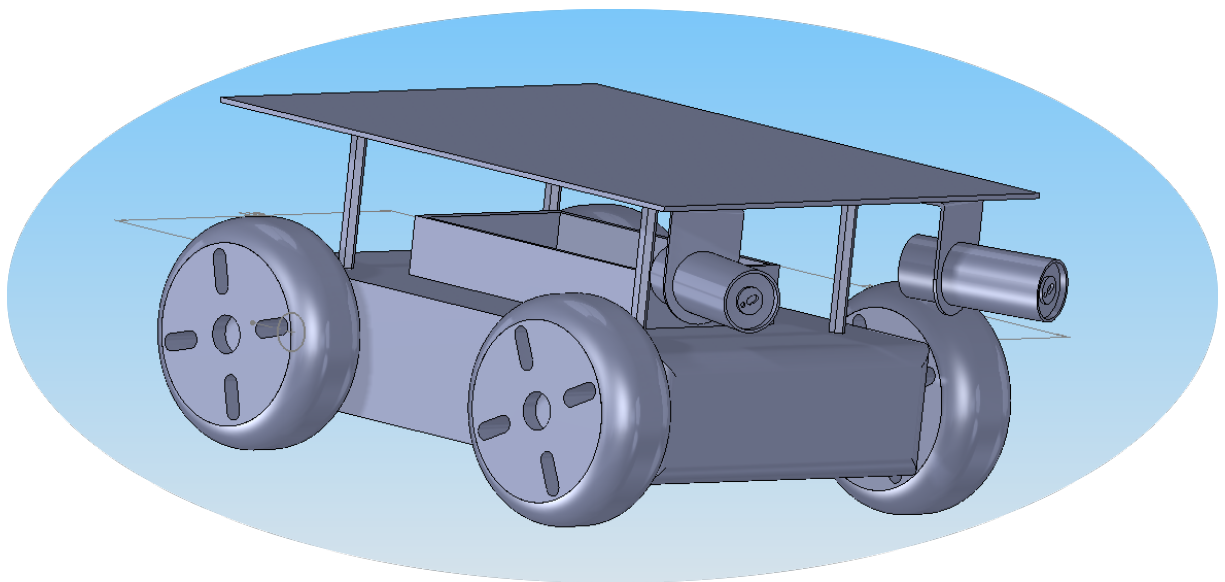


El ABDI EL ALAOUI Soumaya
HARLÉ Pierre-Antoine

Rapport projet industriel

Véhicule mobile interfacé ROS

Véhicule autonome



Période projet : du 31/01/2017 au 10/04/2017
Tuteur technique & client : M. LENGAGNE
Tutrice industrielle : Mme. GOI

Résumé

Notre projet a pour objectif de produire un petit véhicule mobile interfacé avec Robot Operating System(ROS). Ce qui permettra de mettre en place une application pour les travaux pratiques en Génie Electrique à Polytech Clermont-Ferrand pour les futurs étudiants du département en Génie Electrique.

L'objectif étant d'obtenir une application fonctionnelle et surtout que cette application soit ludique et une "prise en main" facile pour les élèves de Génie Electrique, puisque le département génie électrique de Polytech Clermont-Ferrand a la possibilité dans les années avenir de se spécialiser dans le domaine de la robotique.

Nous avons réalisé plusieurs briques à l'intérieur de notre projet, ce qui nous a permis d'avancer pas à pas afin d'obtenir des résultats concrets dans temps imparti pour ce projet et ceci malgré les difficultés rencontrées.

Abstract

Our project aims to produce a small mobile vehicle interfaced with Robot Operating System. This will allow setting up an application for practical work in Electrical Engineering at Polytech Clermont-Ferrand for future students of the department.

The objective is to obtain an application and above all that this application is easy to use for the students of Electrical Engineering, since the electrical engineering department of Polytech Clermont-Ferrand has the possibility of Specializing in the field of robotics in the future years.

We realized several bricks within our project, which allowed us to progress step by step and to obtain concrete results despite the lack of time and despite the difficulties encountered.

Remerciements

Nous tenons à remercier notre client et notre tuteur technique, le professeur M. LENGAGNE, pour son soutien tout au long de ce travail. Nous voudrions exprimer notre gratitude pour l'aide, la confiance et l'encouragement qu'il nous a apporté durant cette période de projet.

Nous tenons également à remercier notre tutrice industrielle Mme. GOI, pour son aide et ses conseils sur la gestion de projet. Ce qui est essentiel pour notre avenir en tant qu'ingénieur.

Nous sommes reconnaissants de l'aide que nous a apporté M. LAFFONT, responsable des projets industriels, sur les domaines techniques ainsi que sur la conduite de projet (micro management).

Finalement nous souhaitons remercier M. SANCHEZ, technicien en génie électrique, pour sa disponibilité ainsi que sur son aide sur des points techniques.

Table des matières

Introduction	6
1. Problématique du projet	7
1.1 Cahier des charges	7
1.2 Schéma structurel.....	7
1.3 Gestion de projet	8
1.3.2 Gantt.....	8
2. Recherches spécifiques sur le projet	9
2.1 Méthodes choisie pour réaliser les différentes tâches.....	9
2.2 Adaptation de résultats du projet ROGHEC	9
3. Electronique	10
3.1 Choix des composants (coût, caractéristiques).....	10
3.2 Différentes problématiques de commandes et d'isolation	11
3.3 Réalisation de la carte électronique.....	11
3.3.1 Schéma électriques sous Proteus	11
3.3.2 Routage de la carte avec règles de CEM et CAO sous Proteus.....	13
3.3.3 Soudage et test de la carte électronique	13
4. Programmation	14
4.1.1 Ros	14
4.1.1.1 Communication entre deux ordinateurs (à l'aide de la webcam)	14
4.1.1.2 Communication entre Raspberry et l'ordinateur (wifi)	16
4.1.2 Pic	17
4.1.2.1 Liaison USB.....	17
4.1.2.2 Commande des moteurs	18
Conclusion.....	19
Bilan personnel	20
Bibliographie	21

Introduction

La voiture autonome de Google correspond bien à ce projet avec quelques différences ; comme la complexité qui est moins élevée et la miniaturisation du prototype.

Nous effectuons ce projet dans le cadre de notre cursus de quatrième année en Génie Electrique en école d'ingénieur. En effet il s'agit d'un projet industriel avec des problématiques de « demain ».

En collaboration avec notre Client ; le département Génie Electrique, ayant conscience des enjeux du domaine de la voiture autonome ; nous étudions et mettons en place plusieurs techniques afin de pouvoir mettre en œuvre les connaissances théoriques et pratiques acquises durant notre formation.

Ce projet nous a permis d'acquérir de nouvelles connaissances dans certains domaines, comme l'informatique, la programmation, ainsi que dans l'électronique et la robotique. C'est le mélange de tous ces domaines qui nous permettrons d'aboutir au résultat attendu.

Pour cela nous avons mis en place différentes méthodes de travail, sur la gestion de projet (GANTT, WBS, Schéma) ainsi que dans le domaine technique.

Dans un premier temps nous allons aborder la problématique de la mise en place de différentes briques afin de réaliser le projet, puis nous allons vous présenter la partie électronique et finalement la partie informatique.

1. Problématique du projet

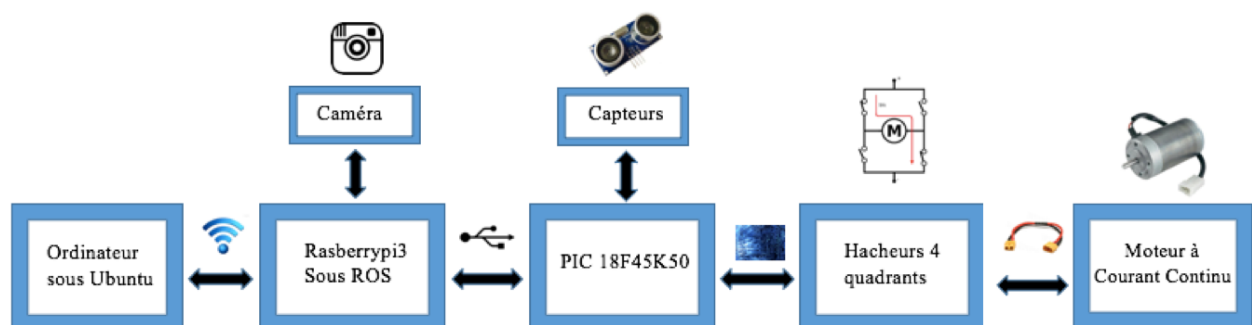
1.1 Cahier des charges

Le cahier des charges est un document qui permet de connaître l'attente du client, il y a une description précise des besoins du client. Nous avons mis en place notre cahier des charges avec la coopération de notre client M. LENGAGNE. Ainsi, après négociation avec notre client nous avons abouti au résultat ci-dessous.

- Réaliser un véhicule mobile interfacé avec Robot Operating System (ROS).
- Utilisation en travaux pratiques en Génie Electrique à Polytech Clermont-Ferrand.
- Il disposera de 3 capteurs pour se diriger dans l'espace.
- Contrôler le véhicule en utilisant les informations des capteurs récupérés.
- Pour assurer la mobilité du véhicule il disposera de deux moteurs à courant continue.

1.2 Schéma structurel

Dans n'importe quel projet il est important de mettre en place certaines méthodes de travail afin d'organiser et de mettre en place les différentes étapes de ce projet. C'est pour cela que nous avons ce schéma structurel.

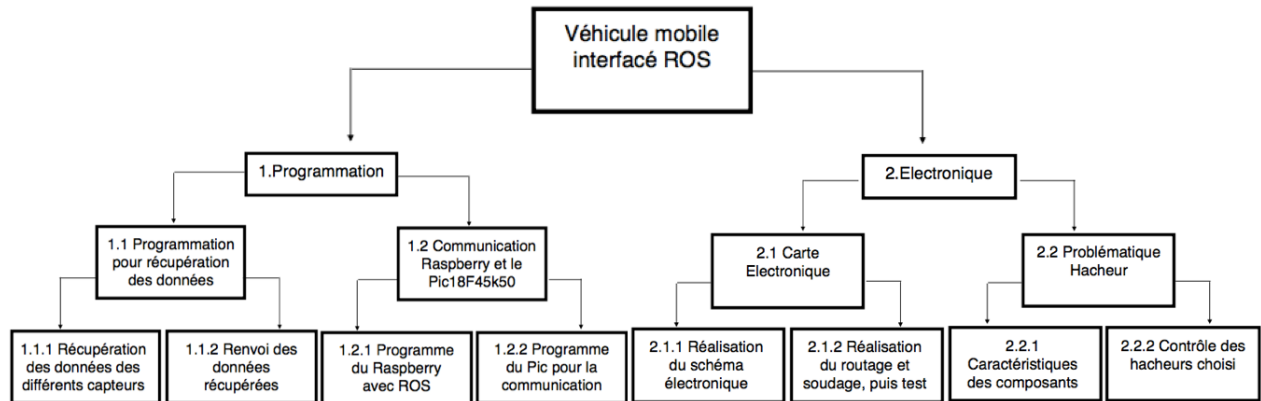


Nous pouvons voir sur notre schéma structurel les différentes étapes à mettre en place pour aboutir au résultat final demandé dans le cahier des charges par notre client. Ce schéma permet de concrétiser les différentes idées de notre projet. Ce qui permet d'être sûr de la bonne compréhension de la demande du client. Puis de voir les solutions possibles à mettre en place.

1.3 Gestion de projet

1.3.1 Work Breakdown Structure(WBS)

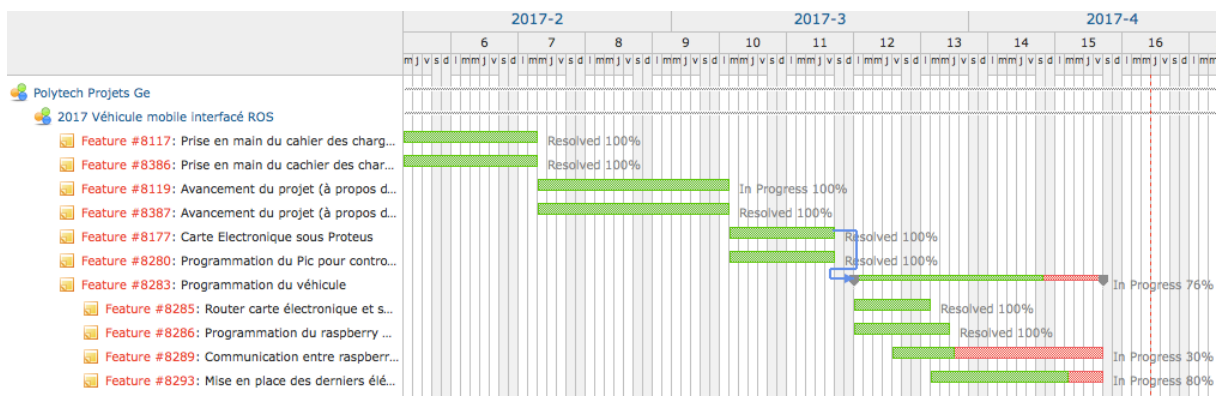
Le WBS permet de diviser notre projet en différentes tâches, qui sont divisées en sous tâches sans inclure de notion de temps.



Dans notre WBS nous pouvons remarquer qu'il y a deux principales parties dans notre projet, une partie programmation, ainsi qu'une partie électronique. Ceci nous a permis de répartir notre projet sous forme de sous tâches et pouvoir avancer pas à pas. Le WBS est un outil très important dans la gestion de projet. Il nous permet de répartir les tâches pour les différents collaborateurs faisant parti de ce projet. Ce qui permet une plus grande efficacité dans l'avancement du projet.

1.3.2 Gantt

Le diagramme de Gantt nous permet d'introduire une notion de temps dans l'organisation du projet. C'est un complément du WBS, cela nous permet de regarder et de prévoir le nombre d'heures nécessaires pour réaliser une tâche précise.



Nous pouvons constater sur notre Gantt que nous ne sommes pas arrivés au bout de notre projet. Cela est dû au problème rencontré durant la tâche de la réalisation du schéma électrique, routage, soudage et tests de la carte électronique (cela à cause d'un problème de versionnage de nos prédécesseurs).

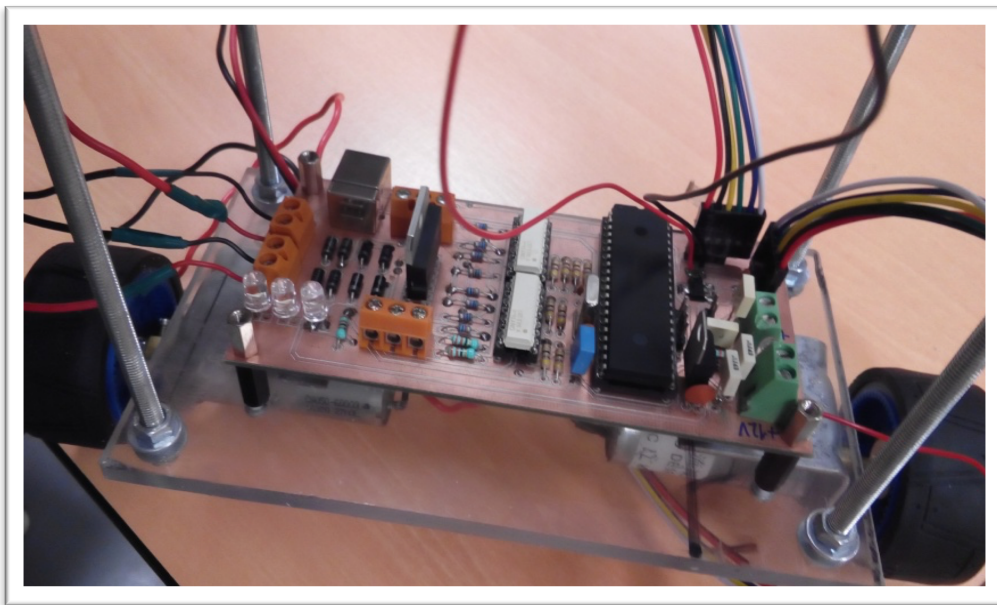
2. Recherches spécifiques sur le projet

2.1 Méthodes choisies pour réaliser les différentes tâches

En s'inspirant du schéma fonctionnel réalisé nous avons pu choisir les méthodes suivantes :

- Récupération des données des capteurs à l'aide d'un microcontrôleur
- Commande des moteurs à courant continu grâce aux ponts en H (Hacheurs)
- Utilisation d'un Raspberry Pi pour l'implantation du système d'exploitation ROS
- Communication en liaison USB entre le microcontrôleur et le Raspberry Pi

2.2 Adaptation de résultats du projet ROGHEC



Après avoir réfléchi sur le cahier des charges et avoir effectué différentes recherches pour mettre en place les différentes tâches à réaliser nous nous sommes référés à un ancien projet qui lui a été achevé en 2015 par des anciens étudiants de quatrième et cinquième années de Génie Electrique.

Ce projet s'appelle ROGHEC, il est constitué de deux parties. Une partie avec la commande de moteurs à courant continu et une partie avec la commande de deux moteurs à courant continu.

Nous l'avons analysé et comparé ; regardé les similitudes et les différences avec notre projet ce qui nous a permis de confirmer ou au contraire démentir certaines de nos idées.

3. Electronique

3.1 Choix des composants (coût, caractéristiques)

Composants	Référence	Prix
Optocoupleur	CNY_74-4	1,23 €
Double Pont en H	L298	3,48 €
Microcontrôleur	PIC 18F45K50	5,97 €
Vingt résistances	XXX	1,80 €
Six condensateurs	XXX	0,42 €
USB	XXX	0,657 €
Six borniers	XXX	1,26 €
Plaque époxy	XXX	4,76 €

TOTAL	19,57 €
--------------	---------

Donc après l'étude de la carte électronique et le choix des composants nous savons qu'il nous faut 19,57 € juste pour réaliser la carte électronique.

Ayant la chance et la possibilité de pouvoir utiliser le matériel du département Génie Electrique de Polytech Clermont-Ferrand nous n'avons pas besoin d'acheter d'autres composants ou d'autres matériaux pour notre projet.

Si nous n'avions pas cette possibilité, pour effectuer ce projet il nous faudrait un budget de :

- 20 € pour la carte électronique
- 30 € pour la carte Raspberry Pi
- 5 € pour les deux moteurs MCC
- 3 € pour le capteur ultrasons
- 20 € pour le capteur gyroscopique
- 20 € pour la caméra USB
- 10 € pour la batterie 5V
- 15 € pour la batterie 12V

Total de l'application : 123 € TTC

3.2 Différentes problématiques de commandes et d'isolation

Pour la commande des moteurs à courant continu, nous avons choisi la solution d'utiliser des hacheurs quatre quadrants qui ont aussi le nom de pont en H. Cela permet de diriger le sens des moteurs, ainsi que leurs vitesses.

Nous pouvons voir leurs principales caractéristiques du double pont en H (L298) ci-dessous :

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive ($t = 100\mu s$) – Repetitive (80% on –20% off; $t_{on} = 10ms$) – DC Operation	3 2.5 2	A A A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

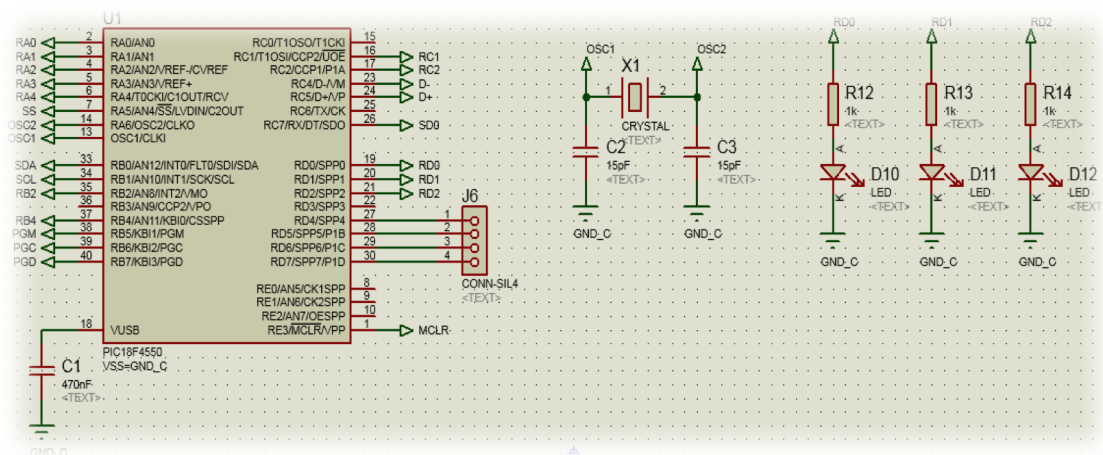
Pour l'isolation galvanique (protection), c'est-à-dire l'isolation entre la partie commande et la partie puissance. Nous avons choisi des optocoupleurs. Ce qui nous permettra de faire une isolation galvanique avec le moins de pertes possibles.

Nous pouvons voir leurs principales caractéristiques des optocoupleurs (CNY74-4) ci-dessous :

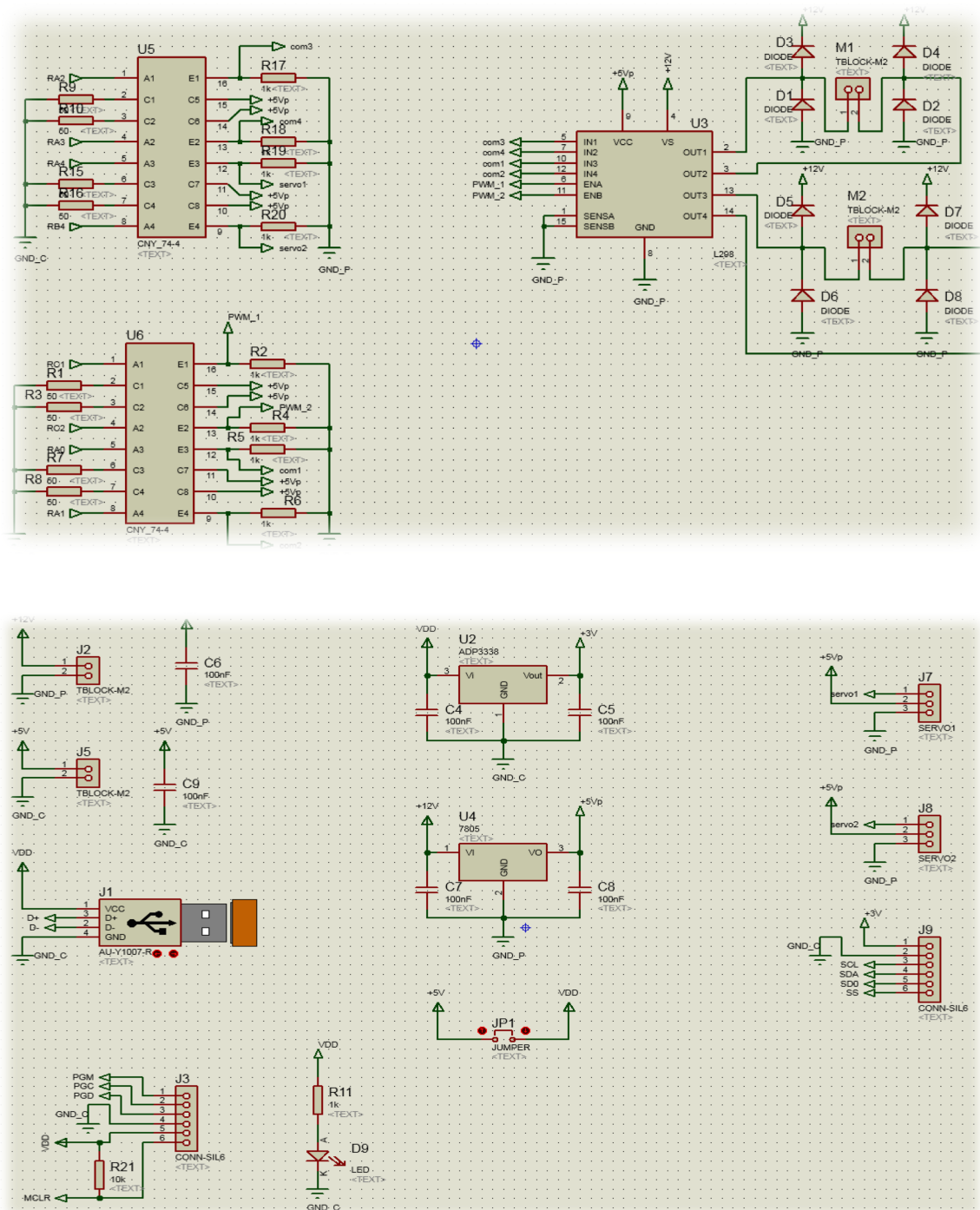
PARAMETER	TEST CONDITION	PART	SYMBOL	VALUE	UNIT
Peak reverse voltage			V_R	3	V
Forward continuous current			I_F	60	mA
Power dissipation			P_{diss}	100	mW
Derate linearly from 55 %				1.33	mW/ $^\circ C$

3.3 Réalisation de la carte électronique

3.3.1 Schéma électriques sous Proteus

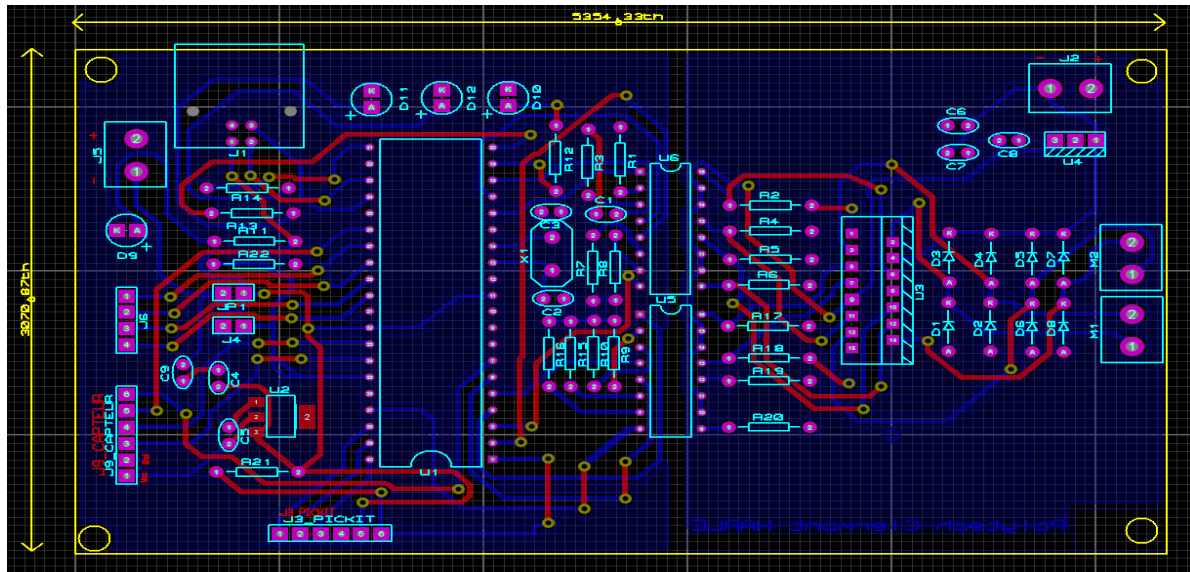


Véhicule Interfacé ROS – GE4A – P17AB10



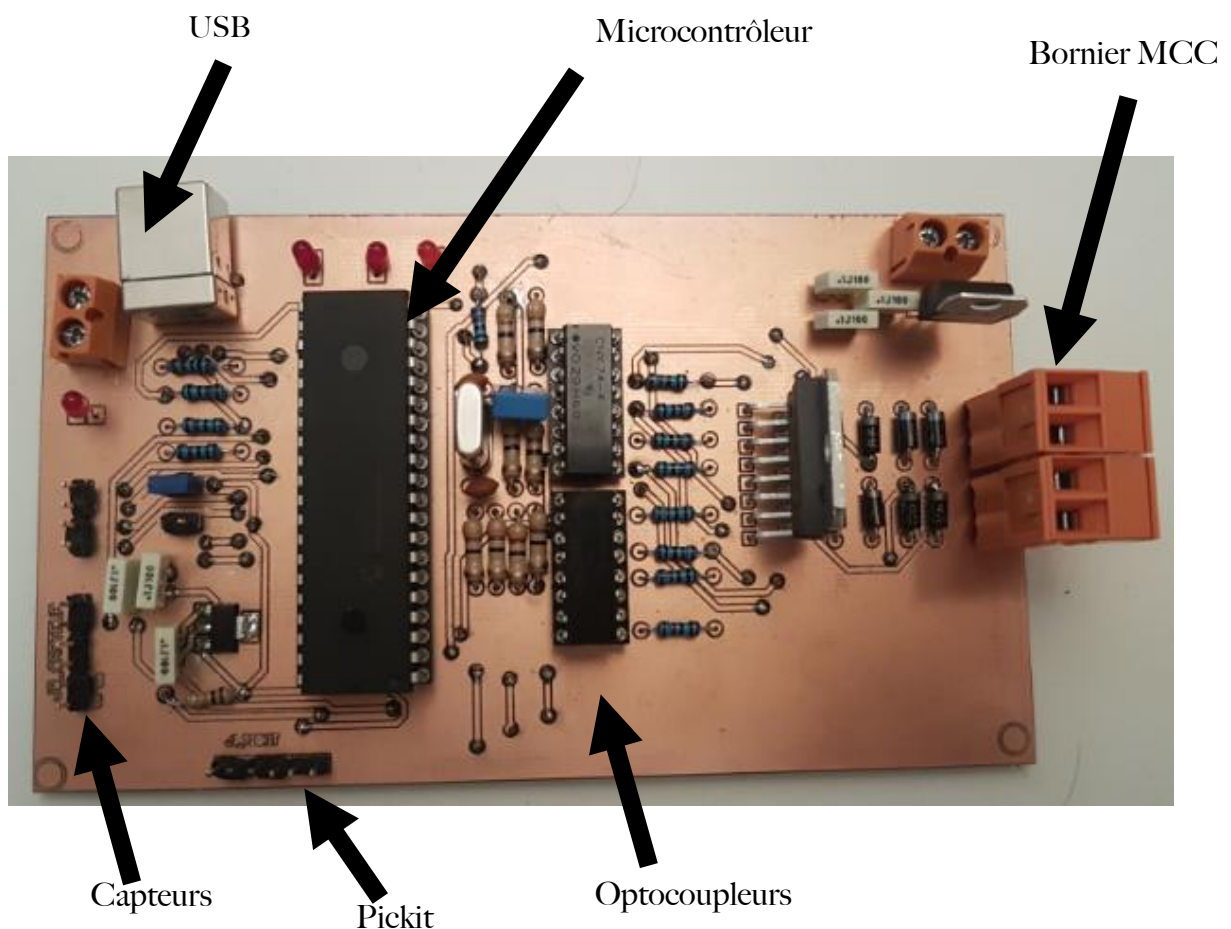
Nous pouvons remarquer sur ces schémas électriques ci-dessus que nous retrouvons bien les différents éléments nécessaires au cahier des charges. C'est la première dans la réalisation de la carte électronique. Sur ce schéma électrique nous pouvons voir, l'USB, les régulateurs, les différents borniers, le microcontrôleur, les optocoupleurs, le double pont en H. Nous avons créé trois pages de travail, ceci pour rendre notre travail plus propre, mieux organisé.

3.3.2 Routage de la carte avec règles de CEM et CAO sous Proteus



Nous pouvons voir ci-dessus le routage de la carte électronique sous **Proteus** un logiciel permettant la Conception Assisté par Ordinateur électronique (CAO). Nous pouvons remarquer que ce routage peut être optimisé en terme de via et de taille.

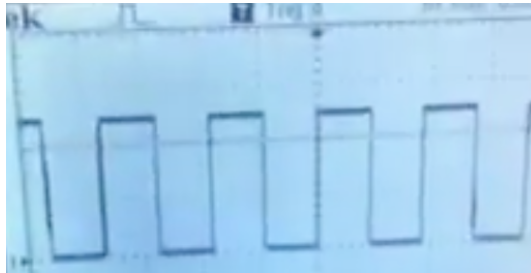
3.3.3 Soudage et test de la carte électronique



Nous avons effectué plusieurs tests pour observer les parties de la carte électronique qui sont fonctionnelles.

- Dans un premier temps nous avons effectué un programme dans MplabX qui permet d'allumer les LEDs tests. Ce test est fonctionnel, ce qui signifie que la première partie de la carte fonctionne correctement. C'est-à-dire l'alimentation, le régulateur, l'USB ainsi que le microcontrôleur PIC18F45K50.
- Puis dans un deuxième temps nous avons mis le programme de la commande des moteurs, c'est-à-dire le programme avec les PWM qui nous permettent de contrôler les moteurs à courant continus. Nous avons regardé le signal en sortie du microcontrôleur et à la sortie de l'optocoupleurs et nous retrouvons bien le même signal, ce qui signifie que cette deuxième partie de la carte électronique et elle aussi fonctionnelle.

Signal observé à l'oscilloscope :



Donc nous avons des résultats cohérents jusqu'à la sortie de l'optocoupleur, mais en sortie du double pont en H nous n'avons plus aucun signal.

4. Programmation

4.1.1 Ros

4.1.1.1 Communication entre deux ordinateurs (à l'aide de la webcam)

On peut faire fonctionner une webcam en liaison USB sous le système d'exploitation ROS.

Voici ci-dessous les différentes étapes pour réaliser ce travail :

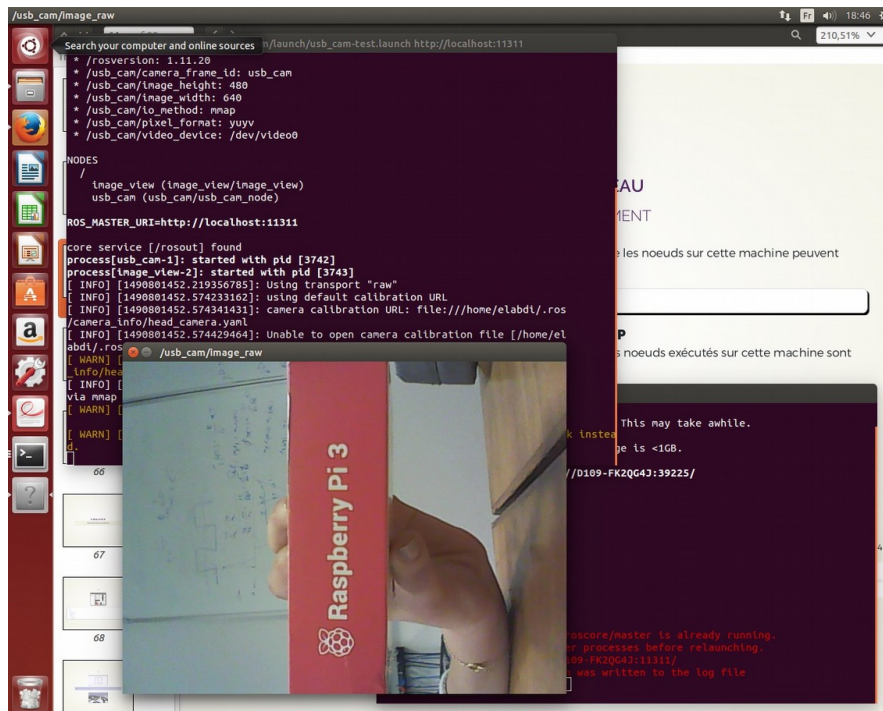
- Etape 1 : Tout d'abord il faut brancher la webcam USB
- Etape 2 : Installer et configurer l'environnement pour utiliser ROS

source /opt/ros/indigo/setup.bash

- Etape 3 : Lancer un ***roscore*** pour que les nœuds puissent communiquer

- Etape 4 : Pour pouvoir utiliser la webcam sous ROS

roslaunch / opt/ ros/indigo/share/usb_cam/launch/usbcam-test.launch



- Etape 5 : Pour récupérer l'adresse IP de l'ordinateur il suffit de lancer

ifconfig

- Etapes 6 : Pour pouvoir communiquer deux ordinateurs entre eux il faut d'abord lancer

sudo gedit/etc/hosts

Ensuite il faut ajouter l'adresse IP du deuxième ordinateur avec la commande :

export D109-DXXXXXXX 172.25.XX.XXX

Le nom de l'ordinateur

L'adresse IP de l'ordinateur principale secondaire (host)

Sur l'ordinateur host il faut refaire l'étape 2 et 3

Puis lancer : ***export ROS_Master_URI=http : //172.27.XX.XX::11311/*** Pour identifier le master

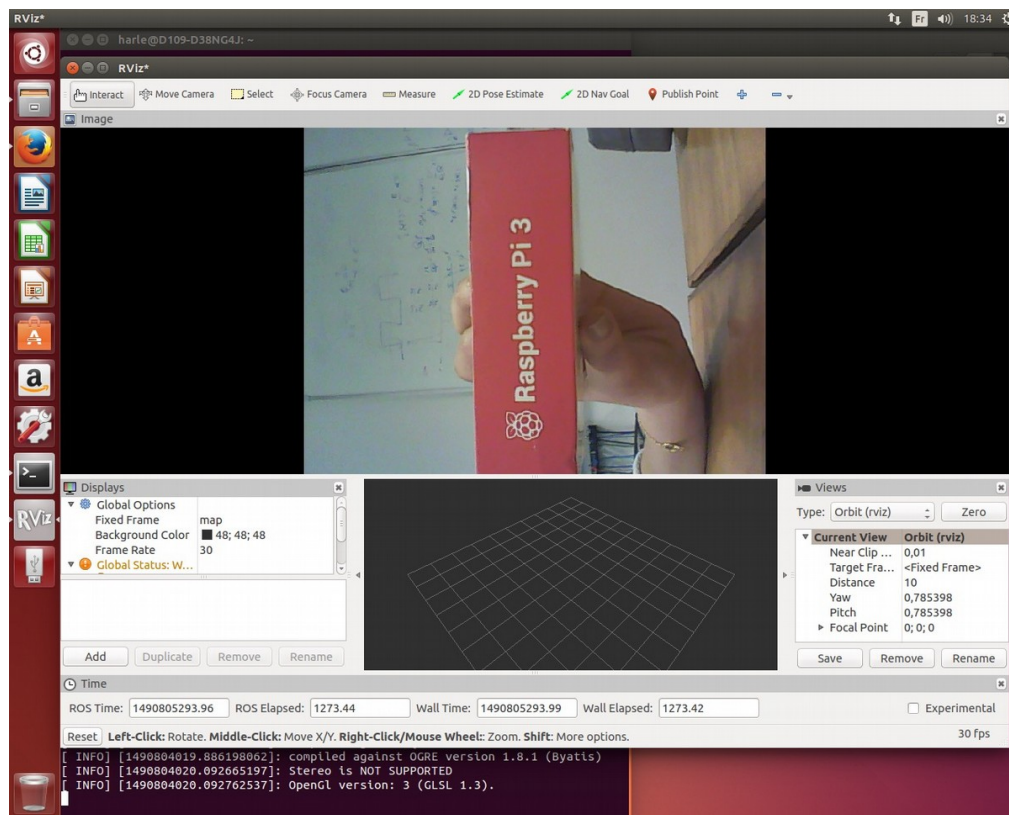
L'adresse IP de l'ordinateur principal

export ROS_HOSTNAME=172.27.XX.XX

L'adresse IP de l'ordinateur principal

export LIBGL_ALWAYS_SOFTWARE=1

Finalement on lance *rviz* pour visualiser l'image récupérée de l'ordinateur principal



Si vous avez besoin d'informations complémentaires n'hésitez pas à vous référer aux sites suivants :

<http://www.ros.org>

http://wiki.ros.org/usb_cam

<http://wiki.ros.org/fr/ROS/Tutorials>

4.1.1.2 Communication entre Raspberry et l'ordinateur (wifi)

Avec la **Raspberry-Pi 3** le Wi-Fi est désormais nativement intégré et il est maintenant possible de se connecter à un réseau, il suffit de brancher le câble Ethernet et son alimentation.

Dès que la Raspberry-Pi est connecter sur internet, on lance en premier temps la commande :

nmap sur le terminal pour pouvoir voir les appareils connectés (leurs adresses IP est affichées).

Parmi ces appareils on trouve bien la Raspberry-Pi avec son adresse IP à côté.

Ensuite on entre les commandes suivantes :

nmap -v -sn 192.168.X.X/24



Adresse IP du routeur

ssh -Y pi@192.168.X.X



Adresse IP du Raspberry-Pi

Après avoir effectué ces commandes nous sommes maintenant connecté à la carte Raspberry-Pi et l'on peut également utiliser toutes les commandes de ROS.

Remarque :

Dans notre projet, on a utilisé une Raspberry-Pi dont le système d'exploitation ROS a été déjà installé précédemment. Si ce n'était pas le cas il faut l'installer d'abord voici quelques sites dont vous pouvez vous référer :

<https://raspbian-france.fr>

<http://wiki.ros.org/ROSberryPi>

<http://www.instructables.com/id/Raspberry-Pi-and-ROS-Robotic-Operating-System>

4.1.2 Pic

4.1.2.1 Liaison USB

Il y a diverses étapes pour mettre en place la liaison USB sur le microcontrôleur, qui sont les suivantes :

- Driver
- Programme contenant les descripteurs
- Énumération entre le driver et programme
- Le fichier USB config

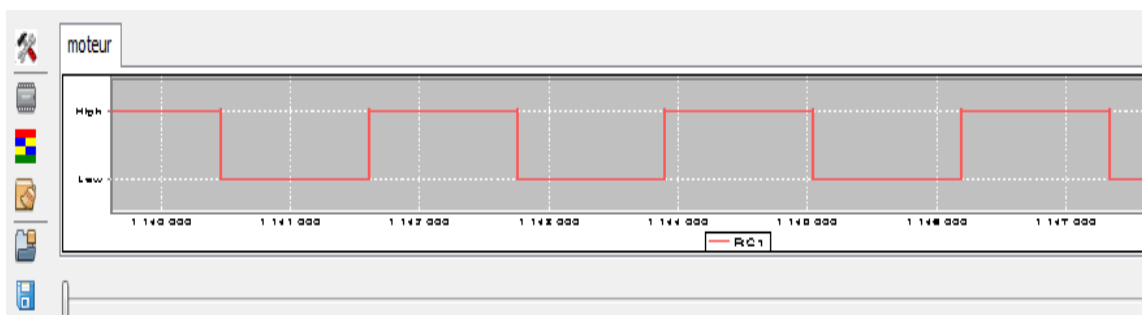
Malheureusement à cause du problème rencontré sur la carte électronique on n'a pas pu établir la liaison USB sur le microcontrôleur.

4.1.2.2 Commande des moteurs

```
76 }
77
78 void init_PWM(void)
79 {
80     CCP1CON=0b00111100;
81     CCP2CON=0b00111100;
82     CCPR1L=0;
83     CCPR2L=0;
84     //T2CONbits.TMR2ON=1;
85 }
86
87 void duty_cycle_PWM1(float duty)
88 {
89     CCPR1L=(int) (duty*PR2);
90 }
91 void duty_cycle_PWM2(float duty)
92 {
93     CCPR2L=(int) (duty*PR2);
94 }
95 }
```

Ce programme nous permet de contrôler les moteurs à courants continus, il a été effectué sur le logiciel MPLABX, C'est un logiciel qui est fourni gratuitement par l'entreprise Microchip, ce qui nous permet de programmer notre microcontrôleur.

Après avoir simulé le programme précédent on arrive à visualiser un signal PWM qui consiste à son tour d'alimenter le moteur avec une tension en créneaux. La tension moyenne dépend alors du rapport cyclique ainsi que de la vitesse.



Conclusion

Durant ce projet industriel en quatrième année de Génie Electrique à Polytech Clermont-Ferrand, nous avons réussi à nous séparer le travail à effectuer, ce qui nous a permis d'obtenir de nombreux résultats sur les différents sous domaines.

Il y a eu de l'auto-formation durant ce projet, dans le domaine de la programmation sur le nouveau système d'exploitation conçu pour la robotique ROS en plus de la formation reçue par M. LENGAGNE.en début de projet.

Nous avons réalisé différentes briques, la réalisation de la carte électronique, la communication avec le Raspberry Pi qui lui communiquera avec l'ordinateur sous Ubuntu, qui nous permettra de diriger notre véhicule.

Bilan personnel

Ce projet fût très bénéfique pour nous, il nous a permis d'apprendre de nos erreurs, ou au contraire, confirmer notre savoir-faire.

Nous n'avons pas réussi à finir notre projet dans le temps imparti, mais cela nous a permis d'acquérir de nouvelles connaissances aussi bien en technique qu'en gestion de projet.

Nous avons pensé qu'il pourrait y avoir des perspectives d'évolution sur ce projet, c'est-à-dire finir les briques manquantes, puis les assembler, ainsi que les optimiser (comme la miniaturisation de la carte électronique).

Bibliographie

<https://forge.clermont-universite.fr/projects/p17ab10?jump=welcome>

https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf

<http://www.vishay.com/docs/83526/83526.pdf>

<http://www.ros.org>

http://wiki.ros.org/usb_cam

<http://wiki.ros.org/fr/ROS/Tutorials>

<https://raspbrian-france.fr>

<http://wiki.ros.org/ROSberryPi>

<http://www.instructables.com/id/Raspberry-Pi-and-ROS-Robotic-Operating-System>