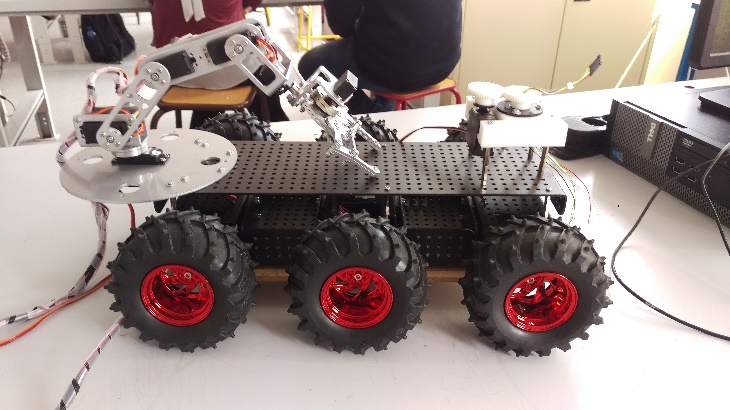
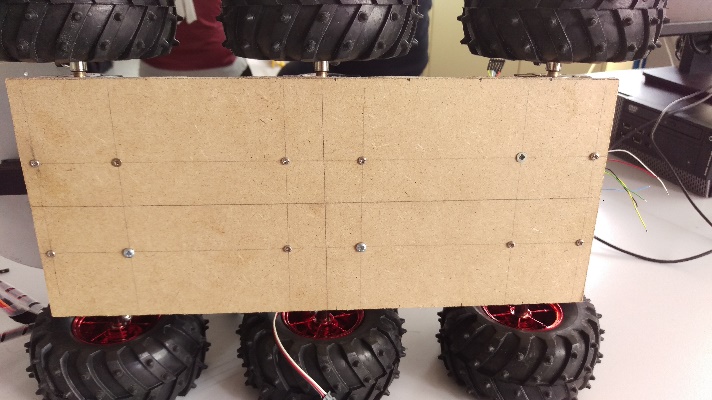
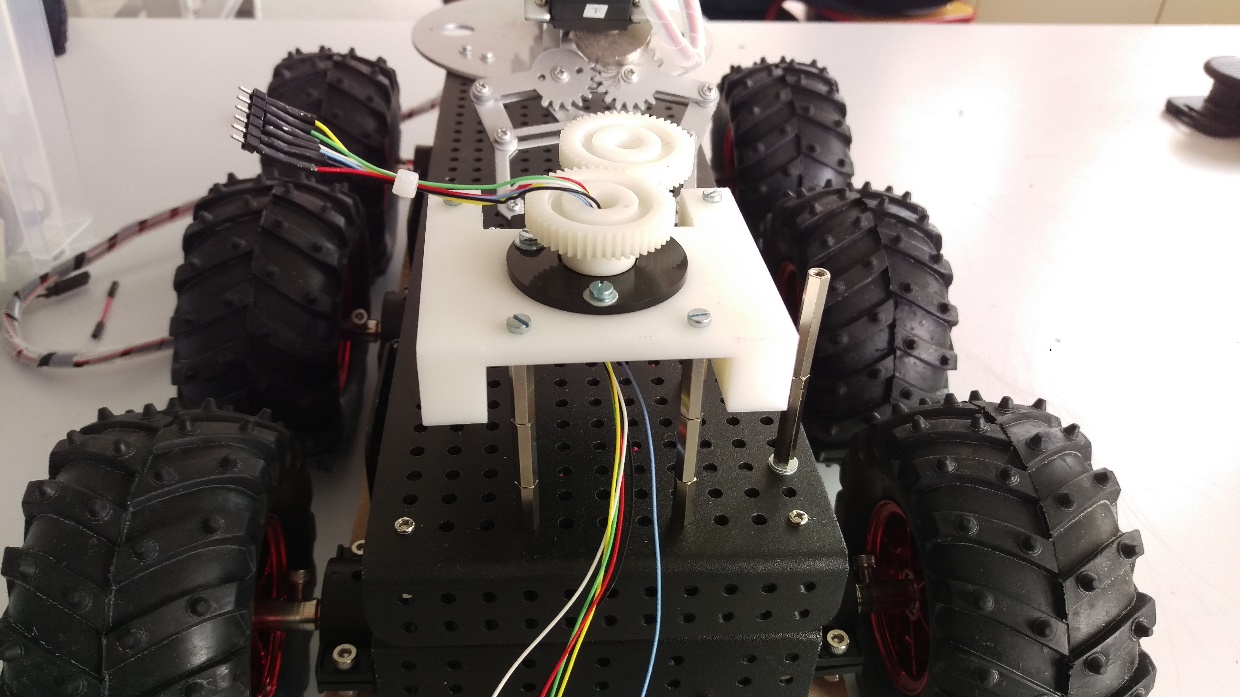
s

**Projet ROBLIP6**



000000000000000



**Client** : Mr ILLIE

**Tuteur pédagogique** : Olivier ROBERT

**Equipe projet** : Raphaël BRESSON, Isabelle ZHENG, Charles ROGER

**Date de rendu** : Mai 2016

**Table des matières**

[**Introduction** 4](#_Toc451094970)

[**Remerciement** 4](#_Toc451094971)

[**Cahier des charges** 5](#_Toc451094972)

[**Contexte du projet** 5](#_Toc451094973)

[**Identification du besoin** 5](#_Toc451094974)

[**Contraintes :** 6](#_Toc451094975)

[**Mécaniques** 6](#_Toc451094976)

[**Electriques** 6](#_Toc451094977)

[**Hardware** 6](#_Toc451094978)

[**Software** 7](#_Toc451094979)

[**Analyse fonctionnelle** 7](#_Toc451094980)

[**Notre méthodologie** 9](#_Toc451094981)

[**Le travail de groupe** 9](#_Toc451094982)

[**Le travail avec le client** 9](#_Toc451094983)

[**Nos objectifs** 9](#_Toc451094984)

[**La gestion du temps** 10](#_Toc451094985)

[**Diagramme de Gantt prévisionnel** 10](#_Toc451094986)

[**Diagramme de Gantt réel** 10](#_Toc451094987)

[**Les tâches additionnelles** 10](#_Toc451094988)

[**Le travail réalisé** 11](#_Toc451094989)

[**Architecture proposée** 11](#_Toc451094990)

[**Modification de la base mécanique** 12](#_Toc451094991)

[**Modification du servomoteur** 14](#_Toc451094992)

[**Réalisation des pièces 3D** 15](#_Toc451094993)

[**Système d’assemblage du servomoteur et du Lidar** 15](#_Toc451094994)

[**Partie programmation** 16](#_Toc451094995)

[**Choix d’un FPGA** 16](#_Toc451094996)

[**Initialisation séquentielle de la communication** 17](#_Toc451094997)

[**Trame d’envoi d’ordre** 17](#_Toc451094998)

[**Trame d’envoi de donnée** 18](#_Toc451095000)

[**Partie software** 19](#_Toc451095001)

[**Partie Hardware** 21](#_Toc451095003)

[**Test unitaire** 22](#_Toc451095004)

[**Reprise du projet** 24](#_Toc451095005)

[**Liste des figures** 25](#_Toc451095006)

[**Annexe 2 : Plan support du Lidar** 26](#_Toc451095007)

[**Annexe3 : Plan du support servomoteur** 27](#_Toc451095008)

# **Introduction**

Ce projet est particulièrement intéressant de par son originalité et sa difficulté technique. Outre nous fournir l’opportunité et la chance de nous former sur une multitude de technologies nous avons eu la possibilité de le façonner en accord avec notre client et l’équipe pédagogique. En effet ni le besoin ni le cahier des charges n’étaient clairement établis. Nous avons réussi grâce à de nombreuses réunions avec notre client à nous approprier les deux. Nous vous proposons dans ce document une explication des deux documents susnommés.

# **Remerciements**

A travers ce rapport nous tenons à remercier tous les intervenants du projet qui nous ont permis d’arriver aussi loin. Nous tenons premièrement à remercier Mr Illie pour sa disponibilité, son sérieux et l’aide qu’il nous à apporter. Nous avons pu à chaque séance présenté nos idées et avancement, ce qui nous à permis d’avancer dans de bonne condition et d’augmenter notre productivité.

Nous tenons ensuite à remercier Mr Viateur et Mr Dinoire pour leur aide technique et leurs conseils toujours plus qu’utile. En effet ils ont été dextrement disponible pour nous aider à surmonter nos difficultés techniques que ce soit électronique informatique ou bien mécanique.

Nous voulons aussi remercier Mr Robert, notre tuteur pédagogique qui nous à suivit pendant le projet. Il nous à apporter beaucoup de recul, d’organisation et de sagesse à travers ses différents conseils. Nous voulons aussi le remercier pour sa pédagogie et nous avoir fait profiter de son expérience.

Enfin nous tenons à remercier Mr pecheux qui a été extrêmement disponible pendant le projet mais aussi en dehors du projet pendant les différentes réunions. Il nous a conseillé dans beaucoup de situation compliquée et a su nous aidé dans l’établissement d’un cahier cahier des charges et la séparation des différents travaux à faire.

# **Cahier des charges**

## **Contexte du projet**

Avant de décrire le besoin ou même le cahier des charges nous tenons à fixer le contexte de notre projet. Notre client est Mr Jean-Michel Illie chercheur au laboratoire d’informatique de paris 6 notre responsable pédagogique est Mr Olivier Robert, Nous tenons aussi à préciser que Mr Francois Pecheux nous à apporter une aide plus que précieuse.

Notre client, expert en informatique, cherche à faire une série de mesure et de test sur l’environnement ambiant. Il peut par exemple chercher à comprendre et améliorer la prise en compte de l’environnement par le robot. L’environnement est pour nous ce qui entoure le robot, nous pouvons ainsi inclure les murs, tables, chaises et tous autres objets.

Nous voulons préciser que tous les tests se feront en intérieure et dans un laboratoire. Il n’y a donc pas de contrainte particulière en matière d’étanchéité ou d’électromagnétisme.

Notre client spécialisé en informatique et développement logiciel ne possède pas de compétence spécifique en robotique ou systèmes mécanique il est donc primordial de réaliser un système performant répondant au besoin mais surtout restant simple et le plus fiable possible. La fiabilité est la contrainte principale de ce projet. Vous pourrez par la suite vous rendre compte que cette contrainte nous a par exemple demandée beaucoup de temps.

## **Identification du besoin**

Notre client a besoin d’un système roulant autonome capable de prendre en compte son environnement afin d’interagir avec. Il doit être capable de reconnaitre une pièce ou un mobilier représenté par un code aruco’s. Nous devons être en mesure de prendre des objets grâce à un bras robotisé.

Le robot doit être capable de détecter des obstacles et les répertorier dans une carte. Il doit pouvoir calculer sa distance par rapport à ce dernier et le cibler afin de se diriger en sa direction. Nous devons être en mesure de détecter la présence de cet obstacle à courte distance.

Le système doit être en mesure d’exécuter une application java fournit par notre client. C’est pourquoi nous devons crée une abstraction logicielle entre la couche qui contiendra les capteurs, actionneurs. Et la couche de contrôle qui contiendra l’intelligence du robot et les capteurs plus compliqués comme les caméras.

## **Contraintes :**

Nous vous proposons une analyse des contraintes sous formes d’items. La grande contrainte de notre projet est que notre client avait déjà acheté toute les pièces du système avant le début du projet. Dans la liste ci-dessous beaucoup des contraintes sont issues des documentations techniques des composants notemment en termes de poids et valeurs électrique.

### **Mécanique**

Les contraintes mécaniques sont les contraintes de formes, poids. Vous pourrez vous rendre compte dans l’analyse des problèmes que cette partie nous a causé énormément de problèmes et nous à couter beaucoup de temps. En effet le chassis WilD Thumper et le bras robotique DAGU ont dû être modifié avant leurs utilisation. Vous pourrez par ailleurs trouvez en annexes un récapitulatif des principales valeurs caractéristiques des deux composants susnommés.

-**système roulant 6 roues** (chassis WILD THUMPER)

-**charge utile** maximum : 5KG

-**Présence d’un bras robotisé** (bras robotique Dagu)

Le bras robotique présente 5 degrés de liberté avec une rotation de 180° par degré de liberté

-**Dimension maximal** : 30cm\*50cm

Le système doit être suffisamment grand pour être une station d’accueil de petit drone. Nous n’avons cependant pas identifié de taille minimale.

### **Electrique**

Les contraintes électriques sont exprimées à travers les tensions d’alimentation des différents composants. En voici une description.

-**Alimentation de la base** : 7.2V

-**Alimentation du bras DAGU** : 4.8 à 6V

-**Alimentation du FPGA** : 5V ou 12V

-**Alimentation de la carte haut niveau (firefly)** : 5V

### **Hardware**

Nous avons besoins de récupérer les données des capteurs et de contrôler les actionneurs en parallèle. Mais aussi permettre une abstraction suffisaient aux niveaux pour pouvoir exécuter les algorithmes d’intelligence artificielle.

-FPGA (parallélisation des acquisitions des capteurs et du contrôle des actionneurs)

-Carte ordinateur embarqué pour permettre l’abstraction logicielle

### **Software**

Il s’agit de la partie haut niveau devant permettre de nous élever en abstraction. Le but étant de contrôler le robot

-Abstraction logicielle composée de plusieurs couches correspondant à différent niveau d’abstraction.

- Elle est organisée sous forme de package afin de la rendre adaptable et facilement interfaçable.

-L’exécution doit être multithread

-Le système doit pouvoir reconnaitre des codes Aruco

-Le système doit pouvoir reconnaitre des formes et des couleurs

## **Analyse fonctionnelle**

Voici ci-dessous le Diagramme FAST de notre système.

Déplacer le système et attraper des objets

Diriger le robot

FP1

Choisir Chemin

FC1

Générer carte des obstacles

FC2

Récupérer les valeurs des capteurs

FC6

Faire avancer le robot

FP2

Commander les moteurs

FC3

Prendre un objet

FP3

FC4

Bouger bras

Commander le système

FP4

Interfacer L’API et la couche bas niveau

FC5

Envoyer la commande

FC7

Caméra/Lidar/capteurs de présence

Hacheur/carte commande de moteur

Servomoteur/ bras mécanique

Carte Firefly(linux embarqué)

Microcontrôleur/ FPGA

Figure : Diagramme FAST

Voici ci-dessous un diagramme pieuvre de notre système

**FP2**

**FP1**

**FP4**

**FP3**

**FC7**

**FC9**

**FC6**

Figure : Diagramme pieuvre

**Fonction principale 1 ou diriger le robot**: Le système doit être capable de se diriger de manière autonome en prenant en compte son environnement.

**Fonction principale 2 ou faire avancer le robot**: Le système doit être de fournir les bons signaux de commandes aux actionneurs en fonction du retour des capteurs.

**Fonction principale 3 ou prendre un objet**: Le robot doit être capable de prendre un objet de son environnement et le déplacer

**Fonction principale 4 ou commander le système** : Le robot doit être capable de commander automatiquement les différents bus et les actionneurs

# **Notre méthodologie**

## **Le travail de groupe**

Le travail à réaliser étant très dense nous avons décidé de nous répartir le travail pour que chacun puisse avancer de son côté et mettre en commun pendant les séances de projet. Nous avons décidé d’attribuer 10 minutes le matin en début de journée pour faire le point sur l’avancée des travaux en cours. Par ailleurs à chaque séance nous avons fait une réunion de durée variable avec notre client. Nous avons choisi de créer un Git Hub de manière à centraliser nos travaux et documents, ce qui nous a permis de faire des documentations petit à petit pour chaque tache terminée. Pour organiser notre temps nous avons choisis de faire un diagramme gantt sous Excel. En effet la version 2016 étant très largement simplifié et beaucoup plus intuitif nous avons remarqué qu’une feuille de calcul excel est beaucoup plus simple à modifier ou à partager.

## **Le travail avec le client**

Notre client étant très présent et disponible nous avons eu la possibilité de faire valider notre travail et nos idées très souvent (à chaque séance et même pendant la semaine). Nous avons fait des réunions à chaque séance ce qui nous a permis d’être clair sur les objectifs mais nous a aussi couter un petit peu de temps. Nous pensons que l’un des apports du projets est justement l’apprentissage de cette compétence. Grace à ce projet nous avons appris à répondre aux attentes d’un clients, à faire et préparer des réunions, des démonstrations. Nous avons appris à avoir une approche client et pas forcément technique.

## **Nos objectifs**

Durant ce projet nous avons essayé de répondre le plus fidèlement possible au besoin de notre client, nous avons essayé de le conseiller sur des choix techniques de manière à conserver la fiabilité du système. Nous sommes conscients que le projet industriel est une phase importante de notre apprentissage. C’est pourquoi nous avons essayé de nous former sur le plus de technologie possible. Que ce soit le Lidar, la caméra, le FPGA ou les autres organes du robot. Nous voulions aborder des thématiques non vus en cours, et après réflexions le module de projet est très certainement fait pour ça. Nous voulions nous former au contact client, à comment être technique et productif tout en restant simple dans nos explications.

# **La gestion du temps**

Les deux diagrammes de Gantt étant sur un semestre il est difficile de les lire si on les ramène sur une seule page nous vous les faisons parvenir dans le dossier de rendu du projet dans le classeur Excel intitulé Gantt\_Projet\_Roblip6

## **Diagramme de Gantt prévisionnel**

Afin d’organiser le travail à faire durant le projet nous avons réalisé un gant prévisionnel. Nous tenons à préciser qu’il n’est pas réel car beaucoup de tache sont apparu pendant le projet au fil des réunions avec notre client. Ceci nous a fait comprendre l’importance de garder une marge de temps dans la réalisation d’un projet. En effet nous avons dès le début réaliser un diagramme de Gantt très chargé et peut être un petit peut optimiste à cause du fait que nous avons eu du mal à nous accorder sur des objectifs précis en début de projet.

## **Diagramme de Gantt réel**

Le diagramme de Gantt réel est différent du prévisionnelle dans le sens ou beaucoup de tache ce sont rajouté au faire et à mesure de l’évolution du projet. De plus nous avons changé de solution technologique pour l’étage bas niveau ce qui a eu pour effet de rallonger le temps réservé au développement de cette couche. En effet nous avons décidé de passer sur un FPGA de manière à parallélisé toutes les actions choses qui était impossible à faire facilement avec un microcontrôleur. De plus le bras demandant 6 signaux PWM et la base 2 la solution à base de FPGA était pour nous la plus viable, elle permet de créer autant de signaux PWM que nous voulons. Nous n’avions pas non plus prévu de passer autant de temps sur les problèmes mécanique du bras ou de la base.

## **Les taches additionnelles**

Voici ci-dessous les taches et réflexions qui se sont rajoutés au faire et à mesure de l’évolution du projet ainsi que les changements qu’il impliques sur le travail initialement prévu.

* Comment avoir un acknowledge des actions du servomoteur et du bras
  + Plusieurs réunions sur le temps projet
* Rajout d’une caméra pour reconnaitre les formes sur l’avant du bras
  + Développement du code
* Rajout de capteurs de distance
  + Modification de la base pour rajouter des supports

# **Le travail réalisé**

Dans cette partie nous allons évoquer chaque tache en expliquant la méthode de réalisation, les problèmes que nous avons rencontré lors de la réalisation de la tâche. Nous ferons à la fin une sous parti consacré aux test unitaire de chaque réalisation. Nous voulons préciser que nous n’évoquerons pas chaque tache réalisé mais seulement les principales ou celles nous ayant demandé le plus de temps.

## **Architecture proposée**

Voici ci-dessous un schéma de l’architecture générale que nous proposons pour répondre aux besoins du client. Cette architecture représente principalement l’interaction entre les deux cartes Nous précisions que cette architecture peut être soumise à modification si le système évolue.

**Low Level - FPGA**

* Gestion bas niveau des capteurs
* Gestions bas niveau des actionneurs
* Réception des ordres
* Envoi des données des capteurs

Lidar

I2C

UART

**High Level - Firefly**

* Abstraction logicielle
* Envoi des ordres
* Réceptions des données capteurs

UART

Données

CAPTEUR

Ordre

UART

Roue PWM

Servomoteurs PWM

Camera

USB

Figure : Architecture générale

## **Modification de la base mécanique**

**Caractéristique générale :**

La base robotisé est composée de deux fois deux roues commandé en même temps (3roues pour la partie gauche et 3 pour la partie droite) chaque partie est commandé par une unique PWM. Voici les caractéristiques principales de notre base :

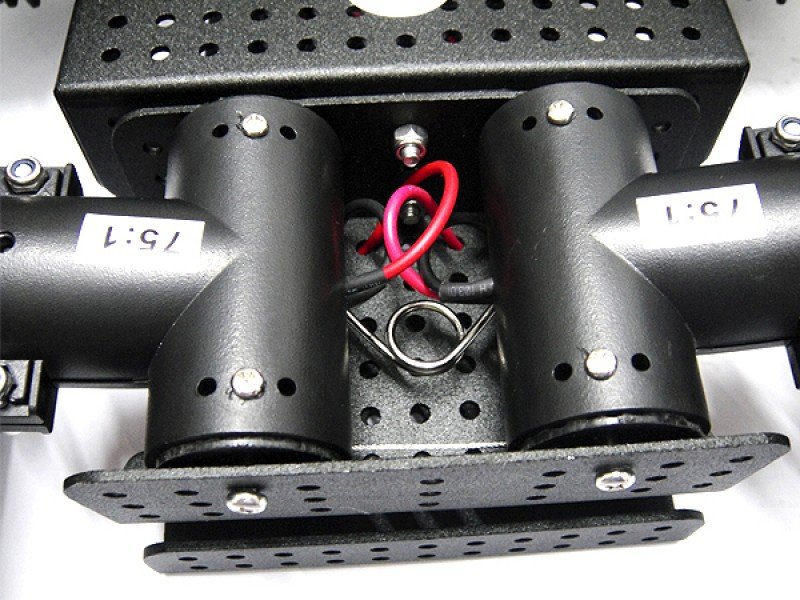
|  |  |
| --- | --- |
| Alimentation | 7.2Vcc |
| Intensité à vide | 350mA/moteur |
| Couple de blocage | 8.8kg.cm |
| Charge utile | 5Kg |
| Dimension | 430\*285\*130 |
| Dimension du plateau | 376\*120 |
| Dimensions des roues | D = 125\*60mm |
| Diamètre axe des roues | 4mm |
| Garde au sol | 60mm |
| Poids total | 2.86Kg |

Pour de plus amples informations voici le lien vers le vendeur de la base :

<http://www.gotronic.fr/art-chassis-wild-thumper-12395.htm>

**Observation des problèmes**

Le chassi wild Thumper présentait une fonctionnalité intéressante qui est un système d’amortissement, système qui au final s’est révéler plus une source de problème qu’un avantage. Le système d’amortissement est composé de ressorts et câble en acier



Ressort unique

Câble en acier

Figure : Illustration défaut de la base

Le ressort tend à rapprocher les points 1 et 2 ce qui a pour effet d’écarter les points 3 et 4. Ce problèmes était normalement annuler par le câble en acier qui contrôle l’écart entre 3 et 4. Toute fois nous avons remarqué que le câble est trop long et permet un trop grand mouvement au points 1 et 2. Voici une Schéma explicatif de notre problème.

Figure : Illustration défaut des carters moteur

Carter des moteurs

Position parfaite des axes moteurs

Position des axes moteurs réel

Pour régler ce problème nous avons décidé dans un premier temps de supprimer le ressort et le câble en acier pour enlever toutes les contraintes. Nous avons par la suite remarqué que cette modification n’est pas suffisante nous avons donc pris la décision de bloquer tous les moteurs ensemble grâce à une planche de bois. Voici un Schéma expliquant la modification

Figure : Explication des modifications de la base

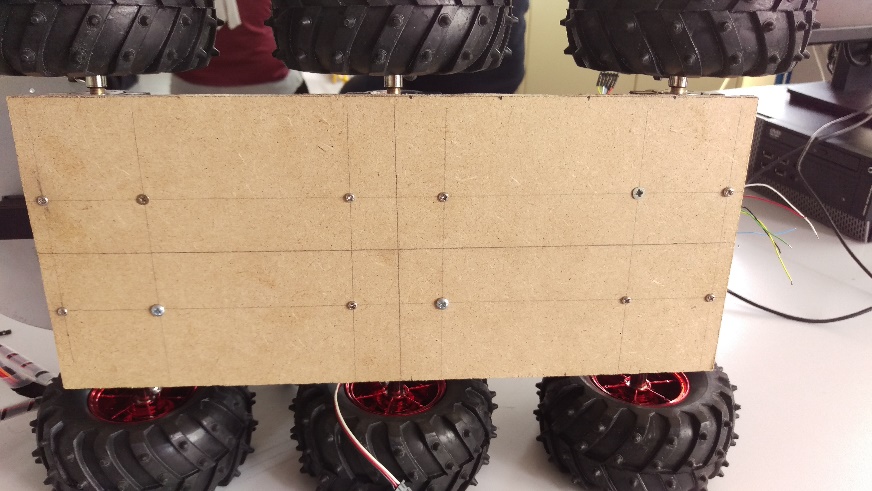


Figure : Photo des modifications

Châssis moteur

Planche de bois

Moteur

Vis de fixation

**Problème**

-Les carters en T visible ci-dessus englobe nos moteurs ils sont réalisés en plastique ainsi que les pas de vis qui ont été usiné très serré. Lors du dévisage des ressort le pas de vis est resté collés sur les vis ce qui a rendu la fixation de la plaque compliquée.

-Les supports des moteurs sont normalement plier avec un angle exact de 90° ce qui permet une fixation parfaite des moteurs. Hors quand les moteurs poussaient vers un coté ils ont augmenté cet angle ce qui a pour effet de ne plus tenir correctement les moteurs. Voici ci-dessous un schéma expliquant le problème.

Figure : Explication du déformement des carters moteur

## **Modification du servomoteur**

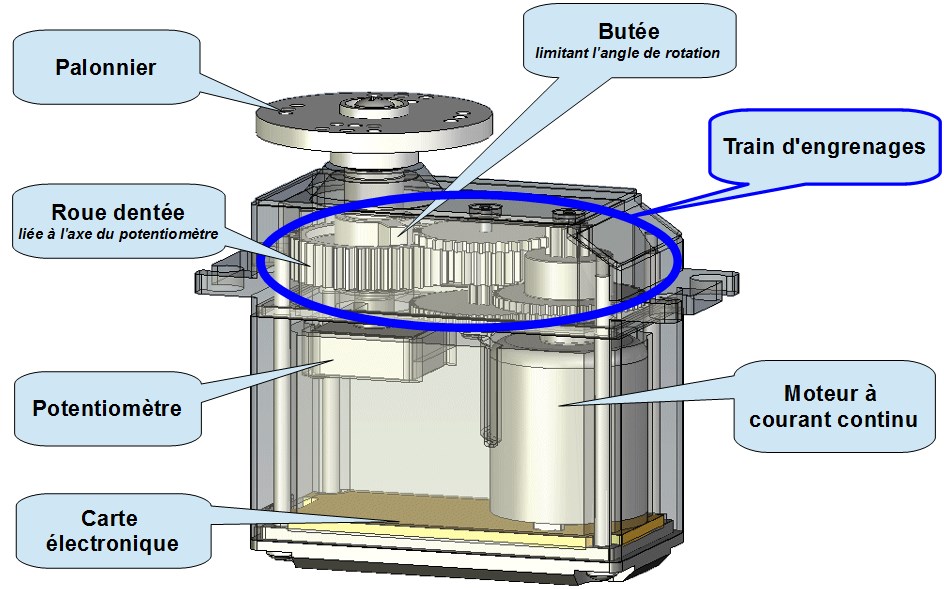
Pour évaluer la distance des objets par rapport au robot nous utilisons un capteur Lidar. Ce capteur à un fonctionnement proche d’un radar, il fait des mesures à 360° si on lui permet de faire une rotation de 360°. Nous avions besoin d’un actionneur permettant cette rotation, nous avons choisi un servomoteur, Nous avons décidé avec Mr Pecheux d’en modifier un afin d’éviter d’attendre un temps de livraison et de retarder plus le projet. Voici ci-dessous le schéma d’un servomoteur classique.

Figure : Vue éclatée servomoteur

Nous attirons votre attention sur la butée limitant l’angle de rotation. La limitation est variable en fonction du servomoteur mais en enlevant simplement cette butée le servomoteur n’est plus limité.

Le système est aujourd’hui monté sur le robot et fonctionnelle.

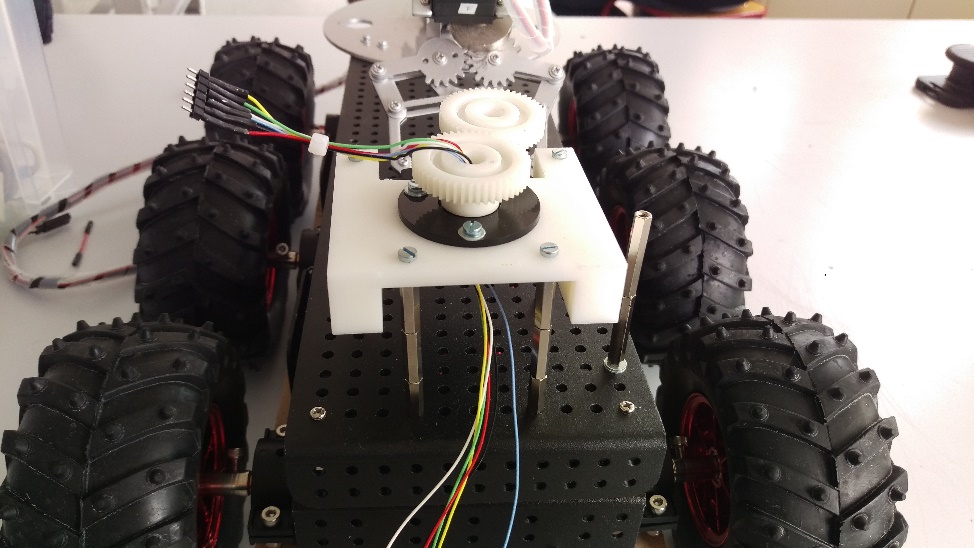
Mr Dignoire et Mr Pecheux nous ayant beaucoup accompagné dans cette tache nous n’avons pas rencontré de problème spécial.

## **Réalisation des pièces 3D**

La base du robot propose une plateforme d’accueil avec un ensemble de trou permettant de fixer les différents organes du robot. Chaque trou étant écarté de 1cm avec ses voisins. Le problème que nous avons rencontré est que cette norme n’est apparemment pas partagé avec les constructeurs de cartes et composants électroniques. Aucun de nos composants ou organes (servomoteurs, cartes firefly, carte de gestion des moteurs ou bras robotisé) ne partageant cette norme nous avons dû créer un support pour l’ensemble ce qui nous à demander énormément de temps. Dans ce rapport nous parlerons uniquement des deux pièces permettant d’assemblé le servomoteur modifié et le Lidar.

## **Système d’assemblage du servomoteur et du Lidar**

Nous utilisons deux engrenages en matière plastique afin d’entrainer le Lidar dans sa rotation. Un engrenage est directement placé sur le palonnier du servomoteur et l’autre sur la pièce que nous avons créé expressément à cet effet



Notre client Mr Illie a mis à notre disposition un système permettant de brancher le lidar sans créer de torsion sur les câble. Vous pouvez voir ce système sur la pièce imprimé.

Nous mettons à votre disposition les plans des deux pièces d’assemblages en annexe 3.

Figure : Support de la tourelle du Lidar

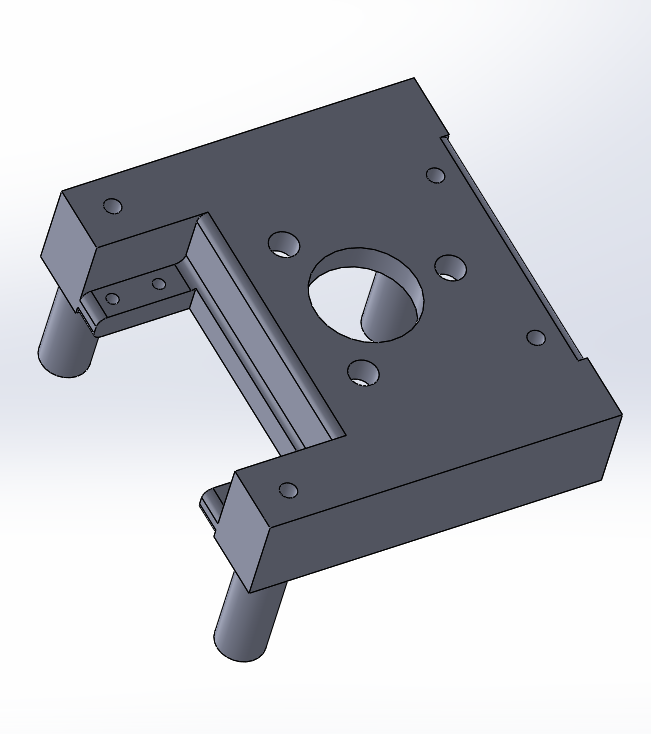


Figure : Pièce 3D de support du Lidar

## **Partie programmation**

### **Choix d’un FPGA**

Pour réaliser le projet roblip6 nous nous sommes très vite rendu compte qu’un microcontrôleur était restrictif c’est pourquoi nous avons choisi d’utilisé un FPGA. Le paragraphe ci-dessous explique ce choix plus en détail.

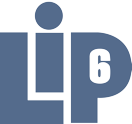
**Parallélisation de la gestion des capteurs et des actionneurs :**

L’un des intérêts principaux du FPGA est de pouvoir paralléliser les calculs ainsi par exemple on peut commander les commander les moteurs, les servomoteurs et traiter les capteurs bas niveaux en même temps.

**Le nombre d’entrées sorties :**

Cette carte possède beaucoup plus d’entrées sorties ce qui nous permet de directement crée nos PWM mais aussi d’envisager de plus grande amélioration du projet pour les années futures.

**La présence d’un FPGA couplé avec un microcontrôleur :**

Sur cette carte sont présent un FPGA SPARTAN 6 et un micro-processeur ATMEGA. Ceci nous permet de rendre le système beaucoup plus performant et réactif car les taches demandant beaucoup de ressources (comme les multiplications de flottant ou divisions) peuvent être réalisées sur le FPGA pendant que les taches plus simples sont réalisées simultanément sur le microcontrôleur.

Nous faisons dans ce rapport une unique section pour la partie informatique puisque nous vous proposons une documentation détaillée pour chaque partie du code dans le rendu du projet. Nous proposons entre autres une documentation de notre code FPGA pour les IPs I2C, pwm et autres mais aussi une java doc pour le code java. Nous avons donc décider dans le rapport d’expliquer la partie la plus primordiale de la partie informatique la communication entre les deux cartes. Communication se faisant grâce à un protocole UART. On peut distinguer deux types de communication un envoie d’ordre et une réception de donnée pour la partie haut niveau, et une réception d’ordre ou envoie de données pour la couche bas niveau. Une image valant 1000 mots nous vous proposons les shémas suivant pour illustrer une initialisation et une communication.

### **Initialisation séquentielle de la communication**

Signal de reveil

(11111111)2 = (255)10

Nombre de capteurs

Nombre d’actionneurs

Identifiant

type

1

2

3

4

**LOW LEVEL**

**HIGH LEVEL**

- ID (8 bits)

- Données (32 bits)

- ID (8 bits)

- Données (32 bits)

Pour chaque capteur

type

6

7

Pour chaque actionneur

5

Figure : Initialisation de la communication UART

Identifiant

### **Trame d’envoi d’ordre**

ID (8bits)

Ordre (8bits)

1

2

**LOW LEVEL**

**HIGH LEVEL**

- ID (8 bits)

- Données (32 bits)

- ID (8 bits)

- Données (32 bits)

Figure : Envoi d'un ordre avec l'UART

### 

### **Trame d’envoi de donnée**

Figure : Envoi de donnée avec l'UART

ID (8bits)

Donnes (31->24)

Donnes (23->16)

Donnes (15->8)

Donnes (7->0)

1

2

3

4

5

**LOW LEVEL**

**HIGH LEVEL**

- ID (8 bits)

- Données (32 bits)

- ID (8 bits)

- Données (32 bits)

**DELAY**

## **Partie software**

Dans cette partie nous vous proposons 3 schémas. Un premier pour l’architecture Java que nous avons établie avec notre client, un second pour l’avancement actuel de cette tâche et un dernier expliquant la communication avec la partie Low Level qui est géré par le FPGA

Classes de gestion de communication vers la carte LOW LEVEL

(UART)

Classes de gestion de la caméra

Abstraction de 1er niveau

Gestion des capteurs et des actionneurs via des classes JAVA

Ordres

Données brutes des capteurs externes

Classes de génération de la carte des obstacles

(OPENCV + LIDAR)

Carte des obstacles

Code IA

Abstraction de 2nd niveau

Identification d’objets

Ce Schéma montre l’avance actuelle du la programmation High Level on peut se rendre compte que la carte des obstacles n’est pas encore réalisée ainsi que l’abstraction de second niveau permettant d’identifier les objets. Nous voulons cependant rappeler l’importance de la carte des obstacles car elle nous permet de connaitre notre environnement.

Classes de gestion de communication vers la carte LOW LEVEL

(UART)

Classes de gestion de capteurs reliés directement à la carte HIGH LEVEL (Caméra)

Abstraction de 1er niveau

Gestion des capteurs et des actionneurs via des classes JAVA

Ordres

Données brutes des capteurs externes

Algorithme de détection de couleur

Interface graphique de debug

UART\_Communicator

Initialise la communication ainsi que les threads de réception et d’émission

UART\_Sender (Runnable)

Gère l’envoi de données au LOW LEVEL via UART

UART\_Receiver (Runnable)

Gère la réception de données du LOW LEVEL via UART

1ère couche d’abstraction

LOW LEVEL VIA UART

LOW LEVEL VIA UART

# 

## **Partie Hardware**

Dans ce rapport nous expliquons très succinctement cette partie pour plus d’informations vous pouvez aller voir directement la documentation du LowLevel fournit dans le dossier projet. Voici ci-dessous un schéma synoptique de cette partie.

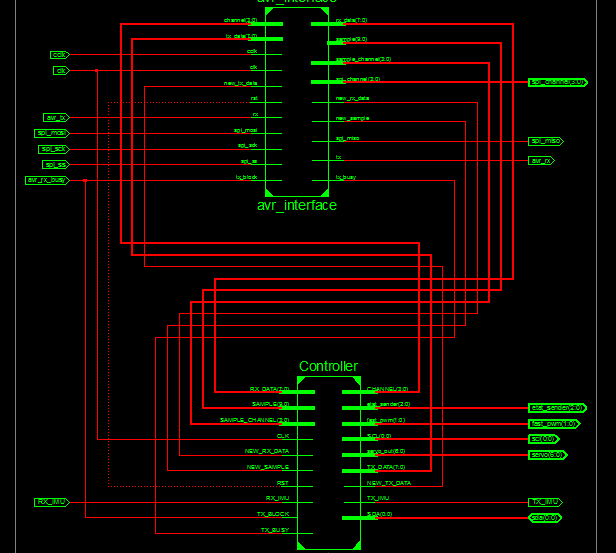
MOJO V3

AVR\_INTERFACE

* Gestion de la bas-niveau de la communication avec le HIGH LEVEL
* Gestion bas-niveau des sorties analogiques

CONTROLLER

* Gestion haut-niveau de la communication avec le HIGH LEVEL
* Gestion des capteurs et des actionneurs



Cette partie est commandée par une entitée « chef d’orchestre » que nous avons appelé controller.

Dans le schéma suivant nous expliquons le fonctionnement sous forme de diagramme d’état de cet automate.

L’exécution de cette machine à état se traduit en 2 phases : une phase d’initialisation, et une phase de communication (cf. partie communication).

Cette entité est fondamentale car c’est la seule qu’il faudra changer à tous les coups en cas d’un ajout de capteur ou d’actionneur.

Cette entité communique avec l’AVR\_Interface pour récupérer les valeurs des ADC et communiquer via UART avec le HIGH LEVEL.

Elle inclut de nombreuses IPs développée pendant le projet qui permettent l’intégration des capteurs et des actionneurs.

**CONTRÔLE**

**TX\_BUSY=1 ou NEW\_TX\_DATA=1 ou byte\_counter>0**

**TX\_BUSY=0 et NEW\_TX\_DATA=0 et byte\_counter=0**

**TX\_BUSY=0 et NEW\_TX\_DATA=0 et byte\_counter=0**

**TX\_BUSY=0 et NEW\_TX\_DATA=0 et byte\_counter=0**

**TX\_BUSY=1 ou NEW\_TX\_DATA=1 ou byte\_counter>0**

**TX\_BUSY=1 ou NEW\_TX\_DATA=1 ou byte\_counter>0**

**TX\_BUSY=0 et NEW\_TX\_DATA=0**

**TX\_BUSY=1 ou NEW\_TX\_DATA=1**

**TX\_BUSY=0 et NEW\_TX\_DATA=0**

**TX\_BUSY=1 ou NEW\_TX\_DATA=1**

RW = 0

# **Test unitaire**

Dans cette partie nous vous proposons notre cahier de recette final. Le projet ayant tardé à s’établir nous ne pouvons vous donner un cahier de recette initial. En effet certains composants ou certaines taches se sont rajouté et d’autres ont été supprimés

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COMPOSANT | PARTIE | DESCRIPTION | SOURCES TESTEES | RESULTAT |
| Interface I2C | LOW LEVEL | Visualiser une trame I2C à l'analyseur logique (émission/réception) | I2C\_master.vhd | OK |
| LIDAR | LOW LEVEL | Visualiser une communication I2C entre le fpga et le lidar (émission/réception) | lidar.vhd | OK |
| PWM | LOW LEVEL | Contrôler un servomoteur à partir du FPGA de façon statique | pwm.vhd | OK |
| Sorties analogiques | LOW LEVEL | Contrôler un servomoteur à partir du FPGA via un potentiomètre | Analog\_in.vhd | OK |
| Interface UART via AVR | LOW LEVEL | Envoyer/recevoir des données depuis un PC | Control.vhd | OK |
| Gestion IMU (UART) | LOW LEVEL | Envoyer l'ordre à l'IMU d'envoyer ses données en binaire, visualisation à l'analyseur logique | IMU.vhd | OK |
| Integration low level | LOW LEVEL | Récupérer les informations des capteurs et envoyer les ordres de bas niveau aux actionneurs via une application JAVA depuis un PC | mojo\_top.vhd | OK |
| COMPOSANT | PARTIE | DESCRIPTION | SOURCES TESTEES | RESULTAT |
| Test de la base (réparer) | BASE MECANIQUE | poser un poids sur la base et vérifier que les 3 roues du même côté restent sur le même axe | - | OK |
| Test de la modification du servomoteur | BASE MECANIQUE | Le tester à l'aide d'un microcontroleur arduino et observer la rotation continue | exemple du arduinoIDE: servo | OK |
| Test de modification du bras articulé | BASE MECANIQUE | Le faire fonctionner et vérifier qu’il ne se devise plus. | pwm.vhd ou via arduino | OK |
| Test de l’assemblage mécanique du servomoteur et du Lidar | BASE MECANIQUE | Brancher le servomoteur et vérifier que la rotation s’exerce correctement sans à-coups et sans mettre en danger le Lidar. |  | OK |
| Vérification de l’intégrité mécanique des différentes pièces 3D | BASE MECANIQUE | Sous le logiciel solidworks on utilise la fonction de calcul des interférences si le résultat est nul, la pièce est correcte. |  | OK |
| COMPOSANT | PARTIE | DESCRIPTION | SOURCES TESTEES | RESULTAT |
| Test de la reconnaissance de couleur | HIGH LEVEL | brancher la caméra et vérifier qu’elle détecte bien une balle de couleur rouge | package reconaissanceObjet | OK |
| Application JAVA de communication avec le LOW LEVEL | HIGH LEVEL | Demarrer l'application java dont le main est dans le package UART.debug, la fenêtre indique les valeurs des capteurs aisi que celle des actionneurs | packages UART.debug et UART.core | OK |

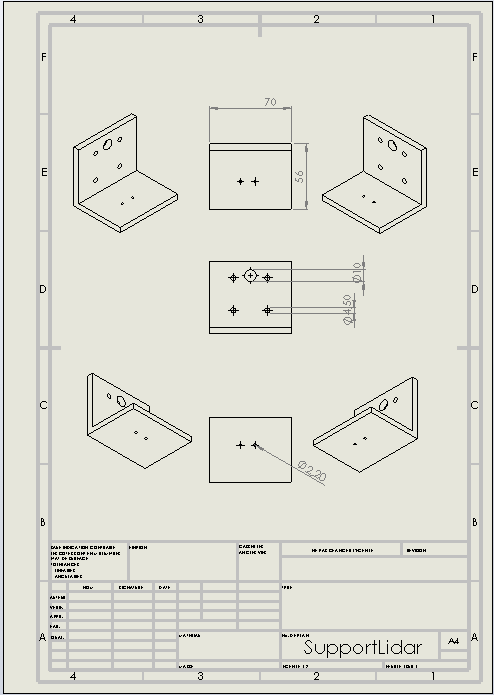
# **Reprise du projet**

Nous faisons cette partie pour les élevés reprenant le projet les années futures. Vous y trouverez uniquement une liste des composants de notre systèmes. En effet Pour toutes les parties de programmation une documentions est faites dans laquelle vous pourrez trouver comment réutiliser ou continuer les codes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom | Qte |  | prix unitaire |  | lien |
| Châssis WILD THUMPER 6WD | 1 |  | 295 |  | http://www.gotronic.fr/art-chassis-wild-thumper-12400.htm |
| Carte commande SaberTouth 2x25A | 1 |  | 120 |  | http://www.gotronic.fr/art-commande-sabertooth-2x25a-11575.htm |
| collecteur Tournant | 1 |  | 21,9 |  | http://www.francerobotique.com/connectiques/256-collecteur-tournant-avec-collerette-22-mm.html |
| Lidar-Lite 2 | 1 |  | 114,89 |  | http://www.robotshop.com/en/lidar-lite-2-laser-rangefinder-pulsedlight.html |
| bras Robotique DAGU | 1 |  | 220,74 |  | http://www.robotshop.com/eu/fr/bras-robotique-dagu.html |
| FireFly ARM RK3288 | 1 |  | 129 |  |  |
| Camera Moebius 1080p | 1 |  | 84,9 |  | http://www.studiosport.fr/camera-miniature-mobius-1080p-p-3085.html |
| Batterie LIPO | 1 |  | 52,99 |  | http://www.conrad.fr/ce/fr/product/254638/Accu-LiPo-HACKER-111-V-4500-mAh-34500331;jsessionid=782ED8000A6A88D3C7A6B8DA1479B468.ASTPCEN19?ref=list |
| carte MOJO V3 | 1 |  | 74,95 |  | https://www.sparkfun.com/products/11953 |
| Capteur de distance | 3 |  | 12 |  | http://www.gotronic.fr/art-capteur-de-mesure-sharp-gp2y0a021yk-11539.htm |
| Support capteur de distance | 3 |  | 6,2 |  | http://www.gotronic.fr/art-support-pour-capteurs-11500.htm |

# **Liste des figures**

# **Annexe 2 : Plan support du Lidar**



# **Annexe3 : Plan du support servomoteur**

