

COURSE DE VOITURES AUTONOMES

2020-2021



EMILE MARTINCIC

Groupe D

Créé par : Joséphine LY

Karl-Emmanuel MONGBO

université
PARIS-SACLAY
IUT DE CACHAN



REMERCIEMENTS

Nous tenons tout d'abord à remercier l'ensemble de la Direction de l'IUT de Cachan, qui nous ont permis d'intégrer l'établissement au mois de Septembre 2019.

Nous remercions notre enseignant M. MARTINCIC, qui a su nous épauler lors de ce projet, et également M. BELLANGER qui a pu répondre à nos différentes questions.

Ensuite, nous adressons nos vifs remerciements à tout le personnel de l'IUT, pour nous avoir fourni le matériel dont nous avons besoin pour ce projet.

Nos remerciements vont enfin au corps professoral et administratif de l'IUT de Cachan pour nous avoir donné toutes les connaissances nécessaires durant ce projet.

RÉSUMÉ

Depuis quelques années, nous connaissons l'avènement de la direction assistée ainsi que plusieurs aides à la conduite. Aujourd'hui, l'ère est à la voiture autonome. C'est dans cette optique que des professeurs de l'ENS Cachan ont eu l'idée d'une course de voiture autonome. Le but étant qu'à partir d'une voiture, d'un LIDAR et d'une carte programmable nous puissions construire une voiture livrée à elle-même sur un circuit. Que cette dernière puisse trouver toute seule son chemin, éviter des obstacles et être rapide.

Tels des constructeurs, nous avons les premiers jours assemblé la voiture, installé le moteur et le servomoteur. Une fois la carrosserie faite, nous devons nous interroger sur le branchement du moteur, du servomoteur, du variateur de vitesse, la gestion de la puissance, la communication avec une centrale via un module Bluetooth mais surtout, un programme de base pour faire avancer, reculer ou donner une direction à la voiture. Les tâches ont été divisées, réparties et nous étions lancés.

La batterie fournissant du 7,2V, un transformateur était indispensable pour alimenter la carte Keil, le LIDAR et le servo moteurs alimentés en 5V. Après avoir testé plusieurs transformateurs, notre professeur nous a fourni une carte qui transforme n'importe quelle tension supérieure à 6,8V en 5V. Nous avons conçu une carte d'interface qui permettait de relier les tensions de 5V et les masses entre elles, ainsi que de faire le lien entre la carte Keil que nous programmions et les différents composants qui doivent être reliés.

Au niveau de la programmation, le variateur de vitesse et le servo-moteur doivent recevoir une PWM (Modulation à Largeur d'Impulsion) qui va consigner la vitesse pour le variateur et l'angle de direction des roues avant pour le servomoteur. Le LIDAR et le module Bluetooth communiquent via une « liaison série » : notion que nous ne maîtrisons pas. Il a donc fallu apprendre à communiquer avec les liaisons séries afin d'envoyer des chaînes de caractères permettant de communiquer avec le LIDAR et la centrale via le Bluetooth.

Une fois tous ces éléments développés séparément, nous devons les assembler pour tester le fonctionnement de la voiture. Nous avons donc soudé certains composants, fait les différents branchements et alimenter le tout.

Nous avons réussi à créer une voiture qui roule à partir des informations que nous lui donnons. Il ne reste plus qu'à étudier le LIDAR et les informations qu'il fournit pour rendre la voiture autonome.

Nous étions une équipe de deux pour un projet qui pouvait en demander plus. Cependant, avec une bonne organisation, de l'entraide et le soutien de notre professeur, nous ne sommes plus très loin d'avoir une voiture autonome.

Mots clés : Voiture autonome, LIDAR, Module Bluetooth, Batterie, Transformateur, Carte interface

SOMMAIRE

REMERCIEMENTS	3
RÉSUMÉ	4
INTRODUCTION.....	6
I. Présentation du défi.....	7
1. Le matériel fourni.....	7
2. Le règlement	9
II. Études et présentation du matériel.....	10
1. Propulsion et direction du véhicule.....	10
2. Adaptation des différents éléments	11
3. Mesure de vitesse	12
III. Conception et réalisation électronique	13
1. Le capteur optique de mesure de vitesse.....	13
2. La carte interface	15
IV. Informatique.....	19
1. Communication entre le microcontrôleur et les composants.....	19
2. Algorithme de conduite autonome	22
CONCLUSION	24
LISTE DES FIGURES.....	25
ANNEXE	25

INTRODUCTION

L'apparition des véhicules autonomes est un thème majeur de l'informatique embarquée ces dernières années. Taxis, camions, bateaux, drones, avions et même course automobile, tous les secteurs du transport sont concernés. Des premiers véhicules roulent seuls, en environnement connu ou avec un conducteur superviseur.

Une voiture dite autonome est équipée d'un système de pilotage automatique qui lui permet de circuler sans intervention humaine dans des conditions de circulation réelles.

Pour cela, le véhicule est équipé d'une série de capteurs laser (LIDAR), de radars et de caméras qui lui servent à modéliser son environnement en trois dimensions et à identifier les éléments qui le composent : marquage au sol, signalisation, bâtiments, véhicules, piétons. Afin qu'il puisse se guider en respectant les règles de circulation et éviter les obstacles.

A cela s'ajoutent des calculateurs pour le traitement des images, la fusion des données et les calculs de trajectoires.

L'ensemble de ces informations est traité par un programme d'intelligence artificielle qui décide des manœuvres à effectuer en agissant sur des servocommandes contrôlant les principales fonctions de la voiture : direction, freinage, accélération.

C'est dans ce cadre que l'ENS Paris Saclay a proposé aux étudiants en deuxième années de DUT Génie Électrique Informatique et Industrielle de participer à une course de voitures (réduites) autonomes. Le projet a commencé en octobre 2018 et s'est terminé par les courses en juin 2019. Le présent rapport vise à relater cette expérience, les choix faits par les élèves Joséphine LY, Karl-Emmanuel MONGBO et l'enseignant. Leur analyse permettra à d'autres de bénéficier de cette expérience pour entamer un projet similaire. Une nouvelle course est notamment prévue à l'ENS Paris Saclay en avril 2021.

I. Présentation du défi

1. Le matériel fourni

En débutant ce projet nous savions que nous étions les premiers de l'IUT de Cachan à le réaliser, il n'y avait donc aucun rapport des étudiants des années précédentes pour nous guider, nous étions donc en totale autonomie et motivés pour l'entamer. Au départ nous a été fourni un châssis Tamiya TT-02, ce châssis est l'entrée de gamme pour les amateurs de course de véhicule radiocommandé mais cela ne va pas être utilisé puisque nous effectuons une voiture autonome.

Dans le kit fourni, nous y trouvons un moteur DC 540, ce moteur à courant continu se voit fonctionner à des vitesses plutôt basses. C'est alors que l'on s'est rendu compte qu'il manquait un variateur de vitesse, notre professeur nous en a alors fournit un. Lors des premières séances nous avons construit le châssis et placé le moteur ainsi que le servomoteur.

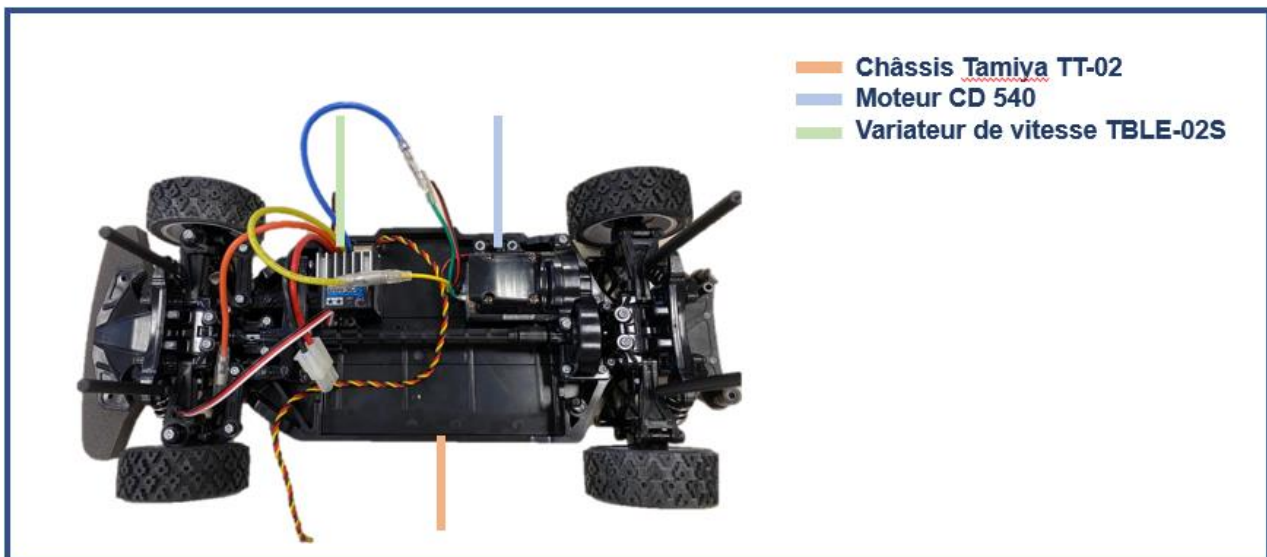


Figure 1: Le châssis construit

Notre professeur référent nous a fourni plusieurs éléments pour le bon fonctionnement du véhicule, plusieurs batteries NiMH nous ont été fournies. Avant cela nous utilisions une alimentation afin d'envoyer une tension au véhicule pour effectuer des tests tel que la programmation de la direction du véhicule.

Ensuite une carte STM 32 nous a été donnée, nous avons essayé de l'utiliser mais malheureusement cette carte était inconnue pour nous. Notre choix s'est porté sur la carte LPC1768 que nous étudions depuis un certain temps, elle n'avait alors aucun secret pour nous.

Concernant la commande à distance du véhicule, un module Bluetooth RN42XVP-I/RM a été placé dans une carte interface que nous avons conçu, ce module a pour but d'effectuer les départs et les arrêts à distance du véhicule.

De plus un LIDAR RPLIDAR Slamtec A2M8 nous a été fourni, il permet de mesurer des distances en analysant les propriétés d'un faisceau lumineux qui est réfléchi vers lui-même. Ce LIDAR est l'élément central de notre projet puisque de c'est grâce à lui que le véhicule va éviter les obstacles.

Par ailleurs, nous avons besoin d'adapter les différentes tensions fournies par la batterie (7,2V) à la carte microcontrôleur, le LIDAR et le servomoteur (5V). Une carte d'adaptation a alors été réalisée par notre professeur, celle-ci avait pour but de remplacer le fonctionnement d'un transformateur que nous voulions utiliser au départ.

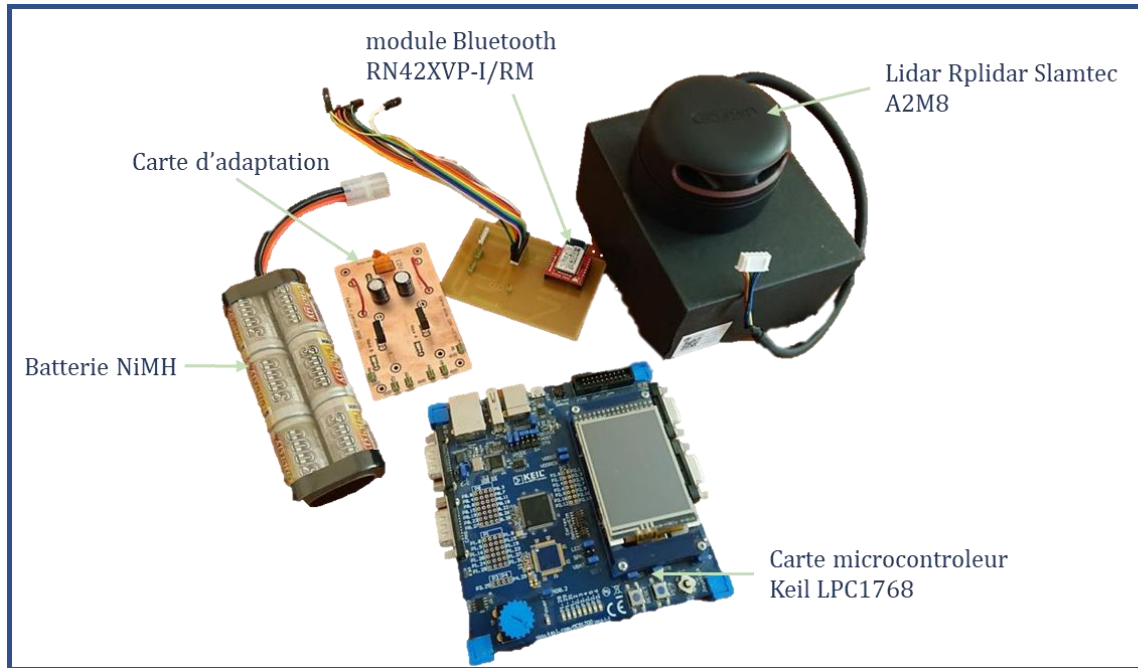


Figure 2: Le matériel fourni par le professeur

Et enfin dans le kit nous avons une carrosserie de Lancia Delta aussi semblable aux réelles voitures de course. Nous avons alors tout le matériel à disposition et nous étions prêts pour débiter l'étude mécanique.



Figure 3: La carrosserie Lancia Delta fournie dans le kit

2. Le règlement

Le but est de terminer premier d'une course composé de 6 véhicules autonomes, une course avec plusieurs tours, afin de déterminer le gagnant. Chaque équipe aura un droit à l'erreur, une deuxième chance sera alors offerte à eux en cas de problèmes techniques.

Pour effectuer cette course on nous impose un cahier des charges, le véhicule doit être fabriqué à partir du kit Tamiya TT-02, de plus la carrosserie fournie doit recouvrir le véhicule à plus de 80%. Pour alimenter le véhicule il faut une batterie NiMH de 7,2V. On nous impose également d'utiliser un LIDAR comme élément central de notre véhicule.

Et enfin, l'ensemble du véhicule composé de ses capteurs et de ses actionneurs doit entrer dans les dimensions du cahier des charges, le véhicule doit avoir une hauteur inférieure à 16 cm, sa longueur doit être inférieure à 46 cm et sa largeur à 19 cm.

Afin de savoir si le véhicule peut concourir, il faut qu'il effectue une homologation pour que le jury valide le respect du cahier des charges ainsi que la vérification de l'aptitude du véhicule à se déplacer sur une piste sans toucher les bordures.

Une fois sur la piste, la communication de l'équipe avec le véhicule doit se limiter à l'envoi d'un signal de démarrage et d'un signal d'arrêt. Afin de garantir la sécurité des biens et des personnes, si la connexion entre l'équipe et le véhicule est interrompue durant plus de 2 secondes, le véhicule devra s'arrêter de manière autonome.

Le tracé de la piste n'est pas connu en avance mais elle doit avoir des bordures de 20 cm de hauteur, ces bordures devront être composées d'éléments droits et d'arcs de cercle de rayon de courbure supérieur à 50 cm, des angles droits sur les bords de pistes sont également envisageables. Il n'y aura aucun marquage au sol, la piste doit avoir une largeur supérieure à 1m et peut contenir plusieurs obstacles en plus des autres véhicules. Afin de mieux comprendre tous ces éléments voici un exemple de piste sur laquelle les véhicules pourront s'affronter.

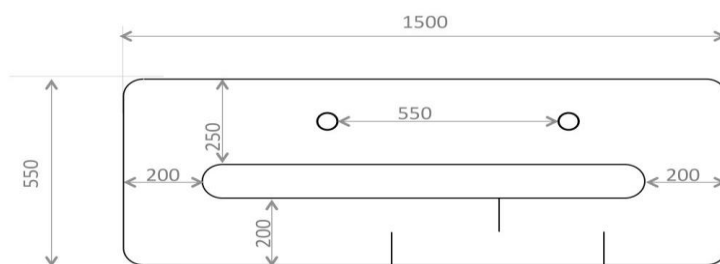


Figure 4: Exemple de piste, dimension en cm

L'ensemble des équipes aura 3 minutes pour installer leur véhicule sur la piste, une fois que les équipes sont prêtes, il est interdit de toucher les véhicules, le signal de départ est ensuite annoncé par l'arbitre. En cas de non achèvement de la course par un véhicule, la distance parcourue par celui-ci sera calculée. Et enfin, l'arbitre a le droit de retirer un véhicule s'il trouve que celui-ci est trop agressif envers ses concurrents.

II. Études et réalisations mécaniques

Une fois l'assemblage de la voiture faite, l'objectif est, dans un premier temps, de faire fonctionner le servo moteur qui va gérer la direction de la voiture et le variateur de vitesse qui gère la vitesse du véhicule.

1. Propulsion et direction du véhicule

Plusieurs des signaux utilisés sont des signaux de Modulation à Largeur d'Impulsion ou PWM (Pulse With Modulation). Ces signaux ont pour caractéristiques une fréquence qui définit leur période et un rapport cyclique qui définit la durée de l'état haut du signal.

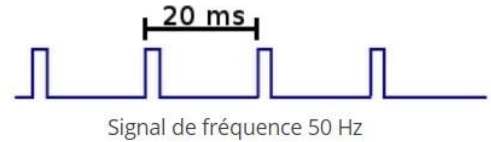


Figure 5: Exemple de PWM pour $f=50\text{Hz}$



Figure 6: Servomoteur HS-422

Le servo moteur utilisé est un HS-422. Il doit être alimenté en 5V (file rouge) et reçoit comme commande un signal PWM. Ce signal a une fréquence de 50Hz ce qui correspond à une période de 20 ms et un rapport cyclique variant selon l'angle de rotation souhaitée des roues. En branchant le servo moteur à un GBF (file orange), on a une déviation totale des roues à gauche pour un rapport cyclique de 6% ; une déviation nulle à 7,5% et une déviation totale à droite pour 9%.

Le variateur de vitesse utilisé est fourni avec le châssis. Sa référence est TBLE - 02s. Il est alimenté par la batterie et fait tourner un Moteur-540-Tamiya. Ce variateur de vitesse doit être réglé en "brushless", ce qui permet d'avoir le mode marche avant et marche arrière.

Pour cela, une fois le variateur en OFF, il faut maintenir son bouton en le mettant sur ON, relâcher, la LED clignote : rouge-vert-orange. Appuyer sur le bouton, une fois la LED sur le Vert (ce qui correspond au "brushless" mode) réappuyer sur le bouton pour confirmer.

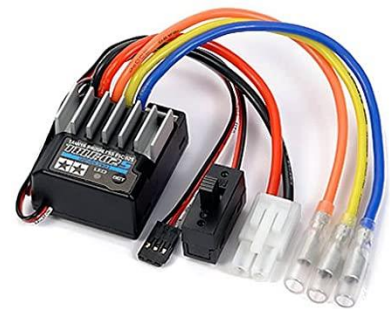


Figure 7: Variateur de vitesse TBLE-02S

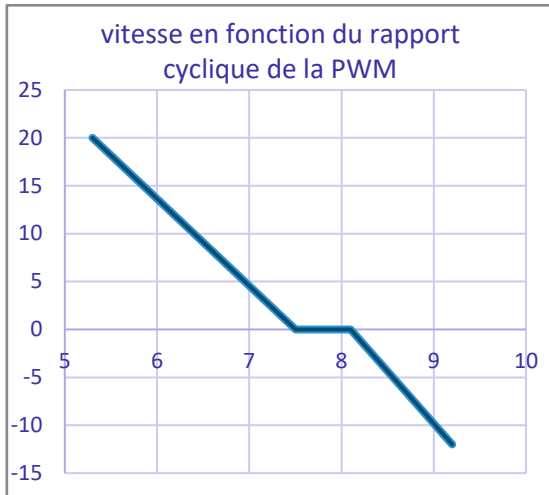


Figure 8: Variation de la vitesse en fonction du rapport cyclique

La commande à envoyer au variateur est une PWM de 50Hz et donc le rapport cyclique définit la vitesse des roues. Afin de connaître la valeur du rapport cyclique à imposer, des mesures ont été faites à partir d'une PWM envoyée par un GBF. Pour différentes valeurs du rapport cyclique, nous avons relevé la vitesse des roues et obtenu la courbe suivante. On constate que :

- la vitesse varie linéairement en fonction du rapport cyclique,
- la vitesse max est atteinte pour 5.6%
- la vitesse nulle entre 7.5% et 8.1% et on a la marche arrière max à 9.2%.

2. Le LIDAR et le module Bluetooth

Deux éléments : le LIDAR et le module Bluetooth communiquent avec le μ C à travers une liaison série : les UART (Universal Asynchronous Receiver Transmitter : émetteur-récepteur asynchrone universel). L'UART est un dispositif matériel informatique pour la communication série asynchrone dans lequel le format de données et les vitesses de transmission sont configurables. Il envoie des bits de données un par un, du moins significatif au plus significatif, encadrés par des bits de début et d'arrêt afin que le timing précis soit géré par le canal de communication. Pour ce faire, il utilise principalement 2 ports : TX pour la transmission d'information et RX pour la réception.

La documentation du LIDAR **RPLIDAR A3**, fournit les informations à lui transmettre sur ses ports :

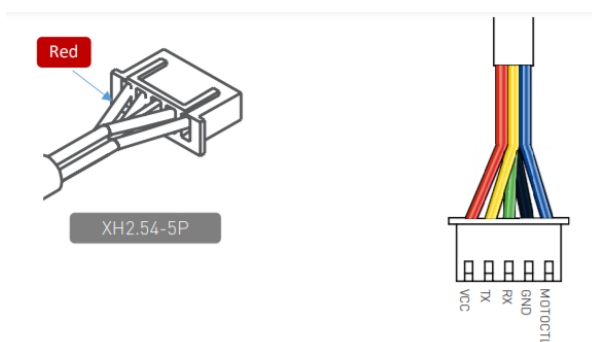


Figure 9: ports d'alimentation et de commande du LIDAR

Color	Signal Name	Type	Description	Min	Typical	Max
Red	VCC	Power	Total Power	4.9V	5V	5.5V
Yellow	TX	Output	Serial port output of the scanner core	0V	3.3V	3.5V
Green	RX	Input	Serial port input of the scanner core	0V	3.3V	3.5V
Black	GND	Power	GND	0V	0V	0V
Blue	MOTOCTL	Input	Scan motor /PWM Control Signal (active high, internal pull down)	0V	3.3V	5V

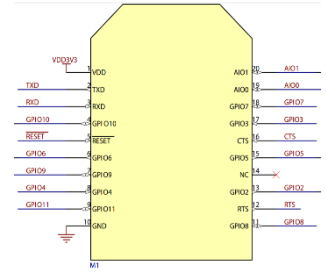
Figure 10: définition des signaux des ports du LIDAR

Il est alimenté en 5V, reçoit un signal PWM (de 25kHz et 60% de rapport cyclique) qui règle sa vitesse de rotation et communique avec le microcontrôleur à partir de deux ports TX pour la transmission d'informations et RX pour la réception.



**Figure 12: module Bluetooth
RN42XVP-I/RM**

Concernant le module Bluetooth, il s'agit d'un RN42XVP-I/RM. Il est à plusieurs ports mais nous n'allons nous intéresser qu'à cinq ports : les 3.3V et la masse pour l'alimenter, CTS pour le statut (connecté ou non), TXD et RXD pour la transmission et la réception d'information.



**Figure 11: ports du module
Bluetooth**

3. Mesure de vitesse

Afin d'asservir la vitesse et de ne pas négliger les possibles aléas qui feront modifier la vitesse réelle de la voiture, notamment un blocage des roues lors d'un choc contre un obstacle, nous avons prévu d'ajouter un capteur optique : un OPB704WZ qui une fois câblé avec ses quatre ports, va renseigner sur la vitesse réelle des roues grâce à une point blanc sur l'arbre de transmission.. Pour cela, une petite tache blanche a été faite sur la roue : le capteur détectant le nombre de fois qu'il a vu cette tache, ce qui correspond à la vitesse des roues.

III. Conception et réalisation électronique

1. Le capteur optique de mesure de vitesse

En plus du LIDAR, la voiture utilisera un capteur optique, nous avons choisi d'utiliser un OPB704WZ :

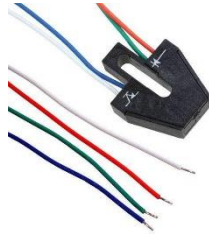


Figure 13: capteur optique OPB704WZ

Ce capteur est l'association d'un phototransistor et d'une diode électroluminescente infrarouge. C'est un capteur réfléchissant utilisé pour détecter la présence ou l'absence d'objet, ou mesurer la distance par rapport à ces objets. Dans ce capteur, la diode est associée à une résistance de $220\ \Omega$ pour maîtriser le courant entrant dans la diode. Le transistor lui, est associé à une résistance de $20k\ \Omega$. Cette résistance est nécessaire pour les deux phases suivantes :

- Lors du passage du capteur optique sur du blanc, ce qui va assurer la saturation du transistor ;
- Lors du passage du capteur optique sur du noir, ce qui va bloquer le transistor.

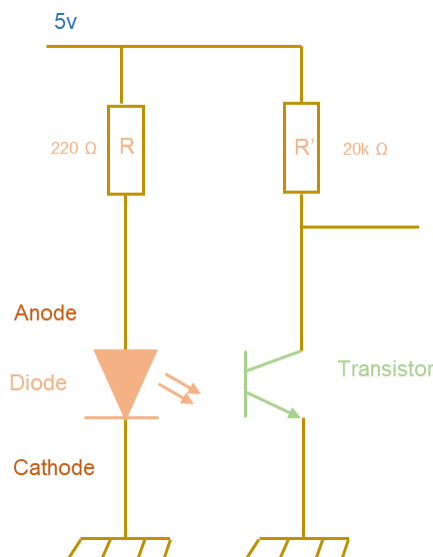


Figure 14: Montage électronique du capteur de mesure de vitesse d'arbre

On a par la suite effectué des tests pour savoir si ce capteur optique fonctionne et si c'était celui qu'il nous fallait. On a alors effectué les branchements sur la plaque Labdec puis alimenté le tout. On a ensuite passé le capteur optique sur une ligne blanche et nous avons observé le comportement du capteur avec un oscilloscope. Nous avons remarqué que lorsque le capteur passait sur une ligne blanche, un changement d'amplitude se manifesté au niveau de l'oscilloscope. Suite à cela, nous avons approuvé le fonctionnement du capteur optique OPB704WZ.

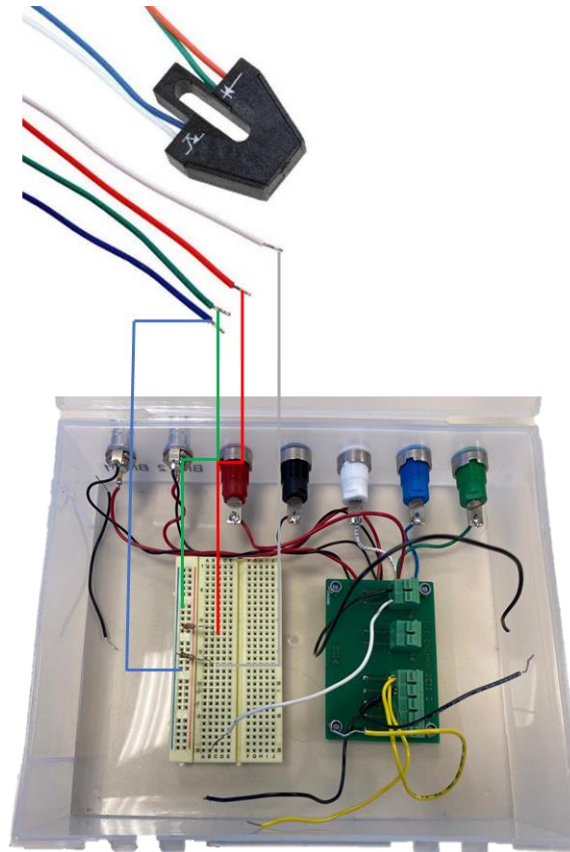


Figure 15: Test du capteur optique OPB704WZ sur la plaque Labdec

Concernant la localisation du capteur optique sur le véhicule, il sera placé à la hauteur d'une roue afin qu'il puisse mesurer la vitesse du véhicule, si ce dernier est à l'arrêt cela va permettre au véhicule d'effectuer une marche arrière.

2. La carte d'interface

La carte d'interface a été réalisée afin d'effectuer le lien entre la carte microcontrôleur LPC1768 et les différents composants (comme par exemple le module Bluetooth). Ce dernier a pour but d'effectuer les départs et arrêts à distance du véhicule.

La carte d'interface a été conçue sous le logiciel Altium Designer, ce dernier est composé de différentes bibliothèques regroupant de nombreux composants déjà créés telle qu'une résistance. La conception de cette carte nous a parue plutôt simple car peu de composant à mettre dessus. En revanche, la tâche s'est avérée plus difficile au niveau des bibliothèques car le module Bluetooth n'a jamais été créé.

Apprendre à créer une nouvelle bibliothèque pour un composant était nécessaire sur ce logiciel, en voici les étapes :

Dans un premier temps, nous recherchons la datasheet (fiche technique) du composant, puis on essaye de trouver ses mensurations. On crée alors la nouvelle bibliothèque, celle-ci est composée du schéma et de l'empreinte du composant, le schéma servira lors du placement du composant sur la carte avec l'ensemble des autres composants. L'empreinte sera utile pour avoir les mensurations identiques au composant. Lors de la création du schéma, on place le nombre de broche nécessaire ainsi que les noms de ces dernières, l'emplacement des broches est important car si on se trompe de sens lors du soudage, le composant ne sera pas dans le bon sens.

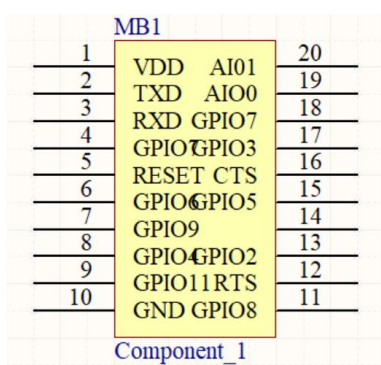


Figure 16: Schéma du module Bluetooth sur Altium Designer

Une fois le schéma créé, on passe à la création de l'empreinte du composant, il faut faire attention aux mesures indiquées dans les datasheets puisque celles-ci sont souvent en anglais puis il faut également les convertir en millimètre et enfin, en mil qui est l'unité utilisée sur Altium.

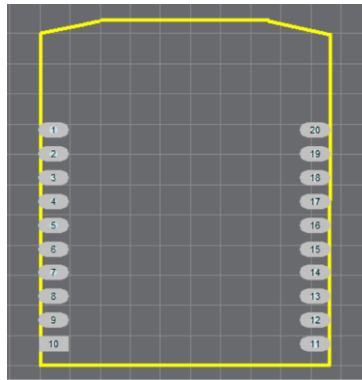


Figure 17: Empreinte du module Bluetooth sur Altium Designer

Une fois la bibliothèque compilée et intégrée dans le projet de la carte d'interface, on peut placer le nouveau composant sur la carte.

La carte est composée de connecteurs, de diodes, de résistances ainsi que du module Bluetooth, nous avons ensuite routé le PCB de la carte et nous l'avons envoyé à notre professeur pour qu'il vérifie notre création avant son impression.

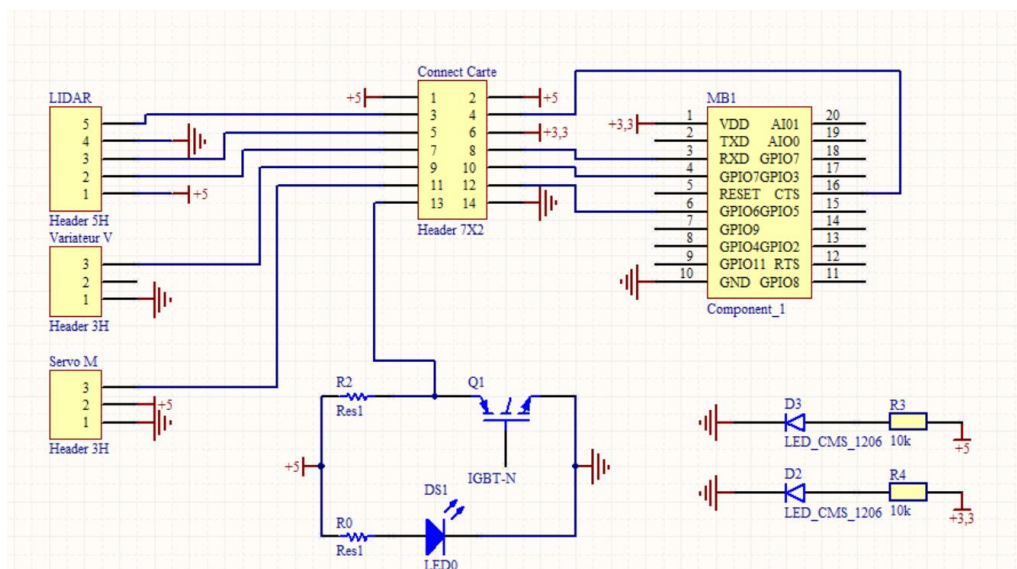


Figure 18: Schéma de la carte interface

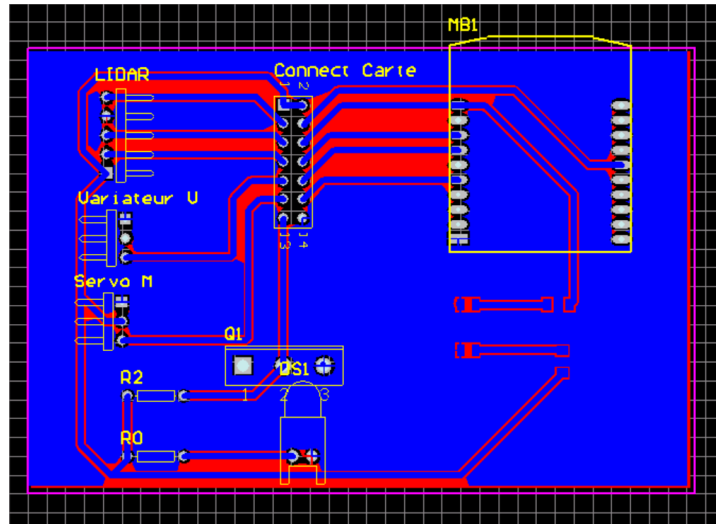


Figure 19: PCB et routage de la carte interface

Voici le résultat final de la carte imprimée puis soudée avec l'ensemble des composants.



Figure 20: Carte interface avec l'ensemble des composants soudés

Une fois les éléments soudés, nous avons constaté qu'une inversion des connections du module Bluetooth a été réalisée. Nous savons dorénavant qu'il faut porter une attention particulière lorsque l'on connecte les différents composants lors de la création de la carte. Refaire une nouvelle carte n'a pas été nécessaire car une solution a été trouvée avec l'aide de notre professeur, en gardant la carte initialement réalisée.

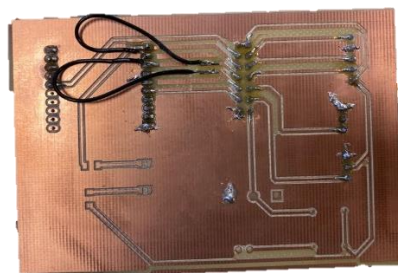


Figure 21: Correction du branchement du module Bluetooth sur la Bottom layer

Pour nos successeurs, voici le branchement que nous aurions dû réaliser.

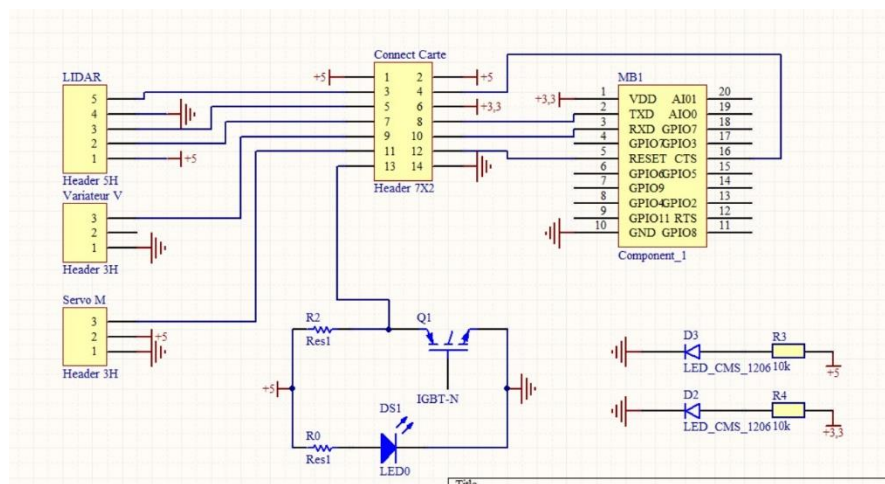


Figure 22: Schéma rectifié de la carte d'interface

Une fois les éléments mécaniques réalisés, nous étions prêts pour la programmation du véhicule.

IV. Informatique

1. Communication entre le microcontrôleur et chacun des composants

Une fois chaque élément connu, il nous a été plus simple de programmer. La programmation se fait en langage C et nous avons créé des fonctions pour simplifier l'écriture des programmes.

Concernant la commande du variateur de vitesse et du servomoteur deux fonctions ont été créées :

La première : `void moteur (int sens ; int vitesse)` ; elle reçoit en entrée le sens de rotation des roues (avant ou arrière) et la vitesse de ces dernières et génère sur le port 2.3 une PWM.

La deuxième : `void dir_Voiture (int position)` ; elle reçoit en entrée la position dans laquelle elle doit orienter les roues : gauche, centre ou droite et génère sur le port 2.4 une PWM.

Ces PWM ont une fréquence de 50Hz : réglée par le code `LPC_PWM1->MR0 = 499999` ; et un rapport cyclique dépendant de l'entrée qui a été passée en paramètre. Ce rapport cyclique alpha est réglé avec le code `LPC_PWM1->MRx = Y` ; avec $Y = \alpha * MR0$. Ce rapport cyclique détermine la durée de l'état haut de la PWM ce qui va commander la vitesse pour la première fonction et la direction pour la deuxième fonction. Aussi, devons-nous générer une PWM de fréquence 25kHz et de rapport cyclique 60% pour commander la vitesse de rotation du LIDAR. Pour ces paramètres on a `LPC_PWM1->MR0 = 999` ; et `LPC_PWM1->MR3 = 599` ;

Chaque PWM doit être envoyé sur des ports différents. La sélection du port se fait avec deux lignes de code : `LPC_PINCON->PINSEL4 |= (1<<4)` ; qui permet de choisir les `PWM1` et `LPC_PWM1->PCR |= (1<<11)` ; qui permet de choisir le générateur 3 des PWM1.

Une fonction type permettant de générer une PWM est la suivante :

```
void lidar_MOTOCLT (void) // P=25kHz; %haut=60%; sur P2.2
{
    LPC_SC->PCONP = LPC_SC->PCONP | (1<<6);
    LPC_PINCON->PINSEL4 |= (1<<4);           //PWM 1.3

    LPC_PWM1->CTCR = 0;                       //mode timer
    LPC_PWM1->PR = 0;                         //Pas de prédivision
    LPC_PWM1->MR0 = 999;                      //valeur de N pour une fréquence de 25kHz
    LPC_PWM1->MCR |= (1<<1);                  //RAZ du compteur si correspondance avec MR0
    LPC_PWM1->PCR |= (1<<11);                 //active PWM3
    LPC_PWM1->MR3 = 599;                      //rapport cyclique de 60%
    LPC_PWM1->TCR = 1;                        //démarrage du timer
}
```

Figure 23: Code type d'une fonction d'initialisation d'une PWM

Deux des composants communiquent avec le μ C à partir de liaisons séries : le LIDAR et le module Bluetooth.

Avant d'utiliser ces liaisons, il faut:

- les paramétrer en leur indiquant la taille des informations à transmettre, la vitesse de transmission des informations, la présence ou non d'un bit de parité et d'un bit de fin de transmission
- définir les ports pour la transition TX et la réception RX.

Cette initialisation peut être intégrée dans une fonction et ainsi être facilement appelée au début du main. On a donc le code suivant :

```
void INIT_BT (void) // broches P0.2 pour TX et P0.3 pour RX
{
    LPC_PINCON -> PINSEL0 |= 0x5 << 4 ; // broches P0.2 pour TX et P0.3 pour RX
    LPC_UART0 -> LCR = 0x03 ;           // 8 bits sans parité, 1 bit de stop

    // Réglage de la vitesse de transmission
    LPC_UART0 -> LCR |= 0x80 ;           // Forçage bit DLAB=1 (Demande autorisation de modification)
    LPC_UART0 -> DLM = 0 ;               // Pas de sur-division de PCLK
    LPC_UART0 -> DLL = 9 ;               // Division principale par 9 de PLCK
    LPC_UART0 -> FDR = 0x21 ;            // Division fractionnaire de 1,5 (DIVAddVAL=1 et MULVAL=2)
    LPC_UART0 -> LCR &= 0x7F ;           // Forçage bit DLAB=0 (Fin d'autorisation de modification)
}
```

Figure 24: Code type d'une fonction d'initialisation d'un UART

En ce qui concerne la mesure de la vitesse, le capteur optique envoie une impulsion à chaque passage sur du blanc. Pour mesurer la vitesse, on compte donc le nombre de fronts montants par secondes sur l'entrée reliée au capteur optique. En générant une impulsion toutes les secondes, on peut relever le nombre de fronts montants qui a été fait (en 1seconde) et le remettre à 0 pour une nouvelle mesure. On aura donc ainsi la vitesse que l'on cherche à mesurer.

Notre carte LPC1768 nous permet d'avoir des interruptions à une fréquence précise et aussi de compter les fronts montants sur un port de la carte. Pour générer les interruptions et le comptage, il faudrait les initialiser :

Le compteur d'évènement est géré par un timer : il compte le nombre de périodes sur un port. Son initialisation consiste à définir le port sur lequel il y aura le signal et à choisir si l'on souhaite compter les fronts montants ou les fronts descendants : dans notre cas les fronts montants. On a donc le code suivant :

```
LPC_SC->PCONP = LPC_SC->PCONP | (1<<22);           //Timer ON
LPC_PINCON->PINSEL0 = LPC_PINCON->PINSEL0 | (3<<10); // P0.5 en mode CAP2.1
LPC_TIM2->CTCR = 0x05;                               // fronts montants voie 1
LPC_TIM2->TCR = 1;                                    //démarre le comptage
```

Figure 25: Code de l'initialisation des interruptions sur timer

Concernant l'interruption, elle doit être reliée à un timer pour être faite à une fréquence de 1Hz ce qui correspond à 1s. Son initialisation consiste à créer le signal carré qui va définir la fréquence de l'interruption et par la suite renvoyer à une certaine fonction : dans notre cas la fonction `void TIMER1_IRQHandler (void)` : il s'agit d'une fonction déjà générée par la carte LPC1768 en vue des interruptions. On a donc le code d'initialisation suivant :

```
LPC_TIM1->CTCR = 0;           // mode timer( signaux carré)
LPC_TIM1->PR = 0;              // prescaler à 0
LPC_TIM1->MR0 = 12500000 -1;    // valeur de N pour avoir 1s
// RAZ si correspondance avec MR0, et interruption
LPC_TIM1->MCR = LPC_TIM1->MCR | (3<<0);
LPC_TIM1->TCR=1;               //lancement du timer

NVIC_SetPriority(TIMER1_IRQn, 0); // TIMER1 : IT de priorité 0
NVIC_EnableIRQ(TIMER1_IRQn);     // active IT du TIMER1
```

Figure 26: Code de l'initialisation du compteur d'évènements

Une fois le compteur d'évènement et les interruptions initialisés, il faut enregistrer la valeur de la vitesse dans une variable pour pouvoir l'afficher ou s'en servir. Vu que la fonction à laquelle renvoie le timer ne peut pas retourner de valeur, il faut donc créer une variable à laquelle toutes les fonctions peuvent avoir accès : une variable globale `#define vitesse` ; Aussi, lorsque la fonction d'interruption est appelée, un drapeau est levé. Ce drapeau indique qu'une interruption est demandée. Lorsque la fonction est exécutée, il faut baisser le drapeau pour le signaler. Dans la fonction `TIMER1_IRQHandler`, on doit donc baisser le drapeau, lire le nombre de fronts montants, l'enregistrer dans la variable vitesse, et le remettre à 0. On a donc le code de fonctions suivant :

```
void TIMER1_IRQHandler (void)
{
    LPC_TIM1->IR = LPC_TIM1->IR | (1<<0) ; //acquittement pour le Match Register 0 du timer0 (abaissement du drapeau)
    vitesse = LPC_TIM2->TC;                 //lecture du compteur
    LPC_TIM2->TC = 0;                       // remise à 0 du compteur, de la mesure
}
```

Figure 27: Code de la Fonction d'interruption

2. Algorithme de conduite autonome

Afin d'avoir une voiture autonome, nous devons modifier la direction et la vitesse en fonction des informations fournies par le LIDAR.

Le LIDAR nous donne pour plusieurs angles θ , la distance d qui nous sépare d'un obstacle. Avec ces valeurs nous pouvons vérifier la présence ou l'absence d'un obstacle sur la trajectoire de la voiture, que cette trajectoire soit rectiligne ou courbée. Nous pouvons aussi déterminer quelle est la distance qui sépare la voiture de la bordure.

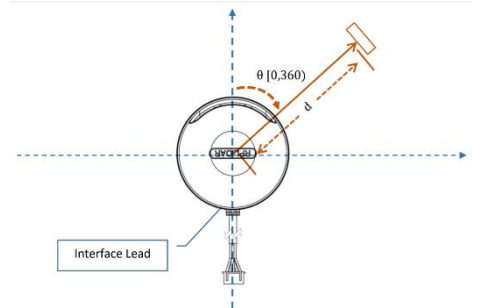
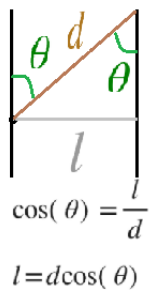


Figure 28: définition de la mesure des données fournies par le LIDAR

- **Détermination de la distance l entre la voiture et la bordure :**



Le LIDAR étant placé au centre de la voiture, les angles de la voiture sont à 45° , 135° , 225° et 315° . En prenant en compte l'angle de mesure θ et la distance d fournie par le LIDAR, la largeur l qui sépare le LIDAR de la bordure est égale à $l = d\cos(\theta)$.

Figure 29: illustration distances LIDAR-bordure

Cependant, si la voiture n'est pas parallèle à la bordure, il faudrait retirer la distance relative entre le lidar et le coté de la voiture le plus proche de la bordure. On a donc la formule suivante : $X = l \cdot \frac{O}{d_2}$. La distance relative entre la voiture et la bordure de piste est donc $l - X = d\cos(\theta) \left(1 - \frac{O}{d_2}\right)$ avec O la distance entre le lidar et l'extrémité de la voiture.

- **Rectification de la trajectoire de la voiture afin qu'elle soit parallèle à la bordure :**

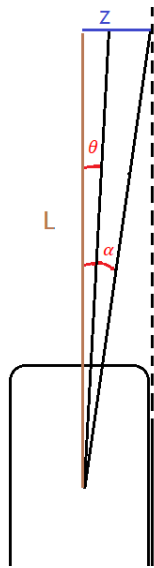
En relevant la distance d fournie par le LIDAR à 45° et à 135° , on pourrait les comparer :

Si $d_1 = d_2$, la trajectoire de la voiture et la bordure sont parallèles.

Si $d_1 < d_2$, il faudrait que la voiture tourne à gauche jusqu'à ce que $d_1 = d_2$.

Si $d_1 > d_2$, il faudrait que la voiture tourne à droite jusqu'à ce que $d_1 = d_2$.

- Vérifier la présence ou l'absence d'un obstacle sur la trajectoire de la voiture pour une trajectoire linéaire :



Le but est de vérifier la présence d'un obstacle à une certaine distance L en face de la voiture. Pour cela, il faudrait considérer les distances d fournies par le lidar entre $(360^\circ - \alpha)^\circ$ et 0° et entre 0° et α° . Comme représenté sur la figure ci contre. On a donc un angle $\alpha = \tan^{-1} \left(\frac{z}{L} \right)$

En fonction de l'angle θ et de la distance d fournie par le lidar, on peut déterminer la longueur $L = d \sin(\theta)$ qui sépare le LIDAR d'un obstacle. Si pour l'un des angles $(360^\circ - \alpha)^\circ$ et 0° et entre 0° et α° , on a une distance $L < Y$, cela veut dire que l'on a un obstacle sur la trajectoire de la voiture à moins de la distance Y .

Figure 30: illustration de la détection d'obstacle

Pour aller plus loin une fois la méthode pour détecter les obstacles trouvés, l'algorithme pour dépasser les voitures mobiles et les obstacles fixes trouvé, on peut écrire une machine à état qui va détailler les différentes étapes possibles et les moyens de passage d'une étape à une autre. Pour exemples :

- Marche avant ; si obstacle à moins de la distance Y ,
- Arrêt ; si aucun obstacle à la droite : entre 30° et 60° ,
- Direction droite et moteur avant ; pendant 2 secondes
- Direction gauche jusqu'à ce que la trajectoire soit parallèle à la bordure de piste
- Etc...

Il restera à coder cette machine à état en switch case. Le projet étant long et complexe, nous n'avons pas mis en application cet algorithme de conduite. Cependant, nous sommes parvenus à commander la direction et la vitesse à partir des informations reçues par le module Bluetooth et nous avons même réussi à faire une démonstration de la détection des obstacles par le LIDAR.

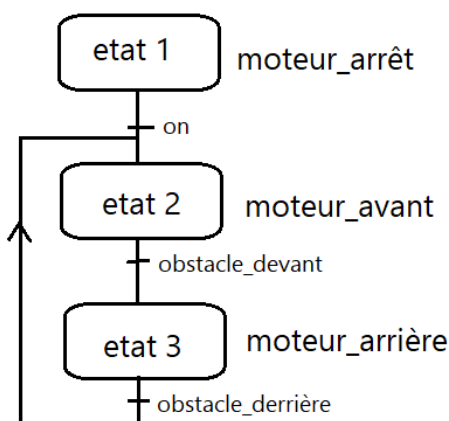


Figure 31: Machine à état illustrant une démo de la détection obstacles

CONCLUSION

Ce projet s'est révélé très enrichissant dans la mesure où il a consisté en une infiltration du métier d'ingénieur. Effectivement, la prise d'initiative, le respect des délais mais aussi le travail en équipe seront les aspects essentiels de notre futur métier.

Cela nous a également permis d'appliquer nos connaissances en génie électrique et en programmation à un domaine pratique mais avant tout l'apprentissage en autonomie.

Les principaux problèmes que nous avons rencontrés, concernaient la compréhension de l'utilisation de chaque élément du véhicule par exemple le LIDAR ou bien le variateur de vitesse car ces éléments étaient tout nouveaux pour nous toutefois grâce à une bonne cohésion et l'aide de notre professeur nous nous sommes toujours en sorties.

À présent, nous n'avons pas rendu la voiture autonome, cependant on nous a donné l'opportunité de finir ce projet qui est en corrélation avec une des spécialités du semestre 4. Il ne reste plus qu'à terminer cette partie et la voiture sera prête à concourir à l'ENS Cachan.

Nous sommes fiers de notre voiture autonome et de son avancement, nous continuerons à l'améliorer et sommes impatient de voir le résultat final.

LISTE DES FIGURES

Figure 1: Le châssis construit	7
Figure 2: Le matériel fournit par le professeur	8
Figure 3: La carrosserie Lancia Delta fournit dans le kit	8
Figure 4: Exemple de piste, dimension en cm	9
Figure 5: Exemple de PWM pour $f=50\text{Hz}$	10
Figure 6: Servomoteur HS-422	10
Figure 7: Variateur de vitesse TBLE-02S	10
Figure 8: Variation de la vitesse en fonction du rapport cyclique	11
Figure 9: ports d'alimentation et de commande du LIDAR	Figure 10: définition des signaux des ports du LIDAR.....
11	11
Figure 11: ports du module Bluetooth	12
Figure 12: module Bluetooth RN42XVP-I/RM	12
Figure 13: capteur optique OPB704WZ	13
Figure 14: Montage électronique du capteur de mesure de vitesse d'arbre	13
Figure 15: Test du capteur optique OPB704WZ sur la plaque Labdec	14
Figure 16: Schéma du module Bluetooth sur Altium Designer	15
Figure 17: Empreinte du module Bluetooth sur Altium Designer	16
Figure 18: Schéma de la carte interface	16
Figure 19: PCB et routage de la carte interface	17
Figure 20: Carte interface avec l'ensemble des composants soudés	17
Figure 21: Correction du branchement du module Bluetooth sur la Bottom layer	17
Figure 22: Schéma rectifié de la carte d'interface	18
Figure 23: Code type d'une fonction d'initialisation d'une PWM	19
Figure 24: Code type d'une fonction d'initialisation d'un UART	20
Figure 25: Code de l'initialisation des interruptions sur timer	20
Figure 26: Code de l'initialisation du compteur d'évènements	21
Figure 27: Code de la Fonction d'interruption	21
Figure 28: définition de la mesure des données fournies par le LIDAR	22
Figure 29: illustration distances LIDAR-bordure	22
Figure 30: illustration de la détection d'obstacle	23
Figure 31: Machine à état illustrant une démo de la détection obstacles	23

WEBOGRAPHIE

- Châssis TAMIYA
https://www.latelierdutrain.com/rally/119328-lancia-delta-integrale-tt02-4wd-kit-110-tamiya-58570-4950344585700.html?search_query=LANCIA+DELTA+INTEGRALE+TT02&results=33
- Datasheet RPLIDAR A3M1
https://www.generationrobots.com/media/LD310_SLAMTEC_rplidar_datasheet_A3M1_v1.0_en.pdf
- Datasheet carte Keil
<https://www.nxp.com/docs/en/user-guide/UM10360.pdf>
- Datasheet module Bluetooth OPB704WZ
<https://docs.rs-online.com/d3e4/0900766b81179734.pdf>
- Datasheet variateur de vitesse TBLE-02s
<https://images.app.goo.gl/TUrRwXVgJA8WYc7h6>
- Datasheet: module Bluetooth RN42XVP-I/RM
<https://www.mouser.fr/datasheet/2/268/50002328A-1101962.pdf>
- Datasheet : variateur de vitesse TBLE-02S
<https://www.tamiyausa.com/media/files/45057ml-801-efb9.pdf>
- Datasheet : capteur optique OPB704WZ
<http://www.farnell.com/datasheets/2331996.pdf>