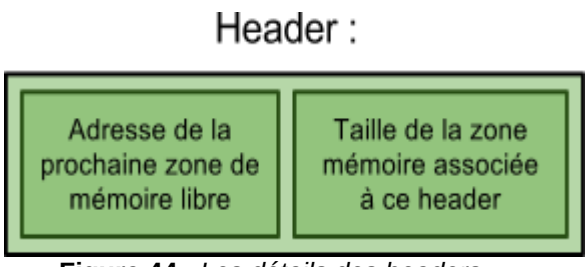


Figure 44: *Les détails des headers*

Lors du lancement du programme, la mémoire libre de la ram est initialisée de telle sorte qu'elle respecte les nouvelles contraintes qu'on lui impose : on crée un header et on initialise un pointer qui pointe vers l'adresse de début de ce header. Ce pointeur pointera toujours vers une zone de mémoire libre. Lorsqu'une demande d'allocation dynamique est envoyée à l'aide d'un new, le programme regarde le header pointé par le pointeur de zone mémoire libre global. Si la taille de la zone mémoire associée à ce header est plus petite que la taille demandée par l'utilisateur, alors on regarde l'adresse de la prochaine zone mémoire libre. Si cette adresse n'est pas égale à l'adresse du header en cours d'étude alors on recommence le teste avec cette nouvelle zone mémoire, sinon c'est qu'il n'y a plus de mémoire libre pour instancier l'objet voulue. Si il y a assez de place, on réserve la place pour cette objet, modifie les informations du header pour qu'elles restent valide, puis on crée si besoin un nouveau header pour la zone mémoire libre restante.

Pour libérer la mémoire, il suffit de mettre dans l'adresse de la prochaine zone libre du header de la zone que l'on souhaite libérer, l'adresse de l'ancien header pointé par le pointeur de zone mémoire libre global et de remplacer la valeur de ce pointeur global par l'adresse du header de la zone mémoire que l'on vient de libérer.

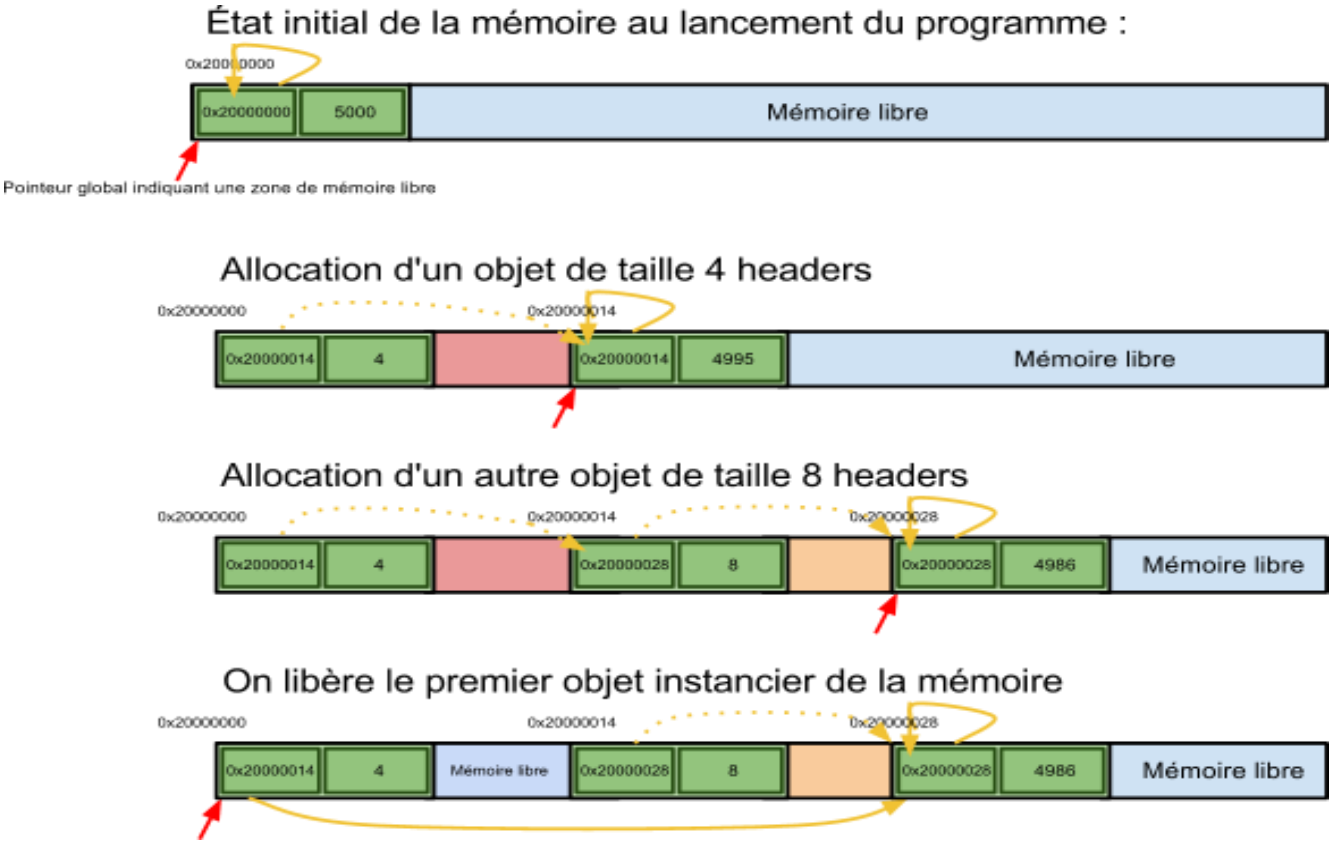


Figure 45: *Exemple de gestion de la mémoire*

Si deux zones mémoire libre se retrouvent sur deux zones consécutives, alors ces zones fusionnent. Ainsi, on limite au maximum la fragmentation de la mémoire.

ANNEXE 6 - CONCEPTION MECANIQUE DU ROBOT PRINCIPAL

A6.1. LA BASE ROULANTE

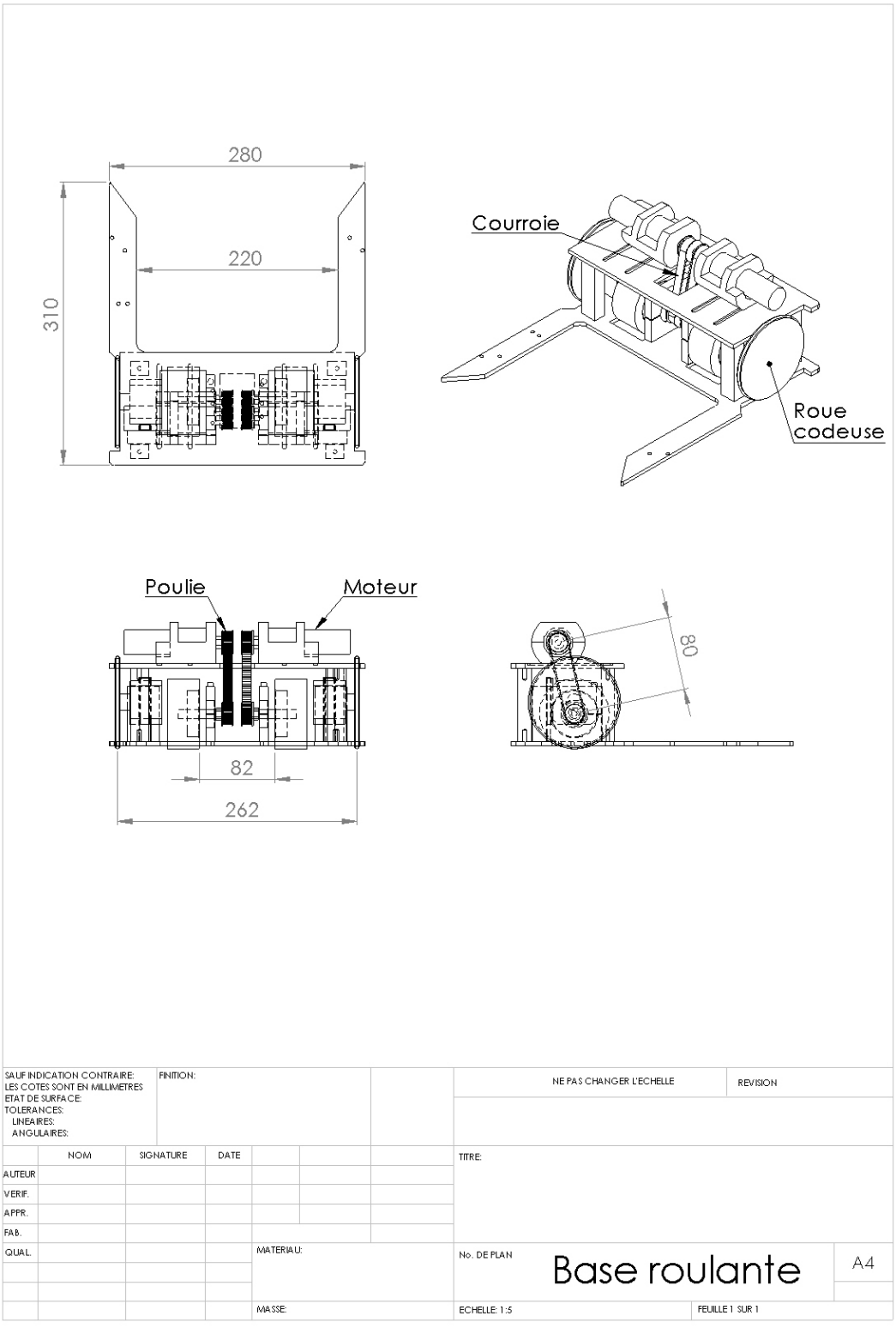


Figure 46: *Mise en plan de la base roulante*

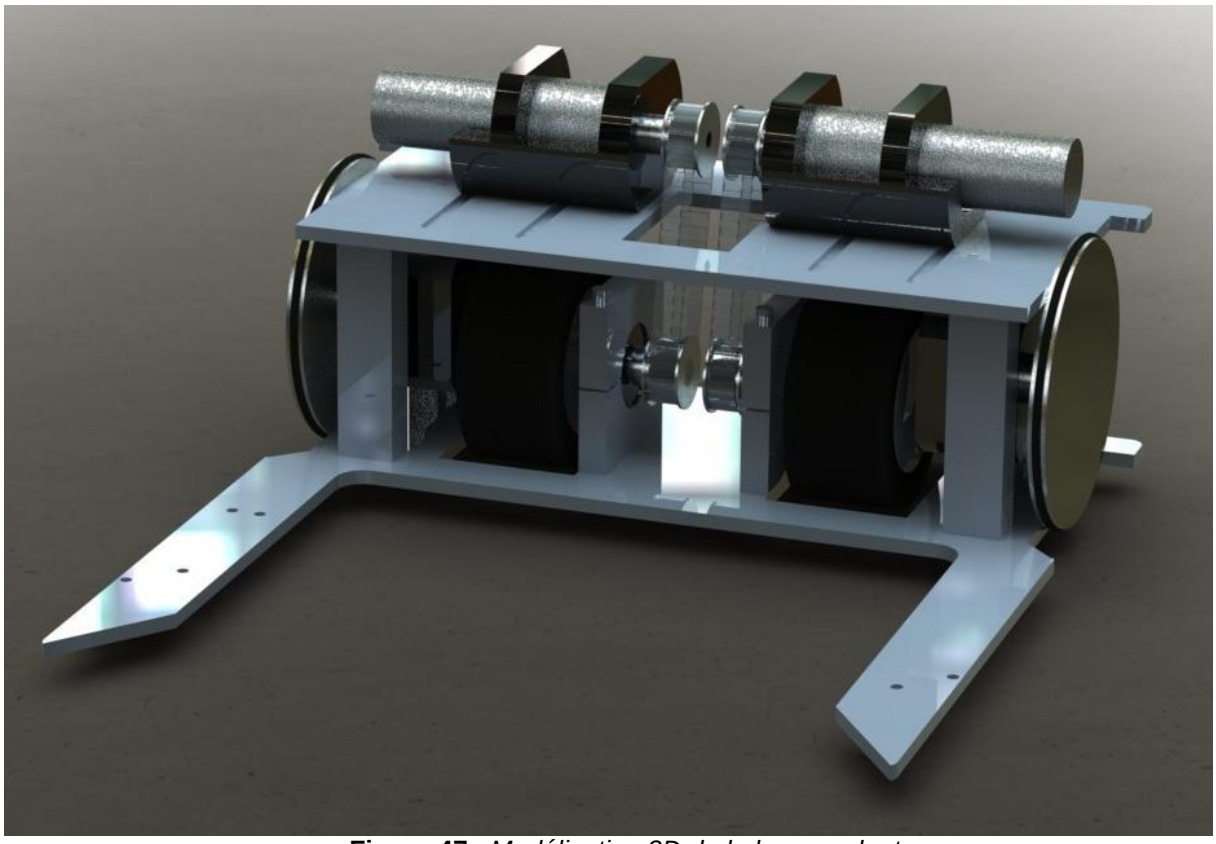


Figure 47: *Modélisation 3D de la base roulante*

A6.2. LES BALAIS LATERAUX

Figure 48: *Modélisation 3D des balais latéraux*

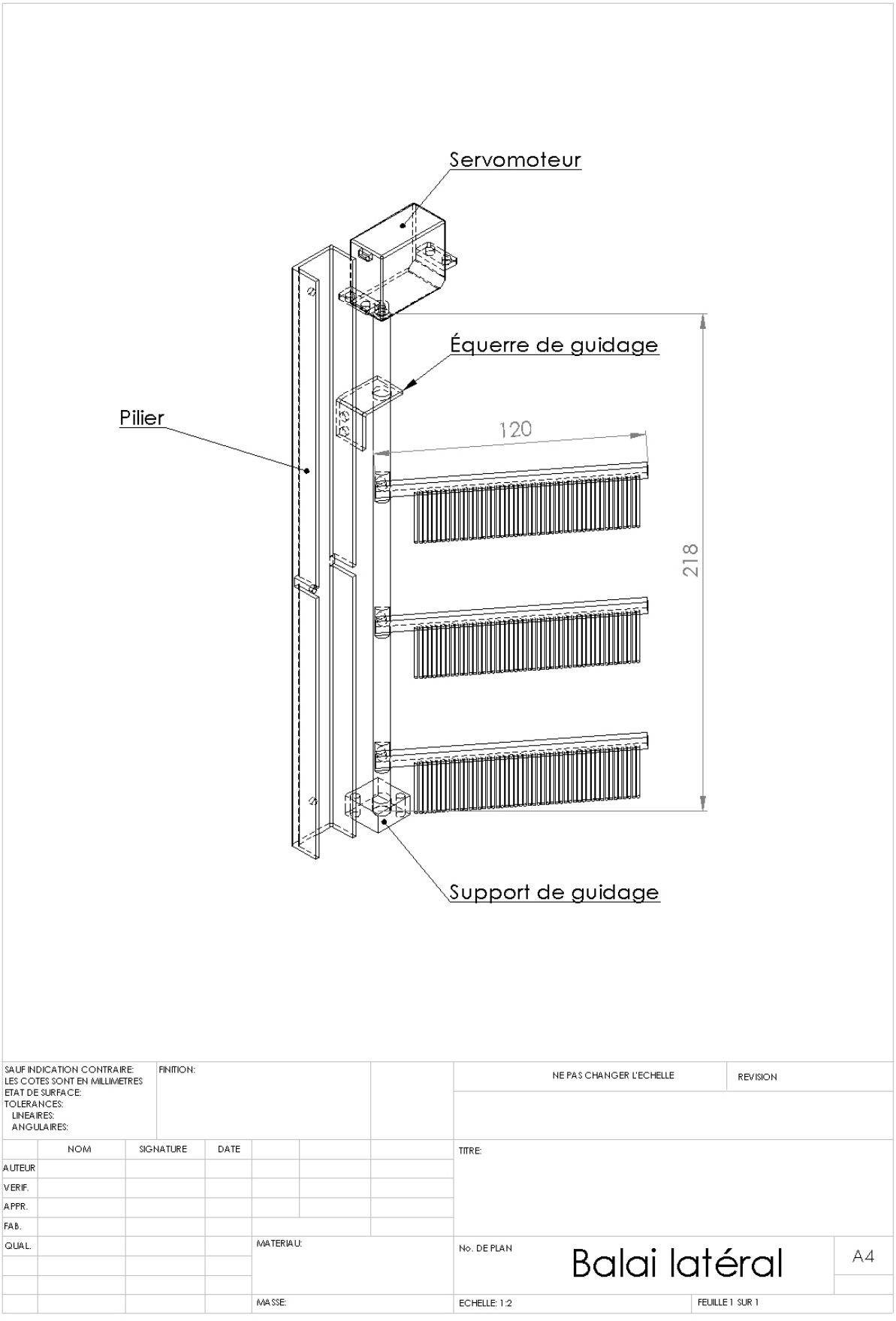


Figure 49: *Mise en plan d'un balai latéral*

A6.3. LE RATEAU FRONTAL

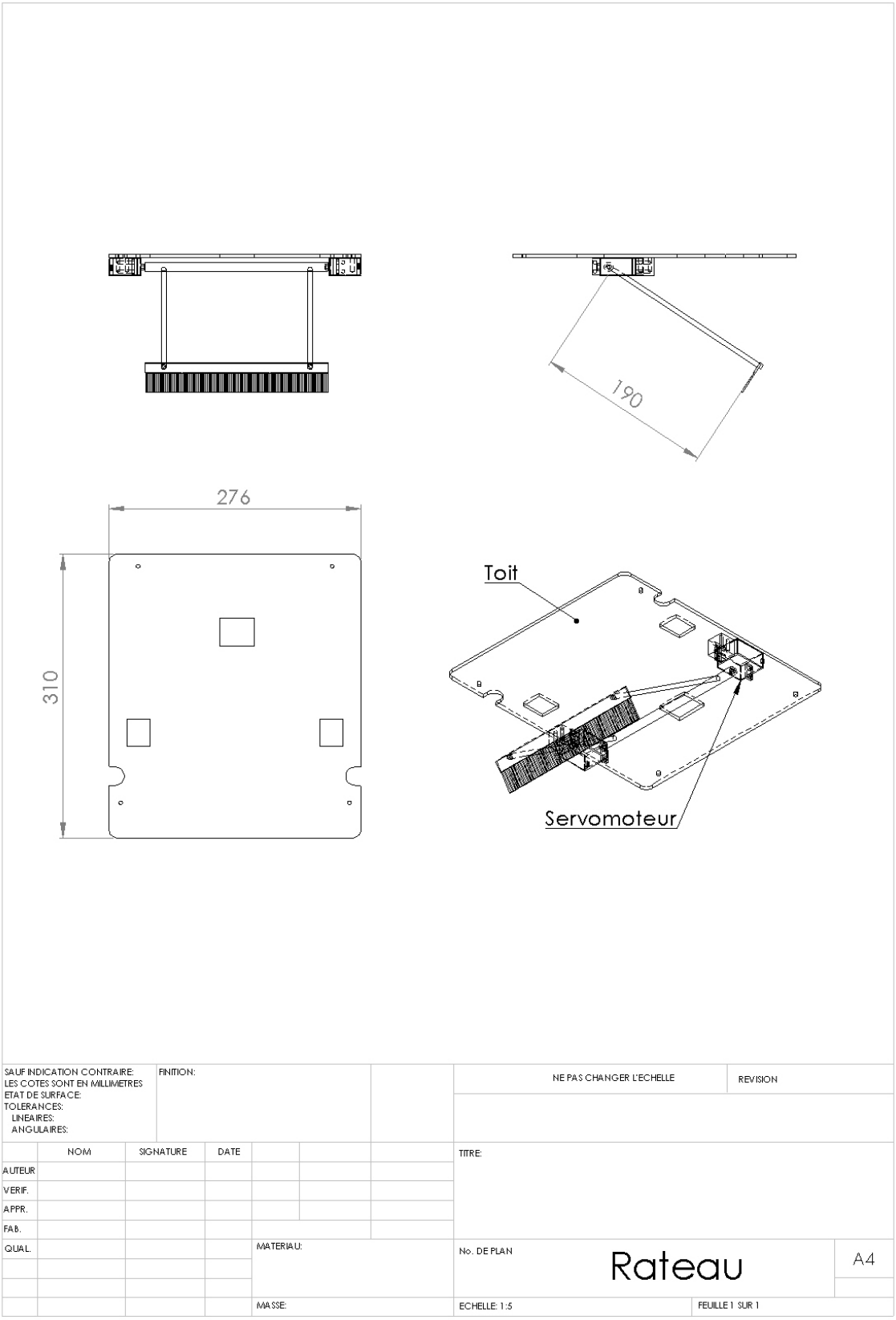


Figure 50: *Mise en plan du râteau frontal*

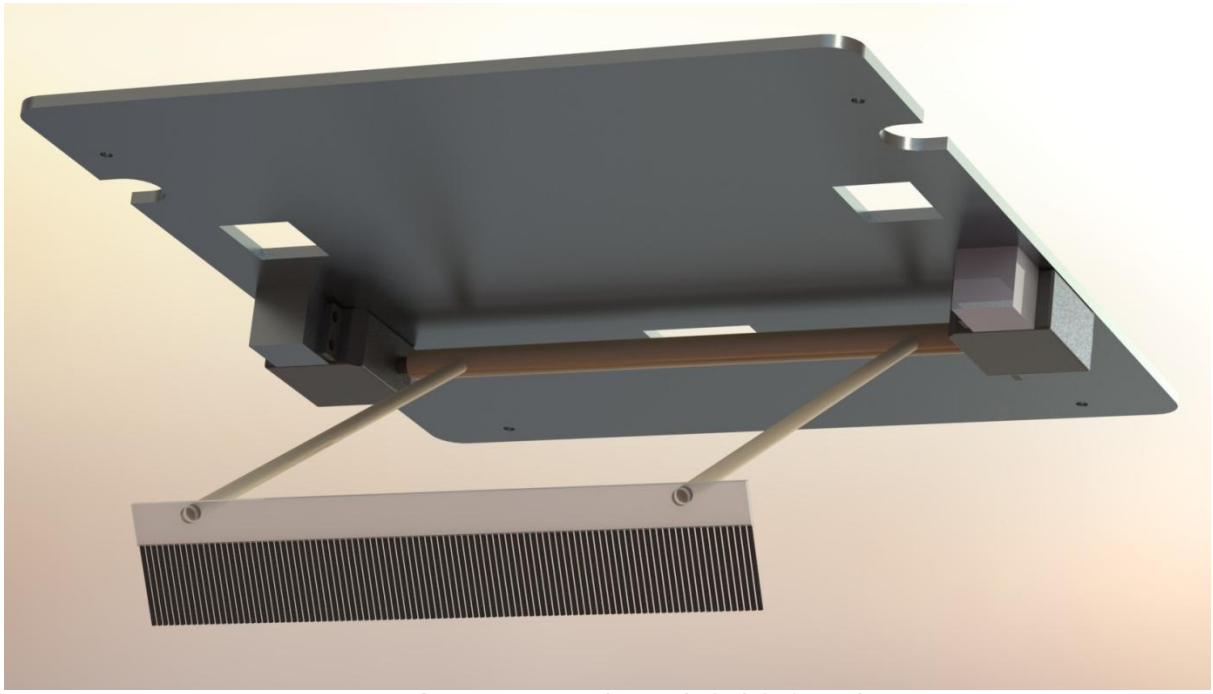


Figure 51: *Rendu 3D du balais frontal*

A6.4. LE ROBOT DANS SA CONFIGURATION FINALE

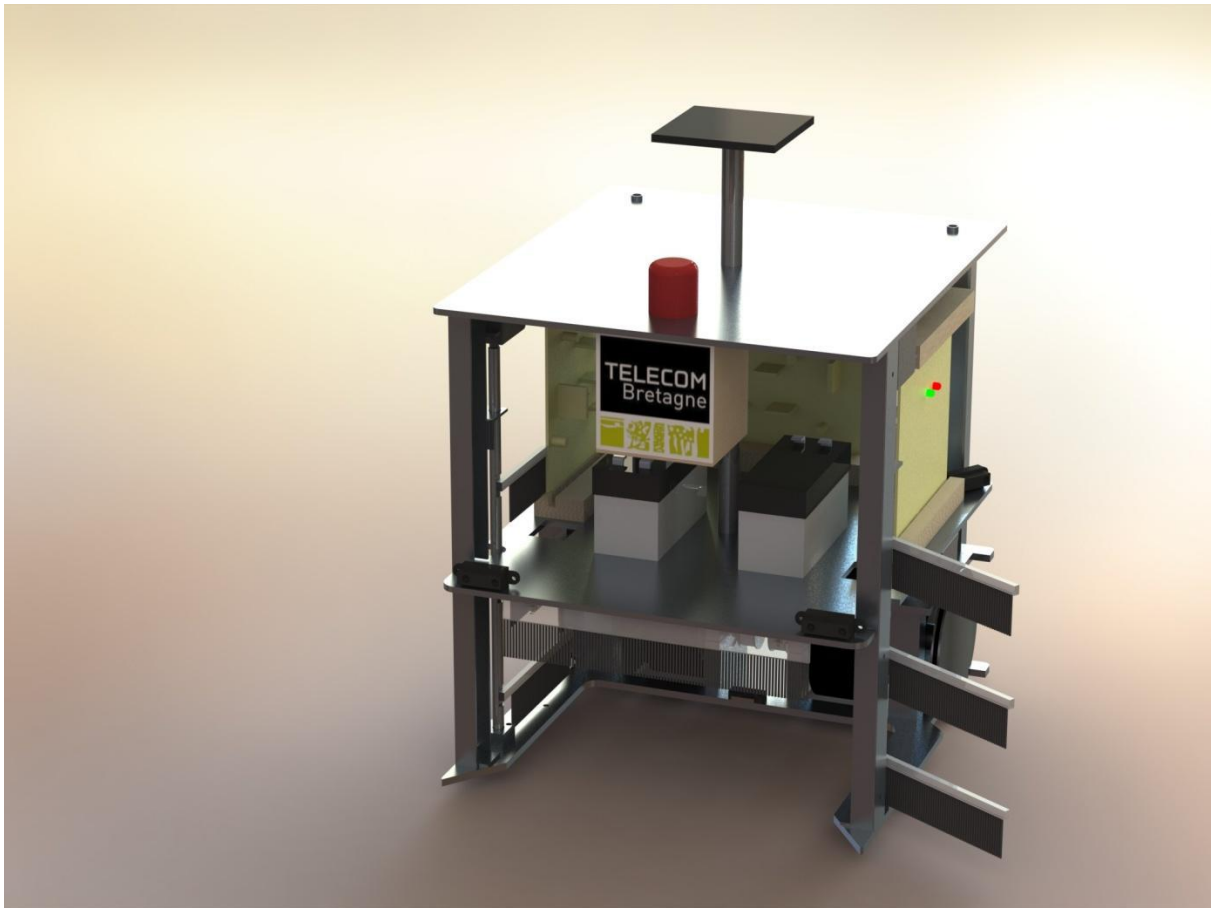


Figure 52: *Modélisation du robot dans sa configuration finale*

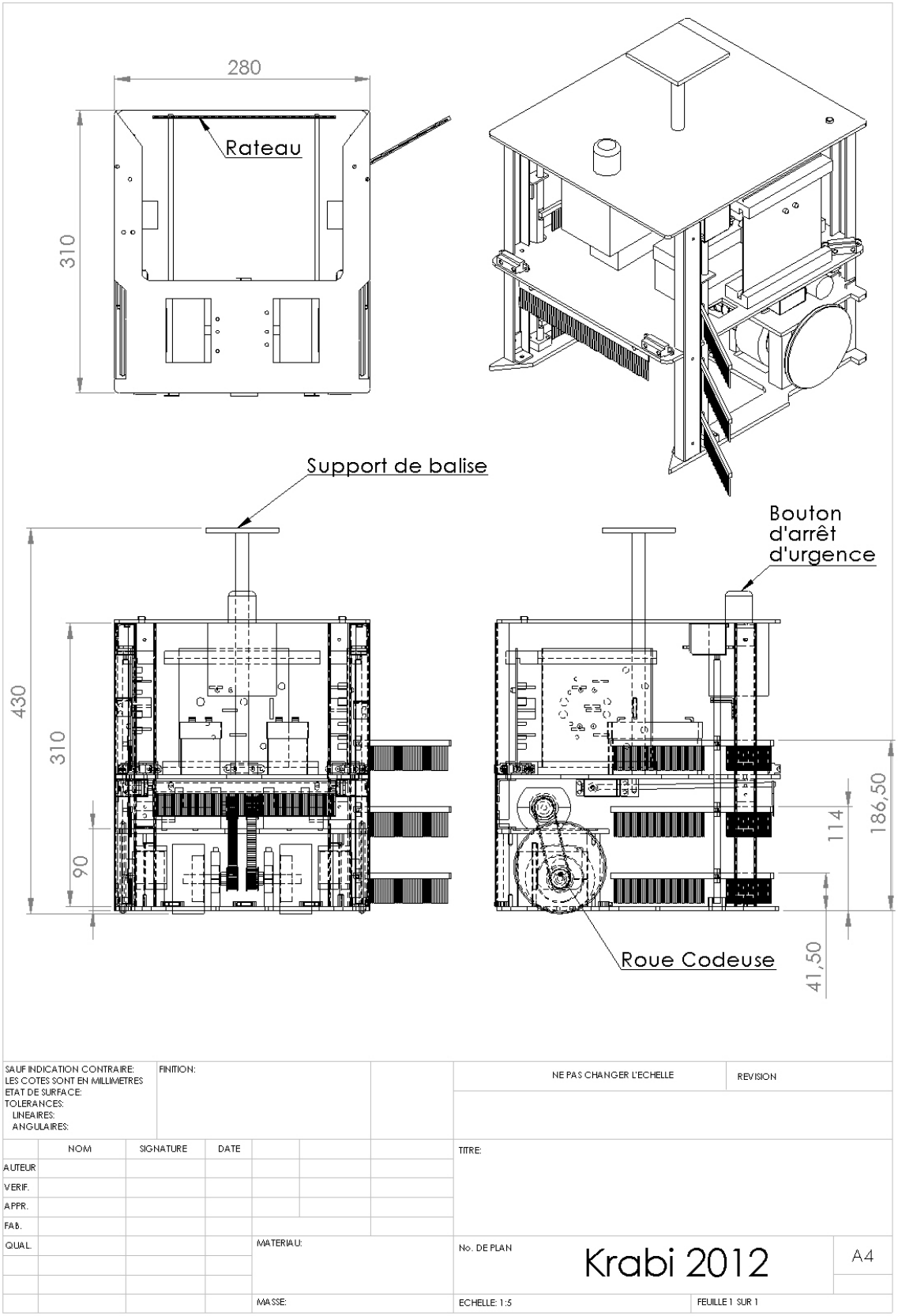


Figure 53: *Mise en plan du robot dans sa configuration finale*

A6.5. MODELISATION DU ROBOT SUR LA TABLE

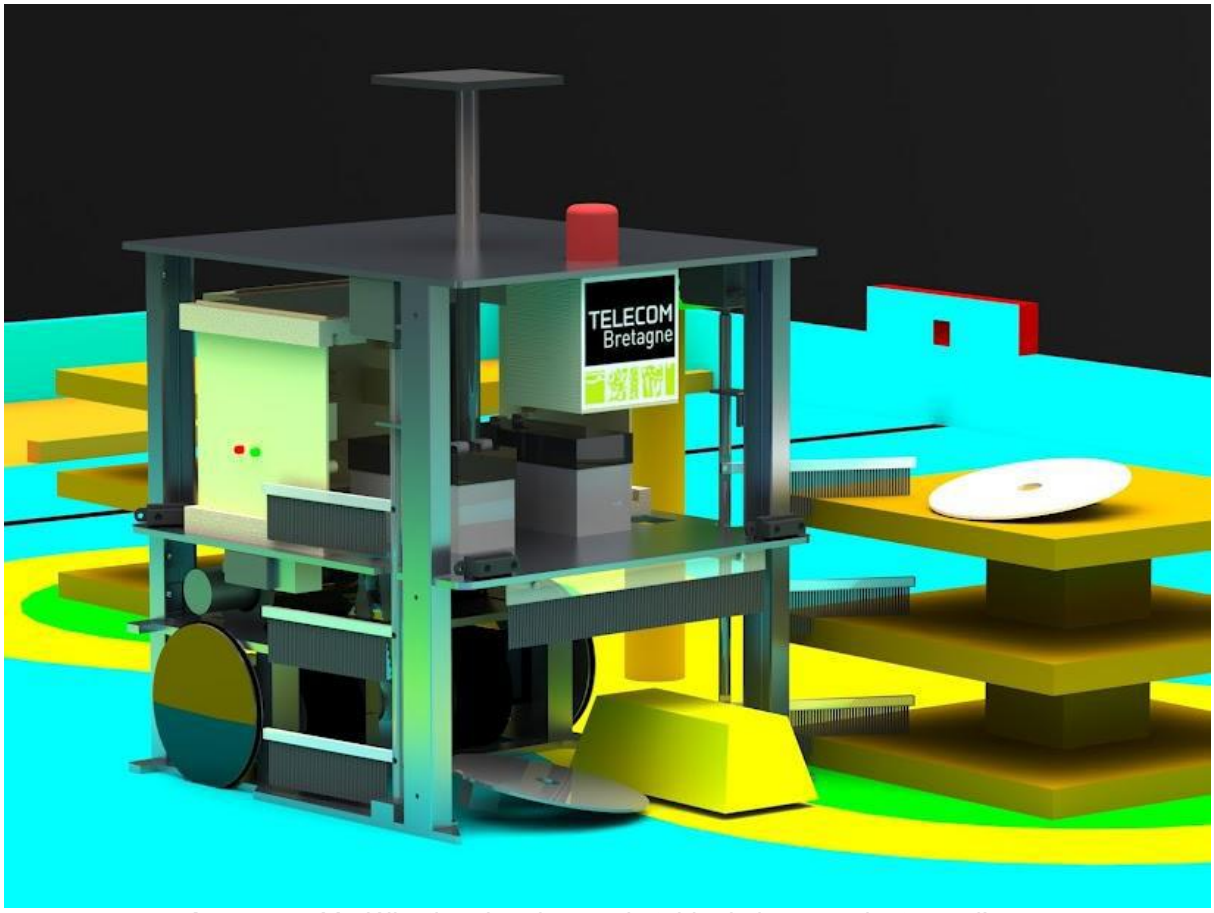


Figure 54: *Modélisation du robot sur la table de jeu aux alentours d'un totem*

ANNEXE 7 - LES CAPTEURS

A7.1. CARTE CAPTEUR ARRIERE

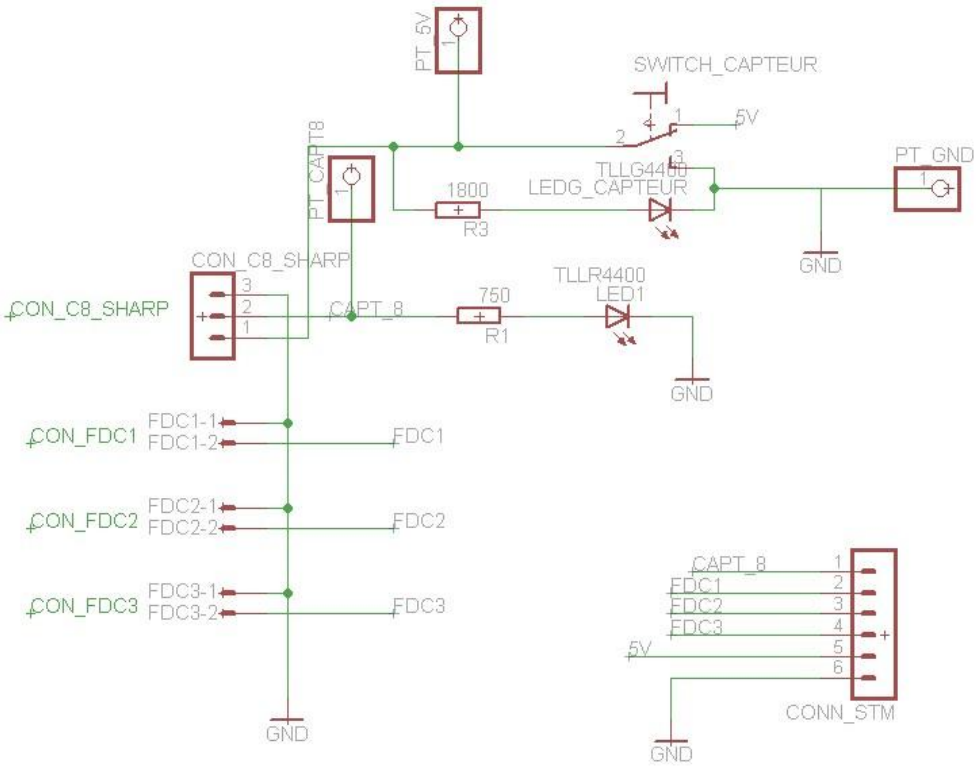


Figure 55: *schéma électrique de la carte capteur arrière*

Entrées :
CON_C8_SHARP : signaux reçus par le capteur arrière
CON_FDC1 / CON_FDC2 / CON_FDC3 : signaux reçus par les trois fins de course
5V : Tension d'alimentation de 5V provenant de la carte STM.

Sorties :
CAPT_8 / FDC1 / FDC2 / FDC3 : envoie tous les signaux reçus par les capteurs à la carte STM

A7.2. CARTE CAPTEUR AVANT

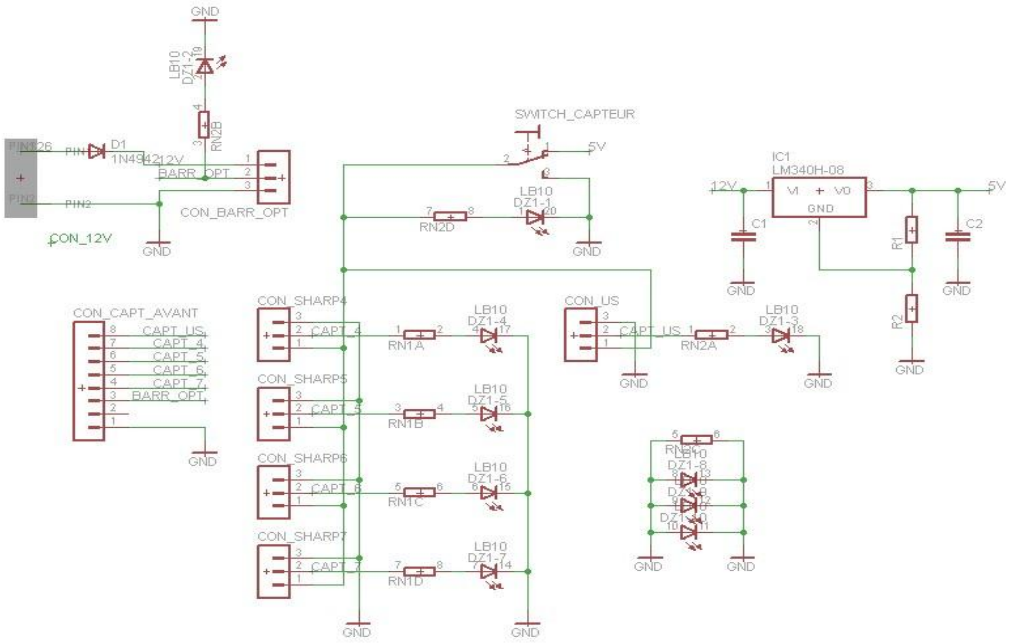


Figure 56: *schéma électrique de la carte capteur avant*

Entrées :
12V : tension continue provenant des batteries.
5V : Tension d'alimentation de 5V provenant de la carte STM.
CON_SHARP4 / 5 / 6 / 7 : signaux reçus par les trois autres capteurs

Sortie :
CON_CAPT_AVANT : envoie tous les signaux reçus par les capteurs à la carte STM

A7.3. GESTIONS DES CAPTEURS

A7.3.1. CAPTEURS SHARP

Les capteurs SHARP permettent de mesurer des distances par détection infrarouge. Ils peuvent détecter une distance à partir de 4 cm et jusqu'à 30 cm. Ils sont alimentés en 5V par les cartes capteurs. La tension de sortie revient dans la carte capteurs. Ensuite la carte capteurs envoie les tensions dans la carte STM pour les traiter.
Pour avoir une bonne stratégie d'évitement, nous avons installé 5 capteurs SHARP autour du robot : 2 à l'avant, 2 au milieu et 1 en arrière. Les capteurs utilisent l'infrarouge pour mesurer la distance entre le robot et l'obstacle. En guise d'exemple, voici un capteur SHARP :

Figure 57: *capteur SHARP*

A7.3.2. ULTRASON

L'ultrason est autre type de capteur qui envoie le son de fréquence supérieure à 20 000Hz. Nous avons utilisé le capteur ultrason "MS-EZ3". Celui-ci est capable de détecter la présence d'un objet de 0 à 6,45 m. Le module est également capable de déterminer la distance qui le sépare d'un objet se présentant devant lui (entre 15,24 cm et 6,45 m). L'angle de détection et la portée maximum de détection du module diffèrent en fonction de la taille, de la forme et de la texture de l'objet à détecter. Au niveau de la stratégie, il permet de choisir le bon chemin et d'éviter de se diriger vers les CD déjà récupérés par l'ennemi. Voici le capteur Ultrason :

Figure 58: *capteur ultrason*

.

A7.3.3. FIN DE COURSE

Les capteurs de fin de course sont des capteurs de contact. Lorsqu'il y a contact physique, le fin de course agit comme un circuit fermé (et ouvert dans le cas contraire), et le STM (cf glossaire) reçoit un niveau logique 0 (et haute impédance s'il y a pas de contacts).

On utilise les fins de course sur la face arrière du robot pour le calibrer si celui-ci a perdu sa position sur la table de jeu. En ce « calant » sur la table, les capteurs fin de courses s'activent : on peut donc savoir que le robot est positionné au bord de la table, et recalculer sa position suivant un axe. Il suffit de répéter la même manipulation pour avoir sa position sur le 2e axe. Nous avons également un 3ème fin de course dans la cale du robot pour détecter lorsque le robot a récupéré des éléments de jeu.

ANNEXE 8 - CARTES ELECTRONIQUES

A8.1. LA CARTE ALIMENTATION

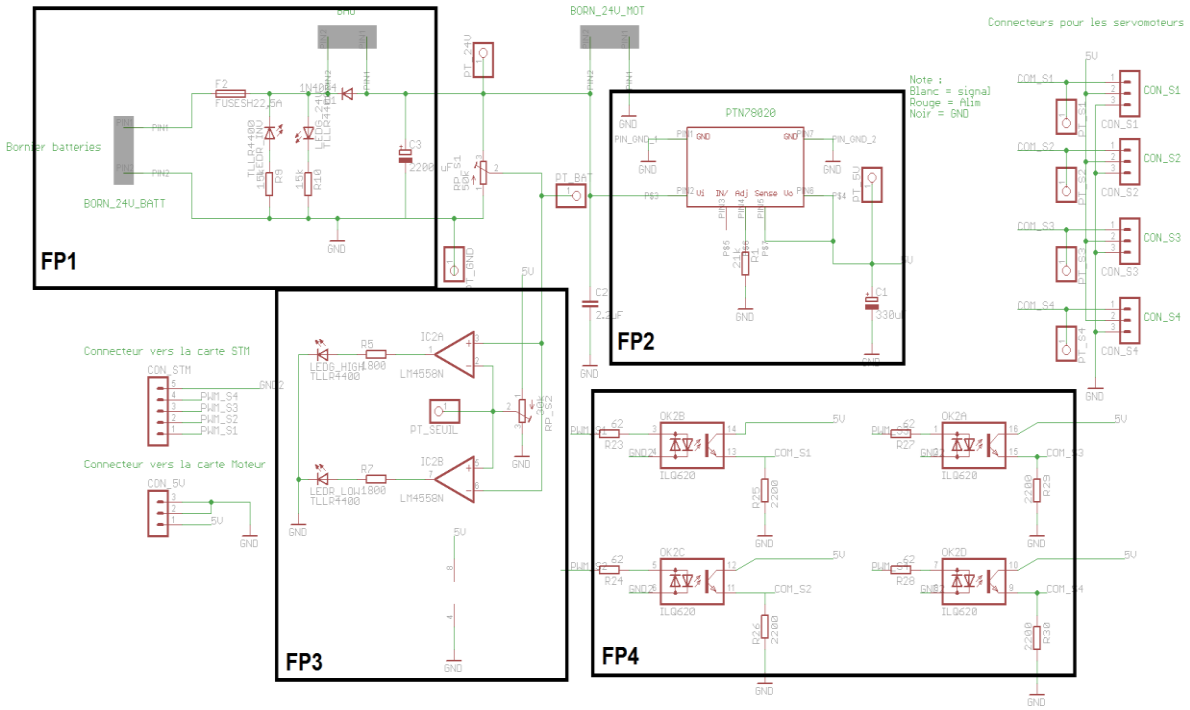


Figure 59: *schéma électrique de la carte alimentation*

Entrées :
12V : tension continue provenant des batteries.
GND : la masse de référence fixée à 0V, présente sur la plupart des connecteurs pour avoir une masse unique. Cette masse est cependant différente de la masse de la carte STM.
PWM_S1 à PWM_S4: signaux PWM provenant de la carte STM. Il s'agit de signaux créneaux d'amplitude 5V, de fréquence 22 KHz, et de rapport cyclique variable.

Sorties :
COM_S1 à COM S4 : Les signaux PWM des servomoteurs (cf glossaire), d'amplitude 5V et de fréquence 22 Khz. Le rapport cyclique de ce signal à une influence sur l'angle du servomoteur
12V_MOT : tension contenue délivré à la carte moteur
5V : tension contenue délivré à la carte moteur pour alimenter les composants logiques.

A8.1.1. FP1 : PROTEGER L'ALIMENTATION

Principe de fonctionnement :

L'alimentation 12V provenant de la batterie doit se faire en respectant les polarités. Si le branchement est correct, le LED verte s'allume, sinon une LED rouge indique que les polarités sont inversées. Pour le moment, seul le bouton d'arrêt d'urgence protège la carte contre une inversion des polarités, il faut donc veuillez à l'enclencher avant de brancher les batteries pour vérifier que la bonne LED s'allume. Le fusible peut toujours protéger la carte en cas d'oubli, comme ce fût le cas pour notre année, mais il y a toujours un risque d'abîmer les composants.

Il est très important de ne pas rajouter une diode de protection en série pour régler ce problème ! Nous avons besoin d'un retour de courant vers la batterie, or, l'ajout de ce composant l'en empêche. La raison est expliquée dans le détail de la carte moteur.

La solution que nous proposons aux années futurs, mais que nous n'avons pas eu le temps de tester, est de rajouter une diode de protection en **parallèle**, avec le fusible **avant** le bouton d'arrêt d'urgence (BAU), comme indiqué dans la figure ci-dessous :

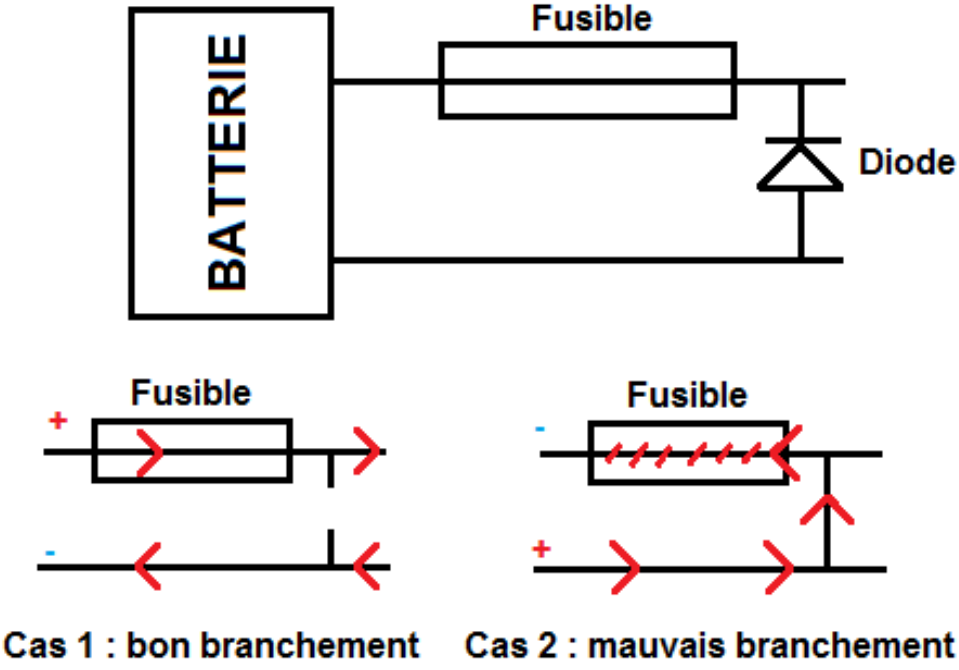


Figure 60: *solution proposée pour protéger d'une inversion*

Dans le 1er cas, la diode se trouve bloqué, et le courant circule dans le bon sens. S'il y a un retour de courant du moteur vers la batterie, on retrouve cette même situation.
Dans le 2e cas, la diode se trouve passante et soumis à une tension très élevée, donc à un fort courant qui la traverse, ce qui va faire fondre le fusible et protéger la carte.

Le fonctionnement du BAU est simple : lorsqu'il est déclenché, il fait office d'un circuit fermé. Dans le cas contraire, c'est un circuit ouvert. Ainsi, il coupe directement l'alimentation. La diode câblée en parallèle est **indispensable**, car elle permet d'autoriser le retour du courant des moteurs.

Enfin, le condensateur C3 après le BAU permet de lisser la tension d'alimentation en cas de fluctuation de la tension des batteries, lors d'un appel de courant important. Il permet également d'absorber une partie du courant de démarrage du moteur qui peut être très élevé, jusqu'à 10A.

Dimensionnement des composants :

Le fusible utilisé supporte un courant inférieur à 10A. Le courant maximum consommé par les cartes alimentations et moteurs étant estimé à 6A, nous avons laissé une marge pour éviter que le fusible ne fonde en fonctionnement normal. Lors d'un court-circuit, la batterie délivre plus de 10A, ce que nous avons pu malencontreusement confirmer lors d'un mauvais branchement de batterie. Ainsi, en cas court-circuit, le fusible s'avère très efficace et rajoute de la sécurité dans nos cartes. A noter que le courant de démarrage des moteurs ne détruit pas les fusibles car c'est un phénomène bref (quelques millisecondes).

A8.1.2. FP2 : REGULER LA TENSION

Principe de fonctionnement :

Cette fonction est réalisée par le régulateur (cf glossaire) PTN 78020W qui « transforme » la tension 12V de la batterie en 5V qui va alimenter les autres parties de la carte. Ce régulateur est protégé contre les court-circuits et peut délivrer un courant allant jusqu'à 6A. Tous les composants passifs autour de lui sont préconisés par le document constructeur afin de garantir un fonctionnement optimal. La résistance R1 permet d'obtenir le 5V en sortie, car ce régulateur a pour particularité d'être ajustable.

Dimensionnement

Il existe 2 grandes familles de régulateurs : les régulateurs linéaires et les régulateurs à découpage. Les premiers ont un avantage conséquent par rapport aux premiers : ils ont un rendement beaucoup plus intéressant (90% contre 50-40%), et chauffe donc beaucoup moins. Ils peuvent délivrer des courants souvent plus importants. Nos servomoteurs (cf glossaire) peuvent demander un courant allant jusqu'à 1A. Ainsi, avec 3 servomoteurs, il faut prévoir 3A. Nous avons choisi le régulateur PTN 78020W qui peut délivrer 6A, afin d'avoir de la marge.

Le document constructeur ***[10]*** préconise une résistance de R1 = 21 kΩ, de tolérance 1%, 0.05 W avec une stabilité en température de 100 ppm/°C (ou plus), pour avoir Vo = 4.996V. Dans le layout, il faut placer la résistance le plus proche possible du régulateur, entre les PINs 4 et 7. Sinon, il risque d'y avoir un manque de précision sur la valeur réelle de Vo.

Il faut placer un condensateur céramique de 2.2 uF en entrée, à 0.5 inch (1.27 cm) de la PIN d'entrée (n°2). Il faut aussi que leurs températures suivent une caractéristique de type X5R ou X7R. On peut toujours mettre un condensateur électrochimique en parallèle (pour nos pont en H), mais ce condensateur est obligatoire.

En sortie, il faut un condensateur d'au minimum 330 uF, avec un 250 mA rms. La technologie n'est pas imposée : on peut prendre du céramique ou de l'électrochimique. Il est conseillé de le place aussi à 0.5 inch de la sortie (PIN 6).

A8.1.3. FP3 : AFFICHER LA CHARGE DE LA BATTERIE

Principe de fonctionnement :

Cette fonction permet de détecter si la batterie est chargée ou déchargée. Nous utilisons des amplificateurs opérationnels (AOP) en mode comparateur à un seuil. Un premier potentiomètre RP_S1 va ramener la tension d'alimentation à une valeur pouvant être traitée par l'AOP, donc inférieur à sa tension d'alimentation de 5V. Nous tenons à insister sur ce point-là, car c'est une erreur qui peut facilement être commise pour une personne non avertie : **l'AOP n'arrive pas à traiter une tension en entrée supérieur à sa tension d'alimentation.** Pour être sûr de ne pas avoir de problème et d'avoir de la marge, le meilleur choix est de choisir une tension inférieure à sa tension de saturation.

Le deuxième potentiomètre (RP_S2) va permettre de régler le seuil de détection. Les AOPs vont ensuite comparer ce seuil à la tension obtenue avec le potentiomètre RP_S1, comme le montre le schéma ci-dessous :

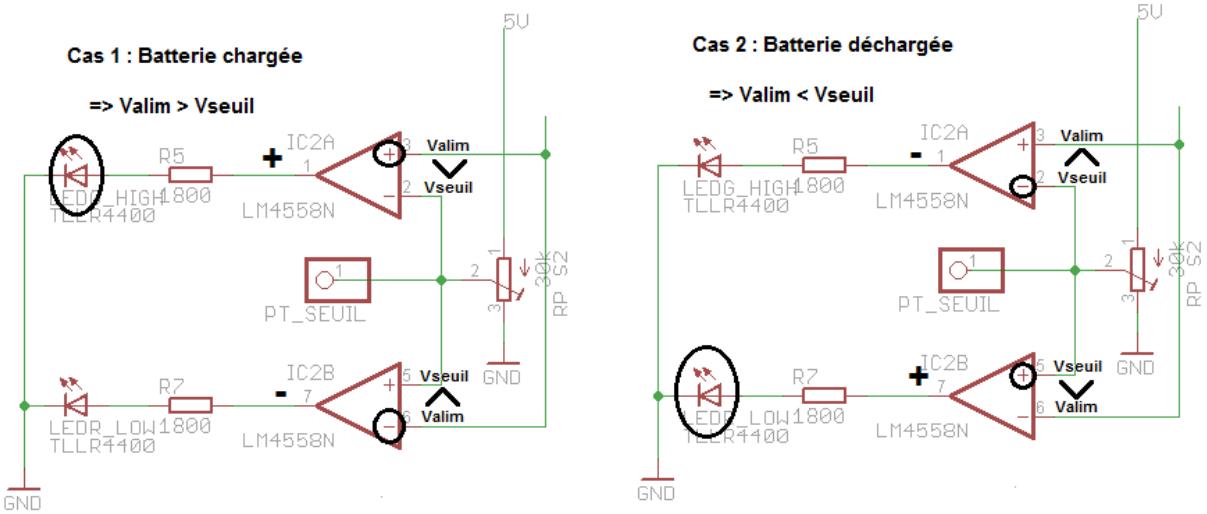


Figure 61: *schéma de principe de la détection de charge*

Si la Valim > Vseuil, LEDG_HIGH s'allume (batterie chargée). Dans le cas contraire, c'est LEDR_LOW qui est allumé.

Dimensionnement

Il n'y a qu'une seule réelle contrainte sur l'AOP a utilisé : il faut que celui-ci est une alimentation asymétrique, car nous n'avons pas moyen simple pour fournir une tension négative. Il faut aussi pouvoir être alimenté en 5V. C'est selon ces critères que nous avons choisis les LM324 (biblio : http://www.datasheetcatalog.org/datasheets/208/62529_DS.pdf). Une remarque importante concernant ce composant : sa tension de saturation n'est pas exactement égale à sa tension d'alimentation, mais diffère de 1.5V. Ainsi, lorsqu'ils sont alimentés en 5V, leur tension de saturation vaut 3,5V. Il faut prendre cela en compte lors du calcul des résistances pour les LEDs.

Pour le réglage du potentiomètre RP_S1, la tension obtenue sur le point test PT_BAT doit être de 3.5V lorsque la batterie est chargée, à 12,6V. La batterie est considérée comme déchargée lorsqu'elle atteint la valeur de 12,2V. Proportionnellement, cela représente un seuil de 3,42V à régler grâce au potentiomètre RP_S2. Les valeurs des résistances totales des potentiomètres ont été choisies de façon à rendre négligeable le courant qui circule (12/50000 = 0.24 mA)

A8.1.4. FP4 : ISOLER LES SIGNAUX

Principe de fonctionnement :

Cette fonction a pour rôle d'isoler électriquement les signaux provenant de la carte STM. Pour cela, nous utilisons un composant nommé optocoupleur (cf glossaire). Ils sont composés d'une diode et d'un phototransistor : lorsque la diode émet, le transistor se trouve passant. Le schéma ci-dessous illustre ce principe. La transmission des informations est donc réalisée de manière optique, et non électronique. Il faudrait une tension supérieure à 1000V pour rompre cette isolation, ce qui n'est pas possible dans notre système.

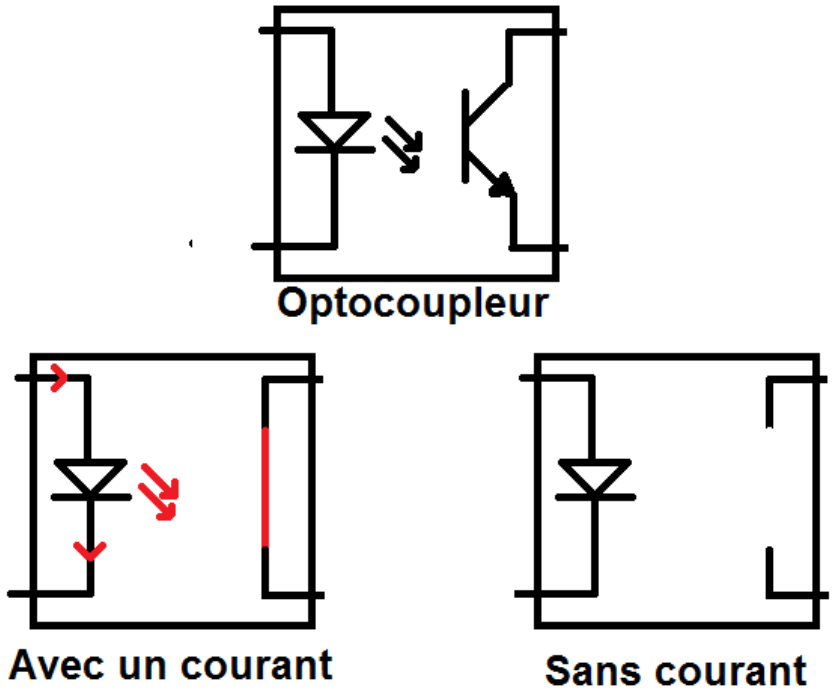


Figure 62: *L'optocoupleur*

Le PWM envoyé par la carte STM permet de contrôler l'angle des servomoteurs (cf glossaire). Il y a proportionnalité entre l'angle de rotation du servomoteur et le rapport cyclique du signal PWM.

Dimensionnement

Les optocoupleurs (cf glossaire) utilisés sont ceux de la série ILQ620 ***[11]***. Ils ont l'avantage de contenir 4 optocoupleurs par boîtier de composants, et ont des prix très raisonnables (environ 4 euros chez Farnell). En revanche, leur fréquence d'utilisation est limitée à environ 40kHz, mais ceci n'est pas un problème car nos PWM ont une fréquence inférieure à celle-ci. De plus, il faut un courant de 10 mA pour polarisé la diode, il faut donc faire attention à ce que le microcontrôleur (cf glossaire)

puisse bien délivrer ce courant. Ce n'est pas le cas du STM H107, il nous a donc fallu prévoir un composant supplémentaire (voir la partie sur la carte STM).

Calcul des résistances :

La résistance en entrée calcul de la même façon que pour polariser une LED classique :
R_in = V – Vf / If
Avec:
V = 5V
Vf = 1.15V
If = 10 mA
Donc R_in = 390 Ω, ¼ W.

Pour calculer la résistance en sortie R_out :
Il faut que le transistor soit saturé, donc icsat < CTRmin * If = 100% * 10^-2 = 10 mA. Nous avons pris ic = 2 mA pour avoir de la marge. Il ne reste plus qu'à calculer la résistance R_out :
R_out = (5 – Vcesat – Vf) / icsat
Avec Vcesat = 0.4V et Vf = 1.7V
Donc R_out = 2200 Ω ¼ W

A8.2. LA CARTE STM

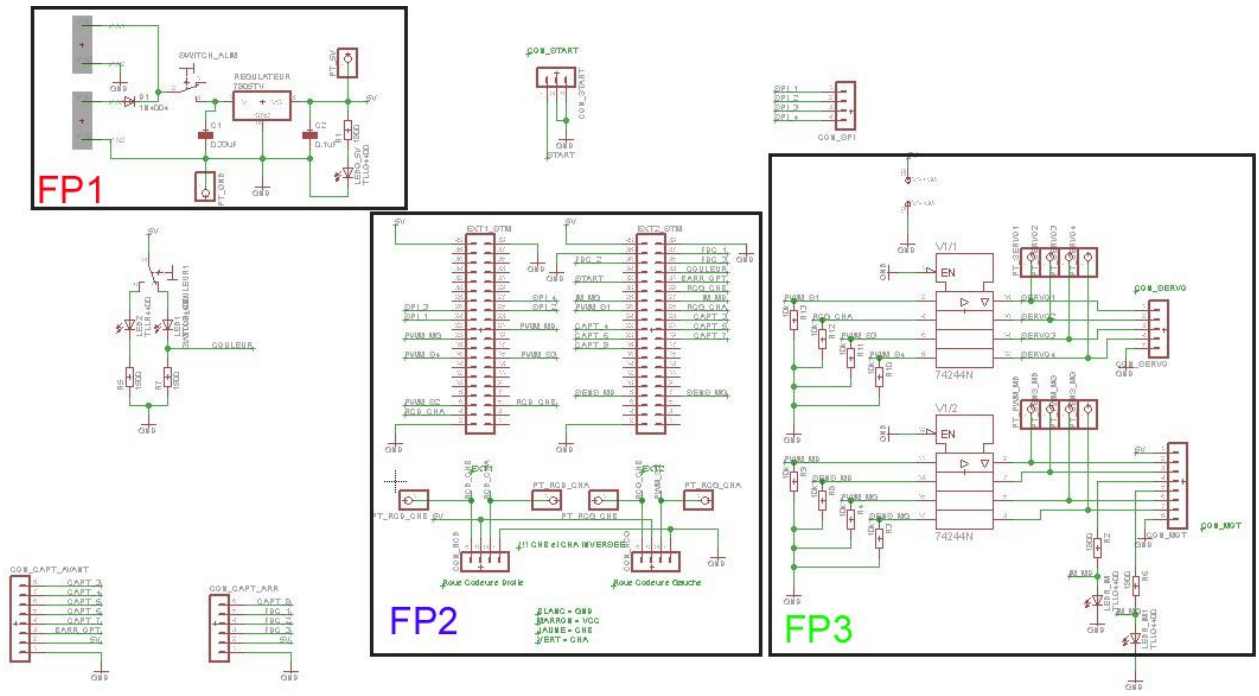


Figure 63: *schéma électrique de la carte STM*

Entrées :
12V : tension continue provenant des batteries.
Start : signal numérique contenant les informations de présence de la tirette.
CAPT_1 à CAPT_5 : signaux analogiques envoyé par les capteurs infrarouges. Leurs tensions est comprises entre 0 et 3V.
FdC1 à FdC3 : informations envoyés par les capteurs de fin de course. Si ceux-ci sont en contact, ils envoient un 0 logique. Dans le cas contraire, ils n'envoient aucune information, car il y a un circuit ouvert (état d'haute impédance HZ).

Sorties :

SERVO1 à SERVO4 : signaux PWM envoyés aux 4 servomoteurs (cf glossaire), commandant leurs positions angulaires.
PWM_MD, PWM_MG : Les PWM envoyés aux moteurs droit et gauche du robot. Permettent de contrôler leurs vitesses de rotation.
SENS_MD, SENS_MG : Signaux numérique indiquant le sens de rotation des moteurs.
5V : Tension d'alimentation des cartes capteurs et des optocoupleurs (cf glossaire) des autres cartes.

A8.2.1. FP1 – REGULER LA TENSION

Principe de fonctionnement :

La carte STM est alimentée par une batterie de 12V. On utilise le régulateur (cf glossaire) de tension 78S05 pour obtenir une tension de 5V stable afin d'alimenter le microcontrôleur (cf glossaire) et les autres composants de la carte. Une diode de protection a été placée avant le régulateur pour éviter d'endommager les composants en cas d'inversion de la polarité.

Dimensionnement

Le régulateur de tension 78S05 ***[12]*** fait partie de la famille des régulateurs linéaire. Ils ont un mauvais rendement, mais ont l'avantage d'avoir un prix très abordable (environ 4 euros) contrairement aux régulateurs à découpage. Comme la partie logique ne consomme pas un courant très important, nous avons choisie de l'utiliser.

La puissance dissipée par le régulateur est de (Vin – Vout) *I, avec Vin la tension délivrée par la batterie, c'est à dire 12V, Vout la tension en sortie du régulateur (donc 5V), et I le courant le traversant, estimé à 0,4A pour la partie logique. Cela nous donne une puissance dissipée de 2,8 W, soit une température avoisinant les Rthj-amb*2,8 = 140°C. Nous avons donc ajouté un dissipateur car cette température est trop proche de la température limite du composant (150°C).

D'après le document constructeur du régulateur, il faut placer deux condensateurs céramiques dont les valeurs sont les suivantes : 0.33 uF en entrée et 0.1 uF en sortie. Cela permet d'avoir une tension « purement » continu, sans ondulations.

A8.2.2. FP2 – GERER L'INTELLIGENCE DU ROBOT

On utilise un STM32-H107 (cf glossaire) comme microcontrôleur (cf glossaire) avec plus des pattes, moins de consommation d'énergie et plus de mémoire que le STM32-H103 utilisé l'année précédente. Pour cela, il a d'abord fallu réattribuer les pattes de STM selon les anciennes et les nouvelles fonctionnalités du robot. En revanche, il est également alimenté en 5V. Les programmes sont stockés dans le mémoire du STM, ils vont donc s'exécuter automatiquement lors du démarrage du robot.

Remarque importante : il faut rajouter des résistances de pull-up aux channels A et B des roues codeuses (cf glossaire). Il arrive que les niveaux haut de la roue codeuse soient mal interprété par le STM car ils ne sont pas parfais, surtout lors des transitions (front montant/descendant). La résistance de pull-up est câblée entre le channel et le 5V.

A8.2.3. FP3 - FOURNIR LE COURANT

Principe de fonctionnement :

Les optocoupleurs (cf glossaire) demandant un courant trop important pour le STM32 H107 (cf glossaire), cette fonction à pour rôle de fournir ce courant. Nous utilisons des buffers (cf glossaire) SN54ABT827 ***[13]*** qui agissent comme des portes logiques non inverseuses, mais qui peuvent fournir un courant plus important en sortie.

Remarque importante : lorsque l'état en entrée de ce composant est en haute impédance (HZ), il génère un niveau logique 1 en sortie, soit l'équivalent d'un PWM avec un rapport cyclique de 1. Ceci est problématique car les moteurs peuvent être alimentés durant la phase d'initialisation du STM où ses entrées sont dans l'état HZ. Pour régler ce problème, nous avons rajouté des résistances de pull down à chaque entrée du buffer, ce qui permet de ramener le niveau logique à 0.

Dimensionnement

Sur les autres cartes, nous utilisons un total de 10 optocoupleurs : 6 pour les PWMs des actionneurs, 2 pour les signaux de sens, et 2 autres pour le retour de courant. Il nous fallait donc un buffer avec 10 entrées. Pour le courant délivré, 15 mA suffit par sortie, et le SN54ABT827 en est capable (maximum 64 mA).

A8.3. LA CARTE MOTEUR

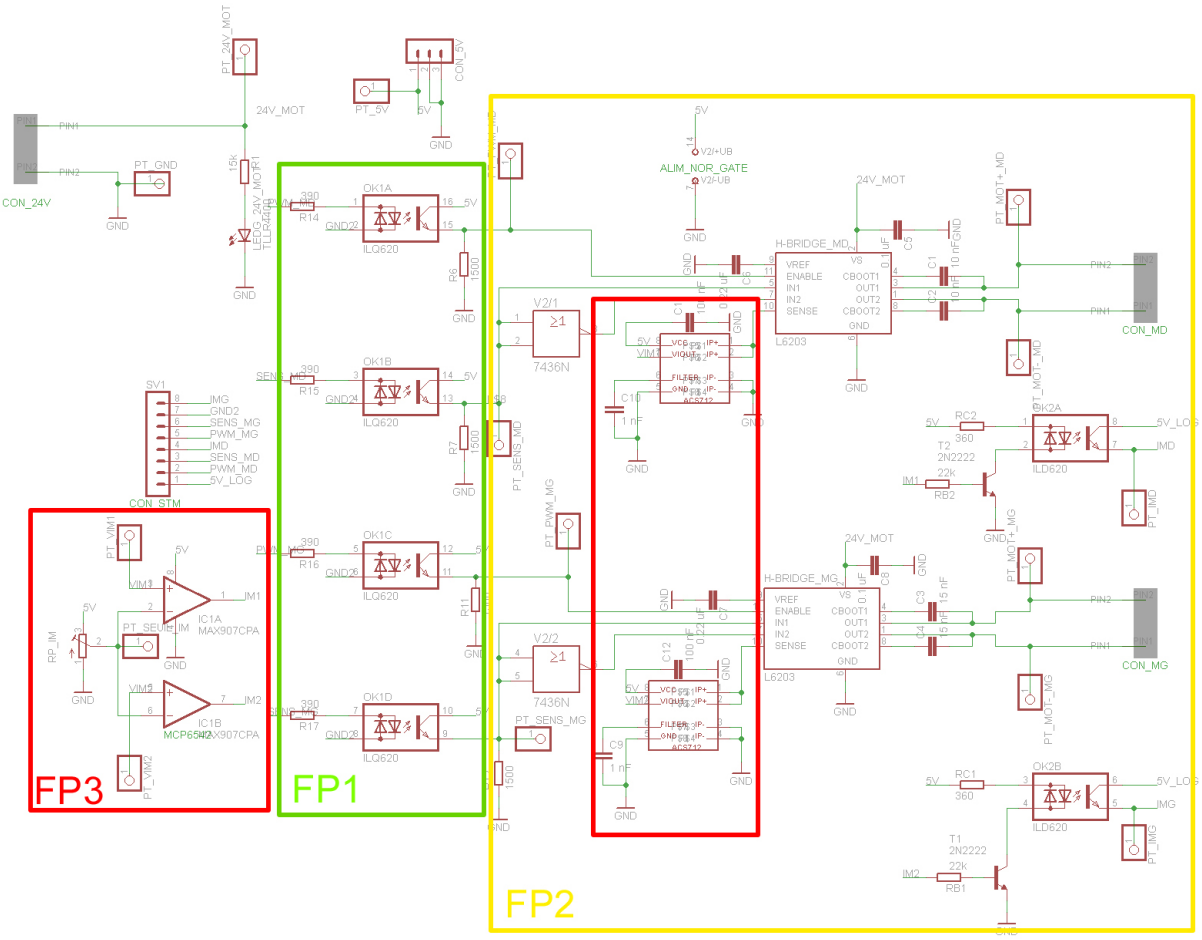


Figure 64: *schéma électrique de la carte moteur*

Entrées :
12V : Tension d'alimentation de 12V provenant de la carte alimentation.
5V : Tension d'alimentation de 5V pour la partie logique de la carte, provenant également de la carte alimentation.
5V_LOG : Tension d'alimentation de 5V des optocoupleurs (cf glossaire), provenant de la carte STM.
GND : la masse commune au circuit de puissance de la carte moteur et alimentation.
GND2 : la masse commune au circuit logique de la carte STM et des optocoupleurs de la carte moteur.
PWM_ MD, PWM_MG : signaux PWM provenant de la carte STM, c'est-à-dire des créneaux de rapport cyclique variable et d'amplitude 5V. Ils permettent de contrôler la vitesse des deux moteurs.
SENS_MD, SENS_MG : signaux numériques SENS provenant de la carte STM. Ils permettent de changer le sens des moteurs.

Sorties :
MOT_MD+, MOT_MD-, MOT_MG+, MOT_MG- : tensions analogiques comprises entre 0 et 12V (par rapport à la masse GND), délivrées aux moteurs, en sortie des ponts en H.
IMD, IMG : Signaux numériques qui indiquent à la carte STM lorsqu'il y a une surintensité qui traverse les ponts en H.

A8.3.1. FP1 : ISOLER LES SIGNAUX

Cette fonction a le même rôle que celle de la fonction FP4 de la carte alimentation. Pour plus de détails, veuillez-vous référer à cette partie.

A8.3.2. FP2 : CONTROLER LES MOTEURS

Le contrôle des moteurs est rendu possible grâce à un composant bien spécifique : le pont en H (cf glossaire). Celui-ci à l'avantage de pouvoir changer le sens des moteurs et de régler leurs vitesses. Le pont en H que nous utilisons, le L6203 ***[14]*** intègre des diodes dites de roue libre, permettant d'éviter d'endommager les transistors du pont en H lors du retour de courant des moteurs. Ces ponts en H agissent comme des bobines, et stockent de l'énergie qu'ils peuvent libérer sur les transistors s'il n'y a pas ces diodes.

Dimensionnement

Un des autres avantages de ce pont en H est sa robustesse : il peut en effet encaisser des courants allant jusqu'à 5A en continu, et 10A durant quelques millisecondes, sachant que nos moteurs ne délivrent pas plus de 2A en continu. Ce surdimensionnement était nécessaire au vu des problèmes que nous avons eus l'année précédente avec ce type de composant. Nous en avons donc choisi un à la fois fiable avec un prix qui reste raisonnable (6 euros).

A8.3.3. FP3 : PROTEGER DES SURINTENSITES

L'intensité provenant des moteurs est récupérée à l'aide d'un composant agissant comme un capteur à effet hall, le ACS712. Les tensions images des courants sont récupérées par deux comparateurs (un pour chaque moteur). Un seuil réglable à l'aide d'un potentiomètre permet de fixer la valeur du courant limite. Si celui-ci est atteint, le niveau logique en sortie du comparateur devient égale à 1, signifiant que le courant qui traverse les moteurs devient critique.

A8.4. PWM

PWM (*Pulse Width Modulation*) (cf glossaire) est une technique utilisée pour obtenir en moyenne, sur une certaine durée, n'importe quelle valeur intermédiaire. Supposons que ß=pulse width / period, on a la tension moyenne égale à ß*U (tension maximale). D'après cela, le pont en H régule le courant continuellement.

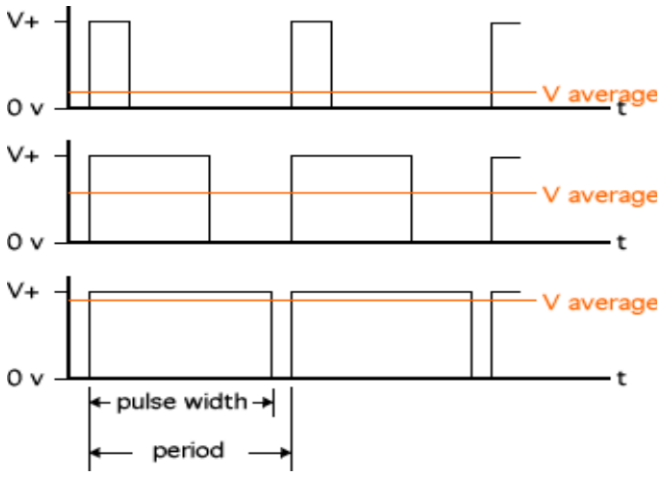


Figure 65: *trois cas différents de PWM*

ANNEXE 9 - DIAGRAMME UML

A9.1. LES CAPTEURS :

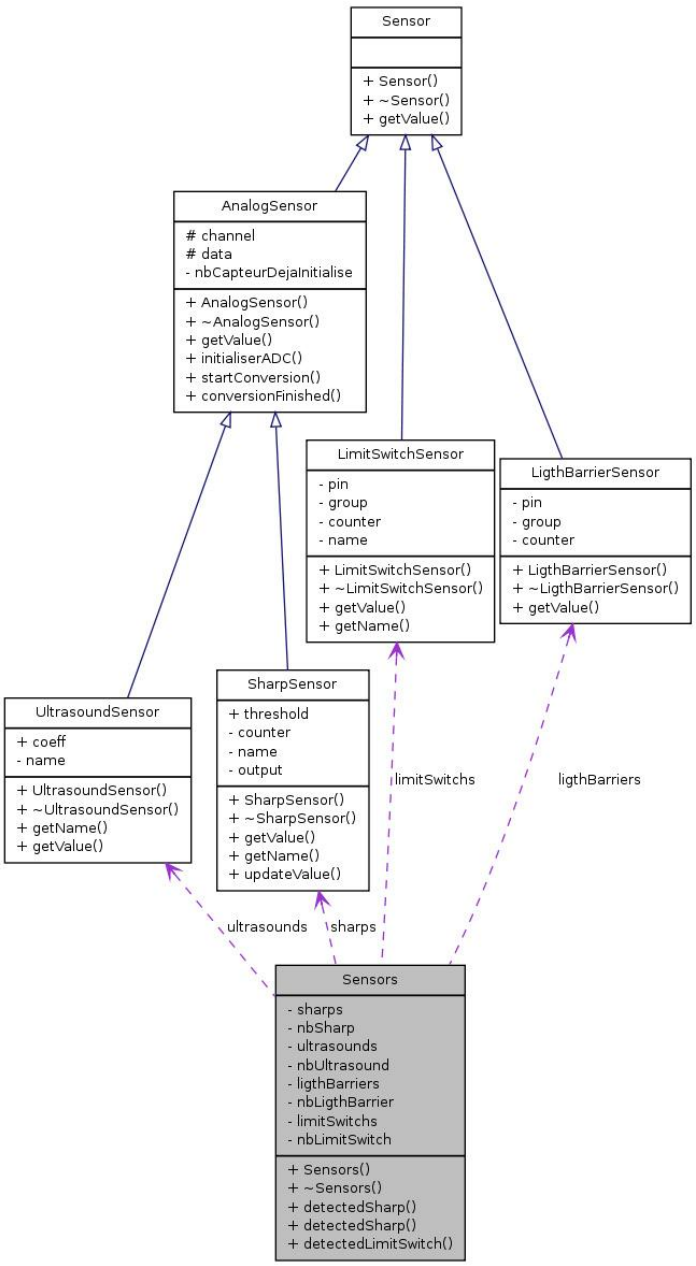


Figure 66: *Diagramme UML pour les capteurs*

Les capteurs sont gérés par un ensemble de classes, qui contiennent tous le mot Sensor. La classe Sensor est la classe de base de tout capteur. C'est une classe abstraite définissant les types, les contraintes et les méthodes de base d'un capteur.

La classe AnalogSensor est aussi une classe abstraite qui hérite de la casse Sensor. Cette classe a pour objectif de regrouper l'ensemble des caractéristiques de base d'un capteur analogique qui nécessite une conversion analogique/numérique.

Quatre autres classes sont ensuite définies, une par type de capteur réel que l'on souhaite manipuler. Ce sont des classes concrètes qui héritent toutes de la classe Sensor (directement ou indirectement par l'intermédiaire de la classe AnalogSensor). Toutes ces classes doivent implémenter une fonction getValue qui retourne une structure spécifiée dans la classe Sensor et qui contient les informations utiles retournées par le capteur réel.

Si l'année prochaine, le club décide d'utiliser un autre type de capteur, il suffira de créer une nouvelle classe qui héritera de Sensor puis d'ajouter ce qu'il faut dans la classe Sensors.

La classe Sensors quant à elle est une classe de gestion de l'ensemble des capteurs. Le reste du code interagira avec les capteurs seulement par l'intermédiaire de cette classe. Elle gère l'ensemble des capteurs et permet de rapidement récupérer les informations utiles retournées par les capteurs réels.

A9.2. LES COMMANDES :

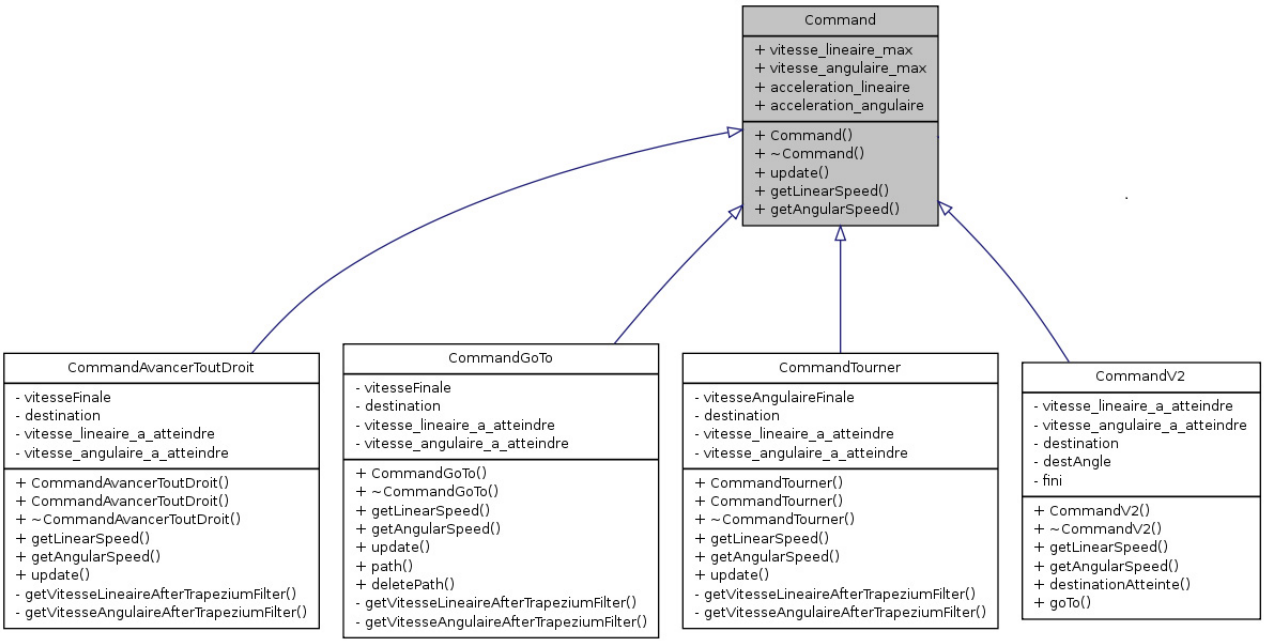


Figure 67: *diagramme UML pour les commandes*

La classe Command est une classe abstraite qui permet de définir les méthodes de base que doivent implémenter toutes les classes Command*. La classe asservissement utilise seulement des objets du type Command pour ne pas à avoir besoin d'adapter le code en fonction des différentes commandes utilisée. Chaque commande hérite de cette classe et a une fonctionne précise. Elles ont toutes pour objectif de générer deux consignes en vitesse (une linéaire et une angulaire). Il ne peut exister qu'une et une seul commande instanciée en même temps en mémoire. En effet, lorsqu'une commandes est instanciée, le constructeur de la classe commande se charge de garder en mémoire l'adresse de cette nouvelle instance et supprime l'ancienne commande, si elle existe, de la mémoire.

A9.3. LA BOUCLE D'ASSERVISSEMENT :

La classe d'asservissement est la classe centrale du code. C'est à partir d'elle que toutes les informations sont mises à jour et c'est elle qui donne la main à la stratégie pour prendre des décisions. Le microcontrôleur (cf glossaire) lance des interruptions toutes les 10ms qui appellent la méthode update de la classe asservissement. C'est cette méthode update qui lance les updates de l'odométrie, de la stratégie et des capteurs. C'est aussi dans cette classe (comme son nom l'indique) qu'est effectué la boucle d'asservissement (voir ***annexe 3*** pour plus de détails).

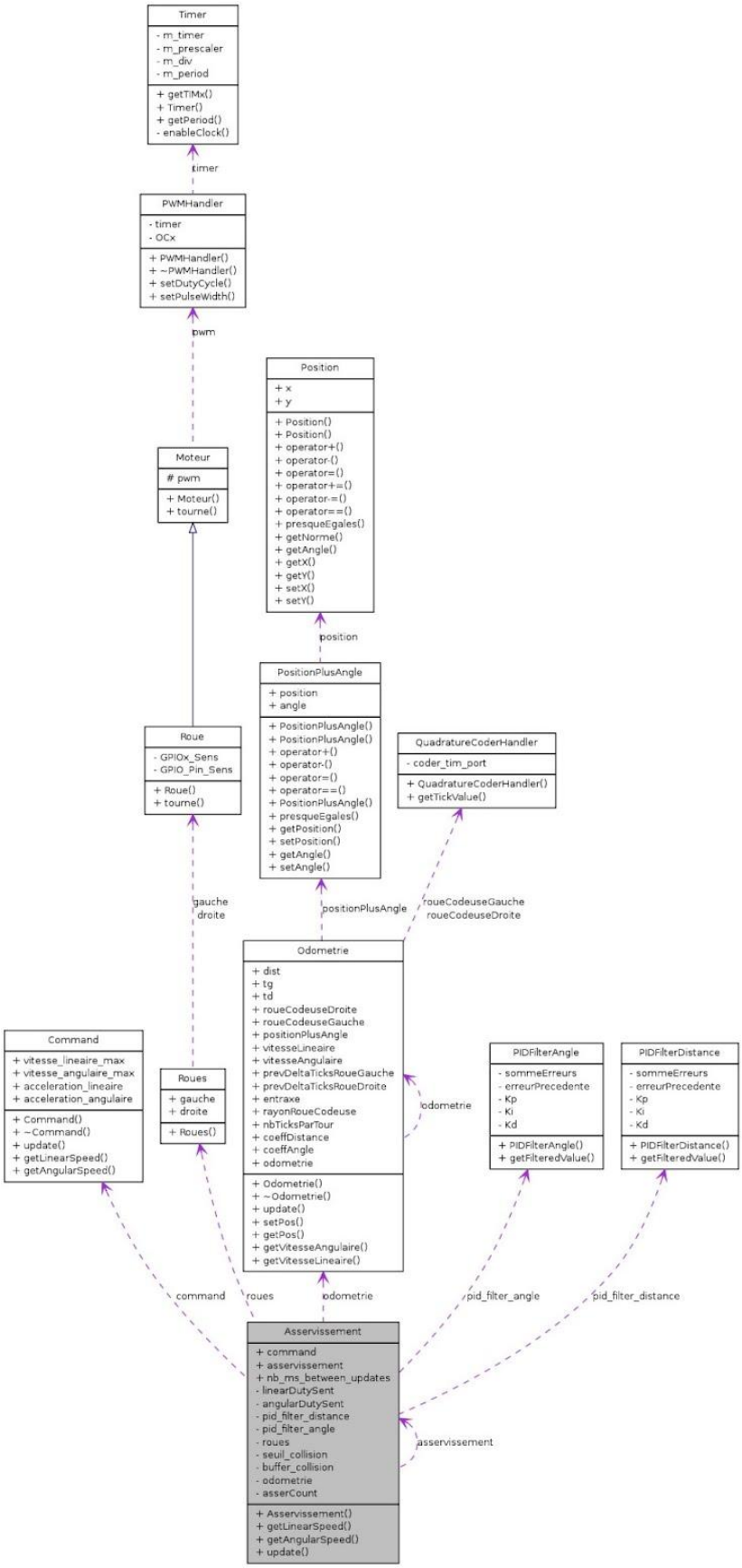


Figure 68: *Diagramme UML de l'asservissement*

A9.4. LA STRATEGIE :

La structure de la stratégie est assez simple. Il existe une classe appelée stratégie, qui possède une méthode update() appelée par l'asservissement à chaque mise à jour. Cette classe sert d'interface avec les classes actions qui, elles, décident de lancer des commandes permettant d'avoir globalement des actions cohérentes. La stratégie gère la transition entre différentes actions et les actions gèrent la transition entre différents ordres.

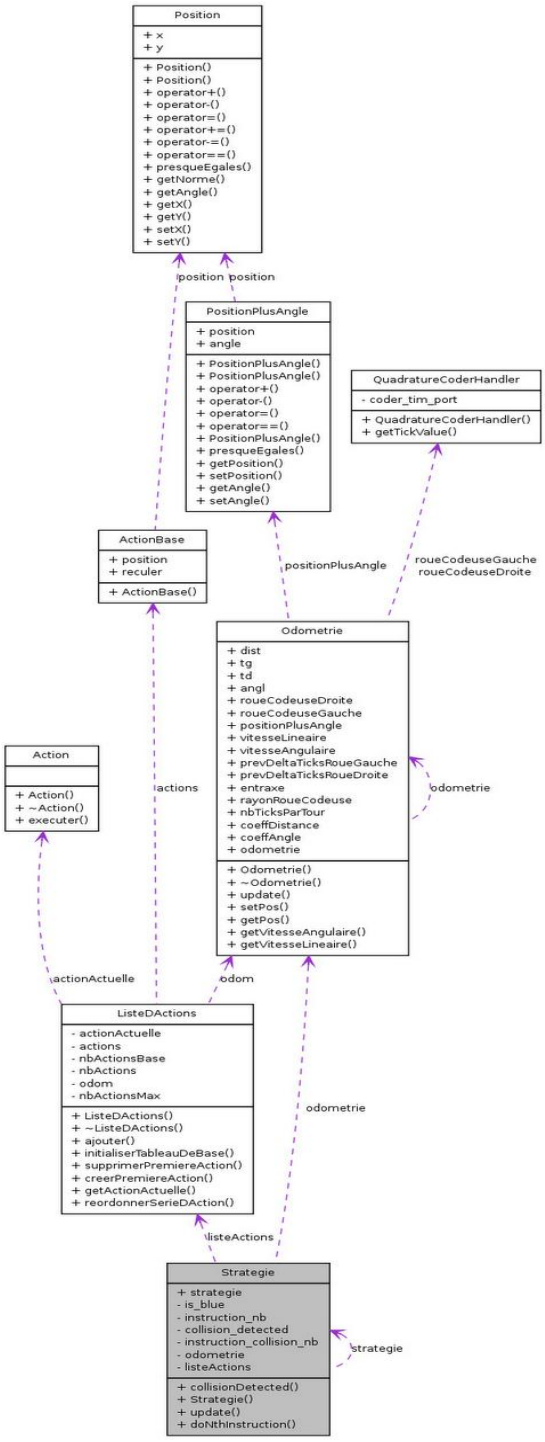


Figure 69: *Diagramme UML de la stratégie*

www.telecom-bretagne.eu

Campus de Brest
Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 3
France
Tél. : + 33 (0)2 29 00 11 11
Fax : + 33 (0)2 29 00 10 00

Campus de Rennes
2, rue de la Châtaigneraie
CS 17607
35576 Cesson Sévigné Cedex
France
Tél. : + 33 (0)2 99 12 70 00
Fax : + 33 (0)2 99 12 70 19

Campus de Toulouse
10, avenue Edouard Belin
BP 44004
31028 Toulouse Cedex 04
France
Tél. : +33 (0)5 61 33 83 65
Fax : +33 (0)5 61 33 83 75