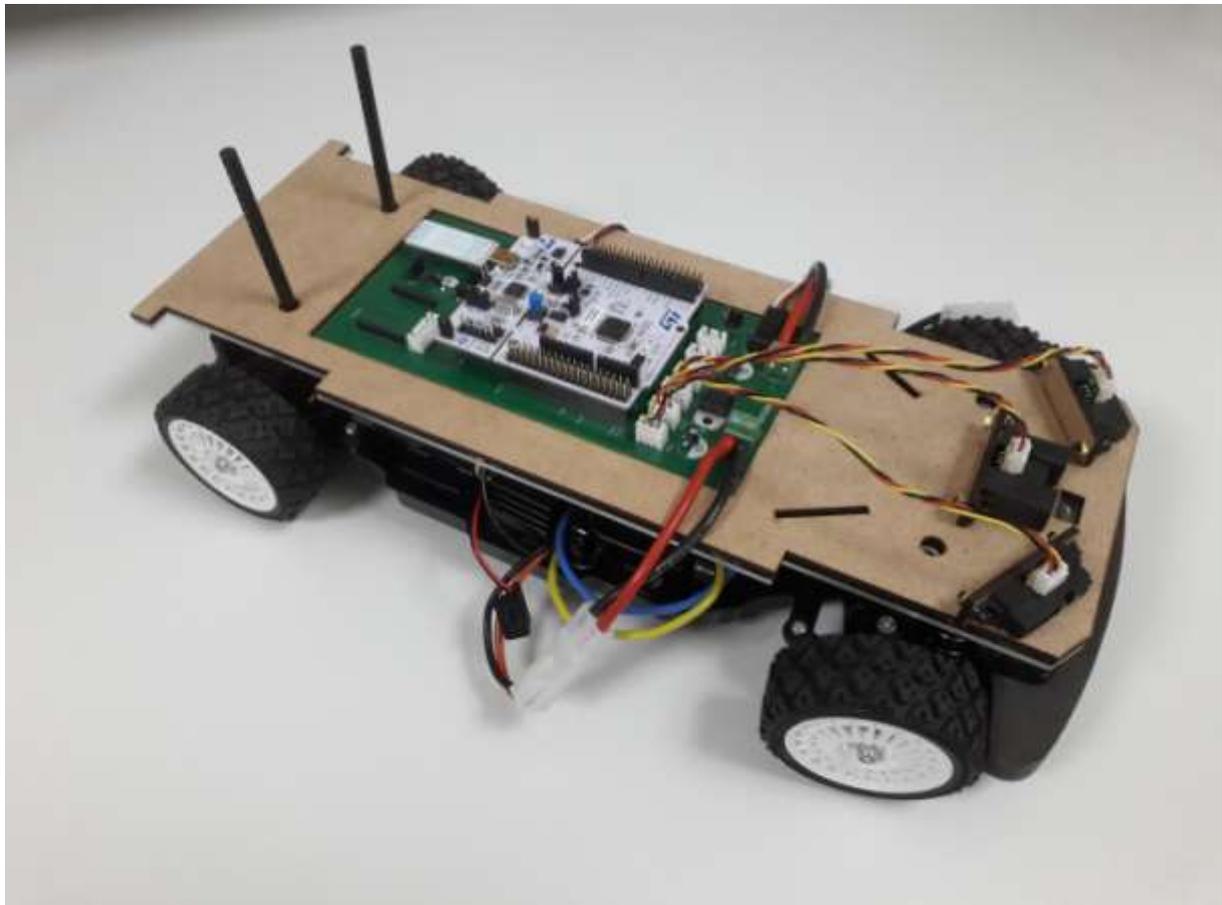


Projet IETI - Voiture autonome

Benoit Kaczmarczuk
Abdellah Elmennaoui
Noé Villemagne

Rapport technique



Janvier – Mai 2021

I] Introduction

Dans le cadre de l'UE IETI inscrite au programme du second semestre de la 1^{ère} année à l'Institut d'Optique Graduate School, les étudiants doivent mener un projet d'électronique au choix parmi plusieurs propositions. Ce projet est ici porté par une équipe de trois étudiants :

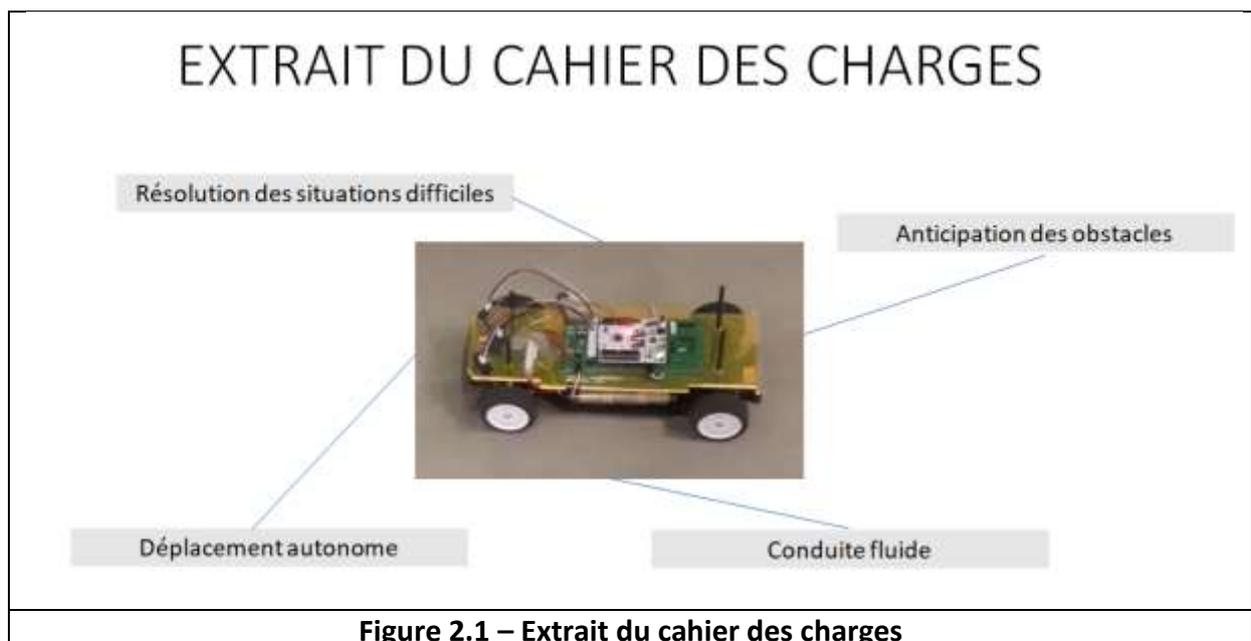
Benoit Kaczmarczuk
Abdellah Elmennaoui
Noé Villemagne

sur le sujet : Voiture autonome

Le but de cet enseignement par projets est de donner du temps aux étudiants pour comprendre, manipuler et maîtriser plusieurs notions d'électronique grâce à des ressources disponibles sur le site du Lense : lense.institutoptique.fr et à l'aide des enseignants.

III] Objectifs du système

Pour donner une direction au travail en début de projet, il est indispensable de se donner des objectifs qui seront adaptés en fonction de l'avancée du projet. Ces objectifs sont définis par un cahier des charges (Figure 2.1).



Comme indiqué sur cet extrait du cahier des charges, la voiture doit donc à terme être capable de se déplacer de manière autonome en évitant les obstacles, en adaptant sa vitesse tout en gardant un déplacement fluide et de se sortir de situations difficiles (dans le cas où la voiture serait dans l'impossibilité d'avancer à cause d'un obstacle devant elle). Ce système embarqué est donc réalisé sur un châssis de voiture avec un moteur à courant continu, un servomoteur, une batterie et équipée d'un microcontrôleur (carte Nucléo) exécutant un code.

III] Découpage fonctionnel du système

Afin de se représenter toutes les fonctionnalités présentes sur le système et quel est techniquement le travail à faire, on crée donc un schéma fonctionnel (Figure 3.1).

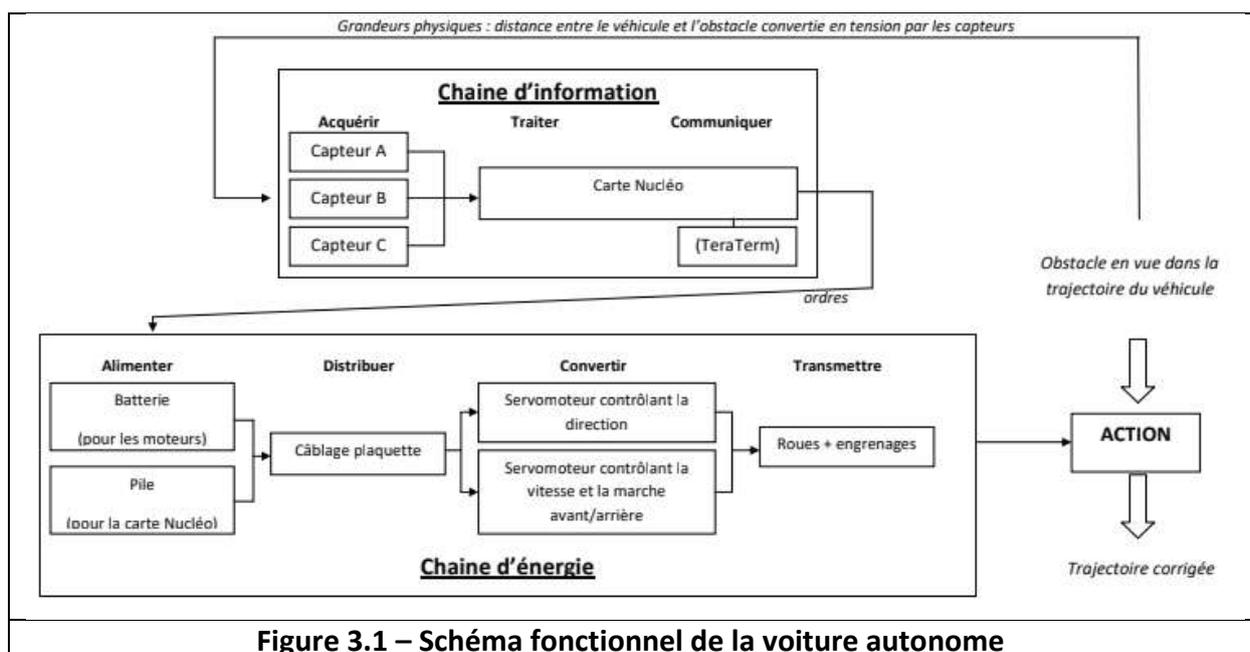
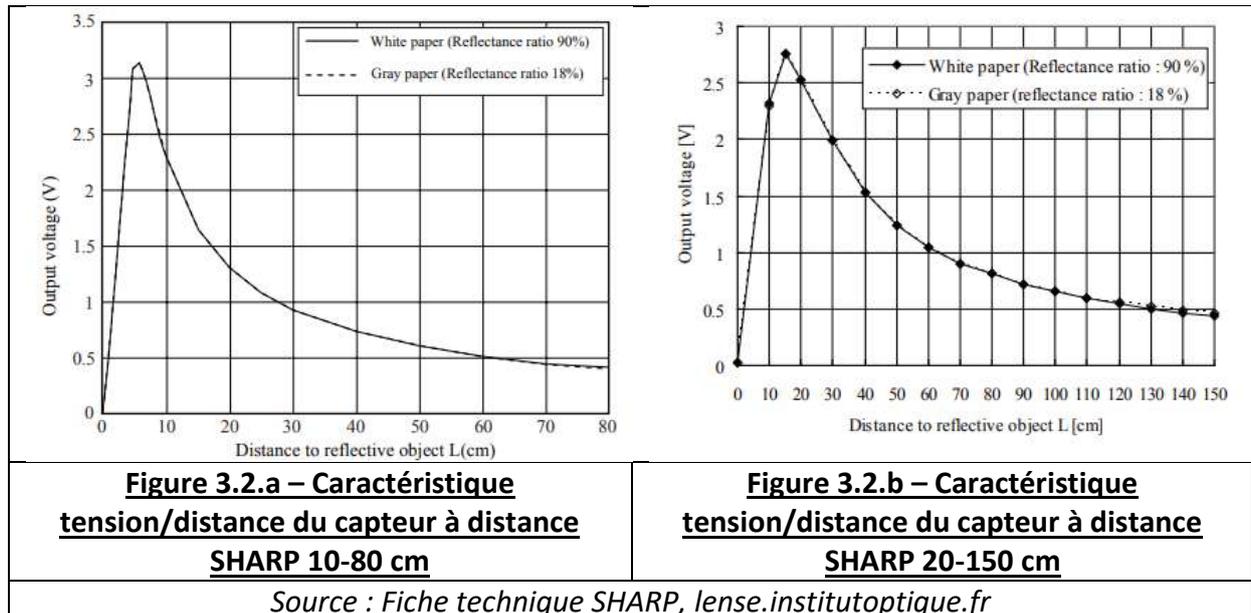


Figure 3.1 – Schéma fonctionnel de la voiture autonome

La détection d'obstacle dans notre problème est donc assurée par 3 capteurs (2 capteurs courte distance sur les cotés et 1 capteur plus longue distance au centre). La détection d'obstacle dans l'espace peut également se faire avec un seul capteur : le LIDAR. Ce capteur tournant permet d'acquérir des données dans les 360° du plan horizontal de la voiture en étudiant à plus longue distance. Malheureusement, cette solution n'a pas pu être présentée car son développement n'a pas pu être terminé à temps. Dans notre projet, les 3 capteurs transmettent donc une tension en fonction de la

distance à l'obstacle. Ces capteurs ont été caractérisés expérimentalement et présentent une caractéristique théorique disponible ci-dessous (Figure 3.2).



Ces informations sont ensuite transmises à la carte Nucléo qui traite ces données grâce au code que l'on téléverse, pour distinguer les différentes situations à laquelle la voiture peut être confrontée. Le microcontrôleur ordonnera donc en fonction de ces situations, l'action à faire sur les deux moteurs : un premier (moteur à courant continu) contrôlant la vitesse de rotation des roues et un second (servomoteur asservi en angle) contrôlant la direction/orientation des roues. Ces actions sont pilotées par des commandes PWM qui envoient une impulsion sur une période de 20ms. Pour le moteur à courant continu, une impulsion de temps inférieure à environ 1.5 ms se traduit par une rotation en sens inverse (marche arrière), pour un temps haut d'environ 1.5-1.6 ms, les roues ne tournent pas et pour un temps haut supérieur, elles tournent (marche avant) avec une vitesse croissante en fonction de ce rapport cyclique. Pour le servomoteur, un temps haut de 1.1 ms correspond à l'angle de « butée » à gauche tandis que 1.5 ms correspond à l'angle de butée à droite et 1.3 ms à un positionnement droit des roues. Ces moteurs sont alimentés par une batterie 8V.

IV] Schéma électrique du système

Evidemment, pour faire fonctionner ce système, il est nécessaire que le montage soit opérationnel. Les moteurs ainsi que les capteurs et la carte Nucléo doivent être alimentés par la batterie. Les informations données par les capteurs doivent être acquises par le microcontrôleur et ce dernier doit être relié également aux moteurs pour leur transmettre la commande à effectuer. Lors des premières séances, pour découvrir le fonctionnement de ce montage, le câblage se fait sur une plaquette de prototypage avec de très nombreux fils. Ensuite, les enseignants nous ont fourni une plaquette créée au Lense (Figure 4.1), sur laquelle les branchements ont été simplifiés, les différents ports reliés à la batterie et tels que leur placement permette une utilisation appropriée de la voiture, sans trop de fils qui s'entremêlent.

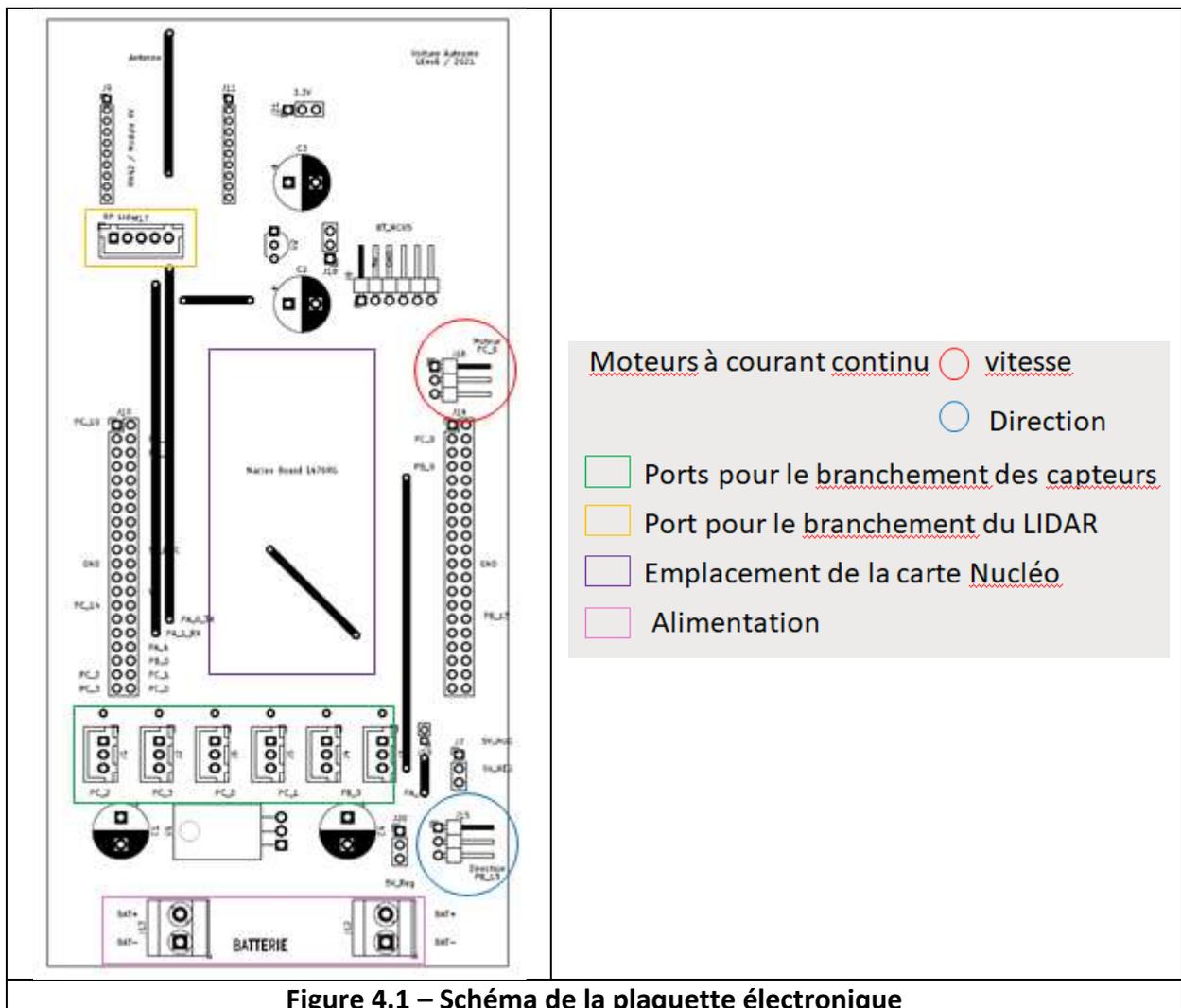


Figure 4.1 – Schéma de la plaquette électronique

V] Structure de l'algorithme

Une fois que le câblage de la plaquette électronique est correct et relié à la voiture, la partie n'est pas terminée. Il faut ensuite avoir un code fonctionnel (ANNEXE 1) pour que la voiture se déplace conformément à nos attentes. C'est une partie compliquée du projet car il faut obtenir un code ordonnant une vitesse adéquate à la voiture (rapport cyclique), qui lui permette d'anticiper suffisamment les obstacles (valeur seuil des tensions). Afin de savoir à chaque modification, l'effet du code sur la voiture, on effectue d'abord les tests sur banc en programmant un affichage sur le logiciel TeraTerm en liaison série avec la carte Nucléo puis on procède aux tests sur sol. On cherche alors à observer si la voiture répond bien comme on le souhaite et selon notre conception du code (Figure 5.1).

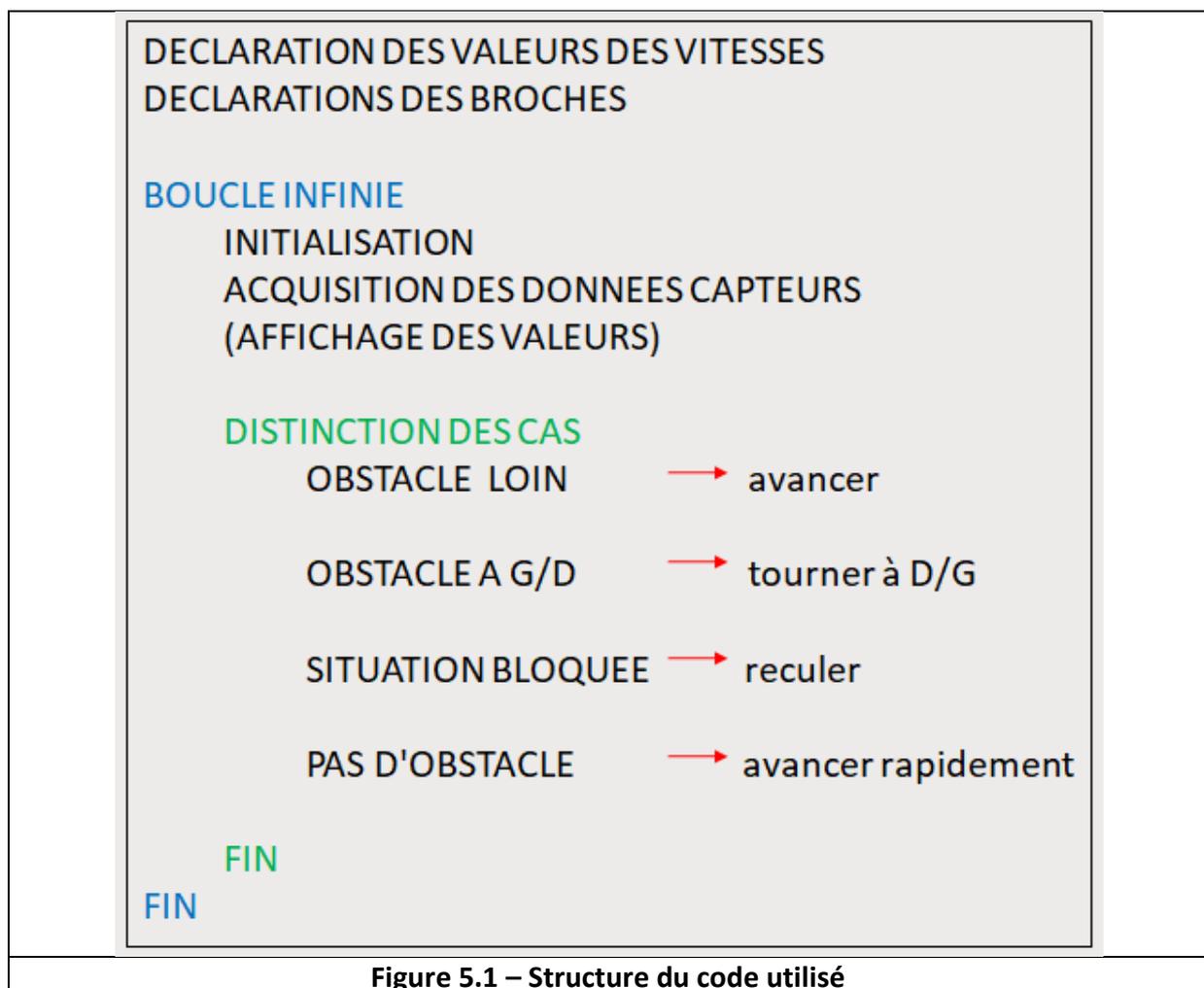


Figure 5.1 – Structure du code utilisé

VII] Tests et validation

Dans le but d'arriver à nos fins, la voiture doit répondre le plus possible favorablement à toutes les situations à laquelle on décide de la confronter. Comme expliqué précédemment, lors de la création du code, il suffit de tester sur le bureau les réactions de la voiture afin de voir si elle tourne bien dans la direction opposé à un obstacle (main). Ensuite, lorsque le code est devenu fonctionnel, il fallait modifier tous les paramètres pour obtenir une conduite la plus fluide possible et optimiser l'anticipation des obstacles. Il a donc fallu tester plusieurs valeurs de rapport cyclique et de seuil de tension (parfois au centième près pour le rapport cyclique). Lors des tests, on propose initialement une direction orientée vers un obstacle pour voir si elle va réussir à l'éviter. On pourra également la positionner directement face à un obstacle (mur) pour voir si elle va bien essayer de reculer pour sortir de cette situation.

Lors de la démonstration finale, la voiture est alors capable de se déplacer dans le couloir en évitant les murs et en augmentant sa vitesse dans le cas où aucun obstacle ne fait barrage. Dans les cas où la voiture vient à prendre une mauvaise décision ou à être bloquée face à un mur, elle fait marche arrière afin de retrouver une situation favorable. Pour observer le comportement de la voiture, une vidéo est disponible à l'adresse suivante (Figure 6.1).



Figure 6.1 – Lien vidéo et QR code

VII] Planning

Voici le planning mis à jour (Figure 7.1) qui reflète l'avancée de notre projet au cours des séances. Il a été adapté par rapport au planning qui avait été établi lors des premières séances en raison des difficultés rencontrées (Partie VIII).

Projet IETI - Voiture autonome			
Benoit Kaczmarczuk Abdellah Elmennaoui Noé Villemagne		Planning	
			
Planning	Abdellah	Noé	Benoit
Séance 1	Découverte	Découverte	Découverte
	Caractérisation des tensions des moteurs	Caractérisation des tensions des moteurs	Caractérisation du capteur "1m"
	Création des codes pour contrôler les moteurs	Prise en main du contrôle des moteurs	Conception d'un code pour relier capteur et moteur
Séance 2	Finition du programme capteur tout ou rien	Schéma bloc, planning	Création du programme 3 capteurs
	Câblage des 3 capteurs, de la pile	Test sur table de fonctionnement du code	Cahier des charges
Séance 3	Tests sur sol du programme 3 capteurs	Tests sur sol du programme 3 capteurs	Tests sur sol du programme 3 capteurs
	Amélioration du programme : Ajout de wait Réglage de la vitesse Réglage des tensions seuils des capteurs	Amélioration du programme : Ajout de wait Réglage de la vitesse Réglage des tensions seuils des capteurs	Amélioration du programme : Ajout de wait Réglage de la vitesse Réglage des tensions seuils des capteurs
Séance 4	Tests sur sol du programme amélioré	Tests sur sol du programme amélioré	Tests sur sol du programme amélioré
	Corrections des problèmes	Corrections des problèmes	Corrections des problèmes
	Passage sur carte électronique adaptée	Passage sur carte électronique adaptée	Passage sur carte électronique adaptée
Séance 5	Dernières corrections du programme avec les 3 capteurs	Dernières corrections du programme avec les 3 capteurs	Dernières corrections du programme avec les 3 capteurs
	Travail sur la marche arrière	Travail sur la marche arrière	Travail sur la marche arrière
	Tests sur sol	Tests sur sol	Tests sur sol
Séance 6	Passage avec le LIDAR : Compréhension du capteur	Passage avec le LIDAR : Compréhension du capteur	Passage avec le LIDAR : Compréhension du capteur
	Premier programme et premier test utilisant le LIDAR	Premier programme et premier test utilisant le LIDAR	Premier programme et premier test utilisant le LIDAR
Séance 7	Amélioration du programme avec le LIDAR	Amélioration du programme avec le LIDAR	Amélioration du programme avec le LIDAR

Mis à jour : 13/04/21

Figure 7.1 - Planning

VIII] Difficultés et analyse du travail

De manière générale, nous sommes fiers du travail que nous avons réalisé. En effet, notre voiture se comporte bien comme on l'avait imaginé, en évitant les obstacles, accélérant dans un espace libre et reculant lorsqu'elle se trouve dans une situation bloquée. Notre code est fonctionnel et les paramètres peuvent être facilement modifiés dans le but d'adapter la vitesse du véhicule ou en cas de déséquilibre dans les tensions des capteurs. Mais l'avancée du projet n'est due qu'à la rencontre de difficultés que nous avons du surmonter. Dans une première partie du projet, le câblage a pu nous poser quelques problèmes avant que nous arrivions à le maîtriser parfaitement, bien aidé par la création de la plaquette électronique par les équipes du Lense. Quelques problèmes ont également été posés par une mauvaise utilisation des moteurs (mauvaise initialisation, oubli de passage au neutre). De plus, il a fallu étudier régulièrement le comportement des capteurs qui parfois ne fonctionnaient pas symétriquement (gauche et droite). C'est-à-dire que l'un des capteurs pouvait donner une tension différente que l'autre pour une même distance à un obstacle. Malgré cela, nous avons pu rapidement obtenir un prototype convenable à la moitié des séances. Cependant, nous avons ensuite pris du temps dans l'optimisation des réglages et surtout nous avons dû faire face à notre plus grande difficulté du projet. En effet, après cela, nous avons décidé de passer à la seconde partie du projet qui consistait à remplacer les 3 capteurs à distance par un seul capteur tournant : le LIDAR. Ce capteur étant complètement nouveau pour nous, il fallait utiliser des codes disponibles pour nous aider à récupérer les valeurs lues par le LIDAR et nous donner les données les plus importantes pour la résolution de notre problème. Mais nous avons eu du mal à faire fonctionner ces codes, il a fallu les revoir un par un pour corriger certaines erreurs qui étaient dues à un problème de compatibilité entre les différentes versions du compilateur en ligne MBED. Ceci nous a fait perdre énormément de temps même si, à la fin des séances, nous arrivions à lire les données acquises par le LIDAR. Il nous aurait manqué quelques heures pour mettre en place un programme fonctionnel permettant d'utiliser le LIDAR sur notre voiture.

Finalement, nous sommes ravis d'avoir pu travailler sur ce projet qui nous a permis d'apprendre de nouvelles notions (ANNEXE 2) et nous remercions les équipes du Lense pour l'aide apportée.

Benoit Kaczmarczuk
Abdellah Elmennaoui
Noé Villemagne

Source : lense.institutoptique.fr



ANNEXE 1: Code complet

```
main.cpp X
1 #include "mbed.h"
2 #define VITESSE_MAX 1.732
3 #define VITESSE_NORMALE 1.726
4 #define VITESSE_NEUTRE 1.55
5 #define VITESSE_ARRIERE 1.49
6
7 AnalogIn capteur_A (PC_0); //gauche
8 AnalogIn capteur_B (PC_3); //centre
9 AnalogIn capteur_C (PC_2); //droite
10 PwmOut moteur_cc(PC_9);
11 PwmOut moteur_direction(PB_13);
12 Serial pc(USBTX, USBRX);
13 void affiche(double A, double B, double C);
14
15 int main()
16 {
17     pc.baud(115200);
18     int mesure_capteur_A, mesure_capteur_B, mesure_capteur_C;
19     double tension_capteur_A, tension_capteur_B, tension_capteur_C, vitesse ;
20
21     moteur_cc.period_ms(20);
22     moteur_direction.period_ms(20);
23
24     // INITIALISATION
25
26     moteur_cc.period_ms(20);
27     moteur_direction.period_ms(20);
28     moteur_cc.write(1.5/20.0);
29     wait_us(2000000);
30
31
32     while(1) {
33
34         // ACQUISITION DES MESURES
35         mesure_capteur_A = capteur_A.read_u16(); // sur 12 bits MSB
36         mesure_capteur_B = capteur_B.read_u16(); // sur 12 bits MSB
37         mesure_capteur_C = capteur_C.read_u16(); // sur 12 bits MSB
38         // 2^16=65536
39         tension_capteur_A = mesure_capteur_A / 65536.0 *5; // en V
40         tension_capteur_B = mesure_capteur_B / 65536.0 *5; // en V
41         tension_capteur_C = mesure_capteur_C / 65536.0 *5; // en V
42
43         //1/distance_A = tension_capteur_A // en V
44
45         // AFFICHAGE
46         //pc.printf("Tension_A = %lf V (%d) \r\n", tension_capteur_A, mesure_capteur_A);
47         //pc.printf("Tension_B = %lf V (%d) \r\n", tension_capteur_B, mesure_capteur_B);
48         //pc.printf("Tension_C = %lf V (%d) \r\n", tension_capteur_C, mesure_capteur_C);
49         affiche(tension_capteur_A, tension_capteur_B, tension_capteur_C);
50         // TRAITEMENT
51
52         if (tension_capteur_B > 0.7){
53             if (tension_capteur_A < tension_capteur_C){
54                 if (tension_capteur_A <1.2){
55                     moteur_cc.write(VITESSE_NORMALE/20.0);
56                     moteur_direction.write(1.1/20);
57                     pc.printf("A\r\n");
58                     wait_us(350000);
59                 }
60                 else{
61                     moteur_cc.write(VITESSE_ARRIERE/20.0); // Frein
62                     moteur_direction.write(1.265/20);
63                     wait_us(100000);
64                     moteur_cc.write(VITESSE_NEUTRE/20.0); // Neutre
65                     moteur_direction.write(1.265/20);
66                     wait_us(500000);
67                 }
68             }
69         }
70     }
71 }
```

```

67         moteur_cc.write(VITESSE_ARRIERE/20.0);      // Arrière
68         moteur_direction.write(1.265/20.0);
69         wait_us(2000000);
70
71         // réglage de marche arriere
72
73         moteur_cc.write(VITESSE_NORMALE/20.0);
74         if (tension_capteur_A < tension_capteur_C){
75             moteur_direction.write(1.1/20.0);
76             wait_us(2000000);
77         }
78         else {
79             moteur_direction.write(1.5/20.0);
80             wait_us(2000000);
81         }
82     }
83 }
84 else{
85     if (tension_capteur_C <1.2){
86         moteur_cc.write(VITESSE_NORMALE/20.0);
87         moteur_direction.write(1.5/20);
88         wait_us(350000);
89         pc.printf("C\r\n");
90     }
91     else{
92         moteur_cc.write(VITESSE_ARRIERE/20.0);      // Frein
93         moteur_direction.write(1.265/20);
94         wait_us(100000);
95         moteur_cc.write(VITESSE_NEUTRE/20.0);      // Neutre
96         moteur_direction.write(1.265/20);
97         wait_us(500000);
98         moteur_cc.write(VITESSE_ARRIERE/20.0);      // Arrière
99         moteur_direction.write(1.265/20);
100        wait_us(2000000);
101
102        // réglage de marche arriere
103
104        moteur_cc.write(VITESSE_NORMALE/20.0);
105        if (tension_capteur_A < tension_capteur_C){
106            moteur_direction.write(1.1/20.0);
107            wait_us(2000000);
108        }
109        else {
110            moteur_direction.write(1.5/20.0);
111            wait_us(2000000);
112        }
113    }
114 }
115 }
116 else{
117
118     if ((tension_capteur_A <0.4) && (tension_capteur_B <0.4)&& (tension_capteur_C<0.4)) {
119         //a=1/tension_capteur_B*0.004
120
121         moteur_cc.write(VITESSE_MAX/20.0);
122         moteur_direction.write(1.265/20.0);
123         pc.printf("avance rapide\r\n");
124         //a=1/tension_capteur_B*0.004
125     }
126     else{
127         moteur_cc.write(VITESSE_NORMALE/20.0);
128         moteur_direction.write(1.265/20.0);
129         pc.printf("avance\r\n");
130     }
131 }
132 wait_us(200000);
133 }
134
135 }
136
137
138 void affiche(double A, double B, double C)
139 {
140
141     pc.printf("A = %lf V, B = %lf V, C = %lf V ", A, B, C);
142 }

```

ANNEXE 1 : Notions travaillées (encadré rouge)

