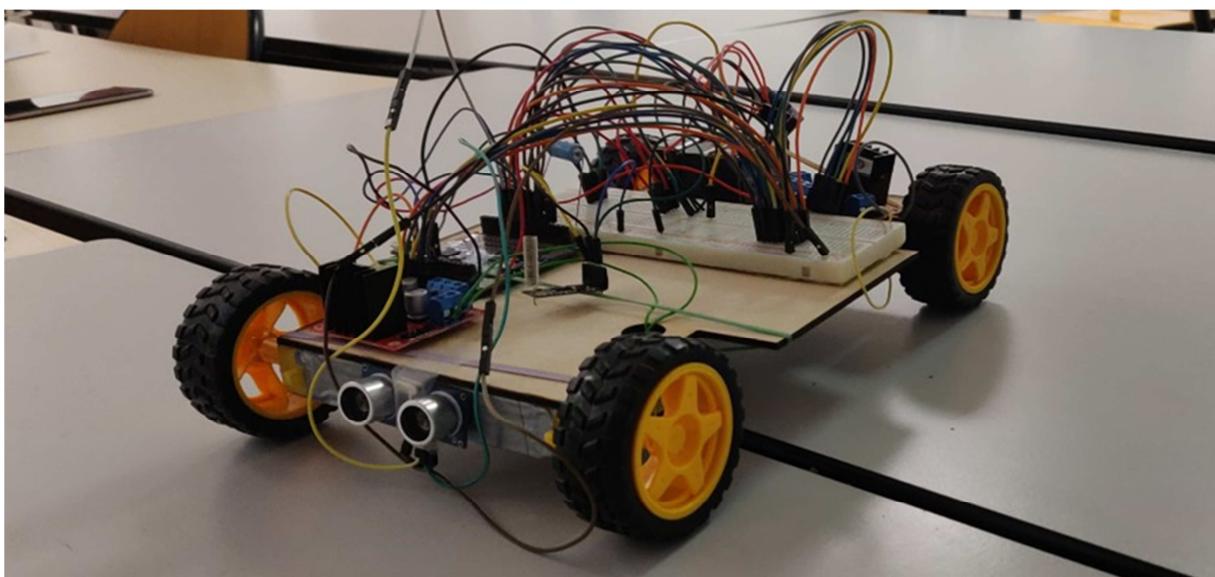




PROJET ELECTRONIQUE AVEC ARDUINO – PEIP 2

Année scolaire 2018-2019

PSYCHOKINESE



ETUDIANTS GERACI LUCAS
LEGBRE JEREMIE
LI HANWEN

ENCADRANTS MR MASSON
MR ABDERRAHMANE

REMERCIEMENTS

Nous remercions Mr Masson de nous avoir encadrés durant la réalisation de notre projet Arduino.

Également nos remerciements à Mr Abderrahmane qui nous a donné de précieux conseils concernant le fonctionnement des composants électroniques.

Enfin nous remercions la personne travaillant au Fab-Lab, qui nous a donné de nombreux conseils concernant la conceptualisation de la structure de la voiture.

TABLE DES MATIERES / TABLE OF CONTENTS

1.	Introduction et cahier des charges.....	5
2.	PLANNING	6
3.	Presentacion de Materiel.....	7
3.1	Présentation des modules	7
3.1.1	Carte Arduino uno	7
3.1.2	Moteur LN98N.....	7
3.1.3	4 Moteurs à courant continu et 4 roues.....	8
3.1.1	Module Emetteur/Recepteur HC-12.....	8
3.1.2	Accéléromètre : GY-61	9
3.1.3	Capteur de distance HC-SR04.....	9
3.1.4	Buzzer Arduino.....	10
4.	Mise en œuvre de l'ensemble des éléments.....	11
4.1	Construction de la voiture	11
4.2	Configuration de l'Accelerometre	13
4.3	Emetteur - récepteur.....	14
4.4	Configuration du capteur de distance	14
4.5	L298N et moteurs	15
5.	La Realisation	17
6.	Les principales difficultés	18
6.1	Manipulation de l'accéléromètre	18
6.2	fonctionnement du module HC-12	18
6.3	Tension des L298N.....	19
6.4	Abandon de l'idée du « gant »	19
6.5	Esthétique.	19
7.	Conclusion.....	20
8.	Références	21
9.	Programmes et Algorithmes.....	22
9.1	Emission des données	22
9.2	Réception des données.....	23

LISTE DES FIGURES

Figure 1 : Carte Arduino uno.....	7
Figure 2 : LN298N	7
Figure 3 : Moteur à courant continu.....	8
Figure 4 : Module HC-12	8
Figure 5 : Accéléromètre : GY-61	9
Figure 6 : Capteur HC-SR04	9
Figure 7 : Montage voiture.....	11
Figure 8 : Découpe du squelette.....	12
Figure 9 : Assemblage du squelette	12
Figure 10 : Exemple de branchement.....	13
Figure 11 : Simulation de mouvement de la voiture.....	14
Figure 12 : Simulation alerte de distance avec Buzzer.....	15
Figure 13 : Exemple de branchement.....	16
Figure 14 : Soudure des bornes du moteur	16
Figure 15 : Commande par interrupteur	16
Figure 16 : Rendu de la voiture, avant l'installation des derniers éléments	17
Figure 17 : test du Buzzer.....	17
Figure 18 : voiture initiale.....	19
Figure 19 : Algorithme (émetteur).....	22
Figure 20 : Algorithme (récepteur).....	23

1. INTRODUCTION ET CAHIER DES CHARGES

Le but de notre projet Arduino est de réaliser une voiture “connectée” pilotée par les mouvements de la main. Les objectifs principaux étant de pouvoir contrôler une voiture à distance et de prévenir les obstacles à l’arrière de la voiture. Pour cela nous allons utiliser des modules vus en cours et qui nous permettront de réaliser les fonctionnalités utiles à notre voiture connectée.

Un émetteur **HC-12** sera posé sur le gant et permettra d’envoyer les valeurs de l’accéléromètre à un récepteur branché à la carte Arduino sur la voiture. La voiture se déplacera en fonction de l’inclinaison de la main porteuse du gant, gant sur lequel on devra installer l’accéléromètre.

Un capteur de distance **HC-SR04**, fixé sur un **servomoteur** en rotation entre 0° et 180°, repérera des éventuels obstacles derrière la voiture et un **buzzer** s’enclenchera en guise d’avertissement : il émettra un signal dont la fréquence sera de plus en plus élevée plus l’obstacle sera proche de la voiture et celle-ci s’arrêtera automatiquement si elle est environ à moins de 2cm de l’obstacle.

Pour le fonctionnement de la voiture, on utilisera un module **L298** pour faire tourner des moteurs à **courant continu**, qui actionneront les roues ; ce module sera alimenté par 3 piles de 9 Volts.

2. PLANNING

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1																				
2																				
3																				
4																				
5																				
6																				
7																				
8																				
9																				
10																				
11																				
12																				
13																				
14																				
15																				
16																				
17																				
18																				
19																				
20																				
21																				
22																				
	<	>		Planning Projet Arduino		+														
Prêt																				

3. PRESENTATION DE MATERIEL

3.1 PRESENTATION DES MODULES

3.1.1 Carte Arduino uno

La carte Arduino Uno est basée sur un ATMega328 cadencé à 16 MHz. Ce contrôleur contient un bootloader qui permet de modifier le programme sans passer par un programmateur. Le microcontrôleur peut être programmé pour analyser et produire des signaux électriques. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires. C'est une plateforme basée sur une interface entrée/sortie simple.

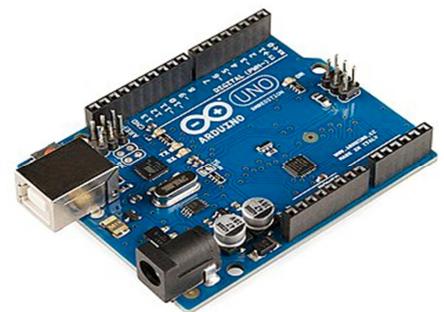


Figure 1 : Carte Arduino uno

3.1.2 Moteur LN98N



Ce module de pilotage de moteurs utilise une puce L298N (pont en H). Il pilote directement la vitesse d'un ou deux moteurs à courant continu DC dans les deux sens de rotation, avec une interface de sortie sous 5V, et fournit une alimentation de 5V. Il peut également contrôler un moteur pas à pas bipolaire.

Figure 2 : LN298N

3.1.3 4 Moteurs à courant continu et 4 roues

Un moteur électrique est constitué de deux parties électriques : le stator et le rotor. En alimentant le moteur, il se crée une interaction magnétique qui met le moteur en mouvement. Lorsqu'on inverse le sens de la tension qui alimente le moteur alors celui-ci tourne en sens inverse.

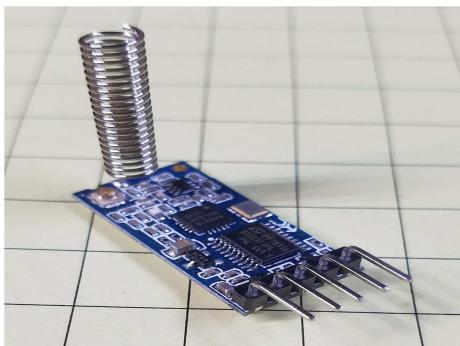
Un système de frottement permet d'alimenter le rotor: des balais frottent sur les contacts en rotation : le collecteur.

On utilise en général des moteurs à excitation constante. L'inducteur est formé d'aimants permanents. Seul le rotor est alimenté par 2 fils. Lorsqu'on inverse la polarité du moteur, il tourne dans le sens inverse.



Figure 3 : Moteur à courant continu

3.1.1 Module Emetteur/Recepteur HC-12



Le HC12 utilise un contrôleur auquel a été ajouté un contrôleur UART qui permet de transmettre des informations en utilisant un port série.

Ce contrôleur permet de modifier la puissance d'émission, le canal, la vitesse de modulation ou de le mettre en mode veille.

Figure 4 : Module HC-12

3.1.2 Accéléromètre : GY-61

L'accéléromètre GY-61 est constitué du module ADXL335 qui est un triple accéléromètre analogique, donnant une tension continue à chaque axe. Ainsi chaque axe peut mesurer une accélération de +/- 3g.

Il faut savoir que la tension à 0g est de 1.5V mais peut aller de 1.35 V à 1.65V pour les axes X et Y.

Comme la sensibilité va de 270 à 330 mV/g, un étalonnage de l'accéléromètre est nécessaire. Il suffit de mesurer les valeurs de tension pour des angles connus.

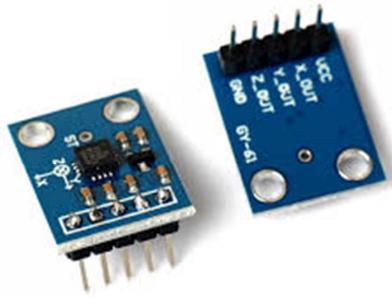


Figure 5 : Accéléromètre : GY-61

3.1.3 Capteur de distance HC-SR04

Le HC-SR04 est un module qui permet de mesurer une distance. Ce module comprend 2 haut-parleurs. L'électronique de gestion de ces haut-parleurs et de mesure de la distance est intégrée au module. Ce module comporte également 3 entrées (VCC, GND et Trig) et une sortie (Echo).



Figure 6 : Capteur HC-SR04

Le principe de ce module et le suivant : pour déclencher une mesure il faut appliquer une impulsion d'au moins $10\mu s$ sur l'entrée Tri. Le module envoie alors 8 impulsions sonores. L'onde utilisée a une fréquence de 40kHz. On obtient en sortie une impulsion dont la longueur correspond au temps que mettent les 8 impulsions pour faire l'aller/retour. Sachant que le son parcourt 340m par secondes, il est donc facile de déduire la distance

$$X (\text{cm}) = \frac{\text{temps de l'impulsion}}{2 \text{ (car aller/retour)}} * 0,0034$$

3.1.4 Buzzer Arduino



Le buzzer fonctionne en numérique. Il faudra donc le relier à une sortie digitale. Avec Arduino, l'utilisation d'un buzzer piezo à fréquence variable est très simple grâce à la fonction "tone" qui va générer la note de la fréquence souhaitée et grâce à un timer intégré au microcontrôleur, tenir la note pendant la durée souhaitée.

4. MISE EN ŒUVRE DE L'ENSEMBLE DES ELEMENTS

4.1 CONSTRUCTION DE LA VOITURE

Les pièces de la voiture ont été fabriquées au fab-lab, puis assemblées. La voiture est composée de deux pièces.

Tout d'abord, il y a le « tronc » qui est une superposition de quatre planches fines découpées au Fab-Lab sur lesquelles sont collés les moteurs ainsi que les roues.

Ensuite, il y a le plateau, où sont positionnés la carte Arduino , les modules L298N, le HC-12 récepteur et la breadboard.

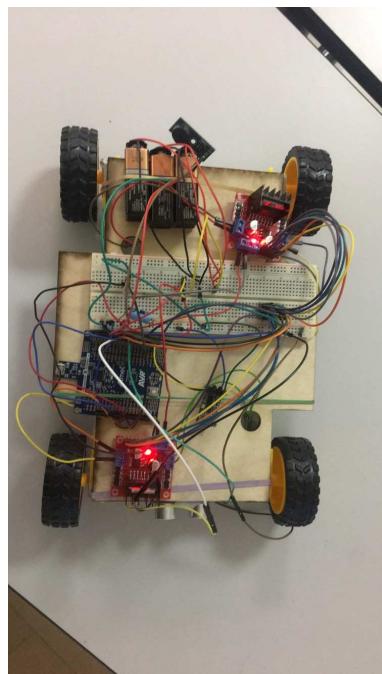
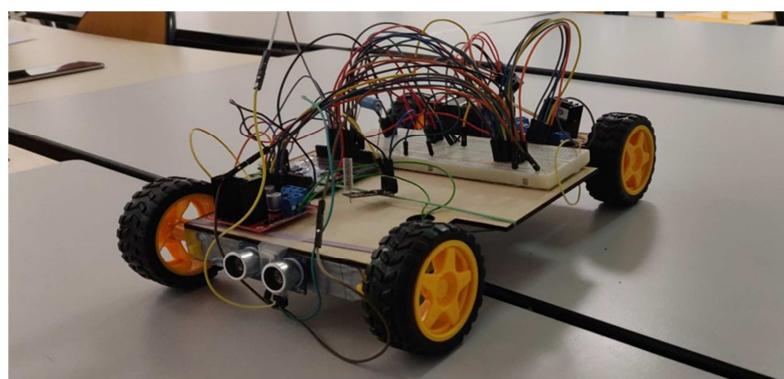


Figure 7 : Montage voiture



Avant le montage nous décidons d'avoir une voiture assez grande pour avoir la place d'installer tout le matériel qui sera collé sur celle-ci. Au fab lab nous avons pu couper les planches nécessaires à la fabrication du squelette de la voiture. Voici une photo du squelette de la voiture, avec les moteurs et roues collés :

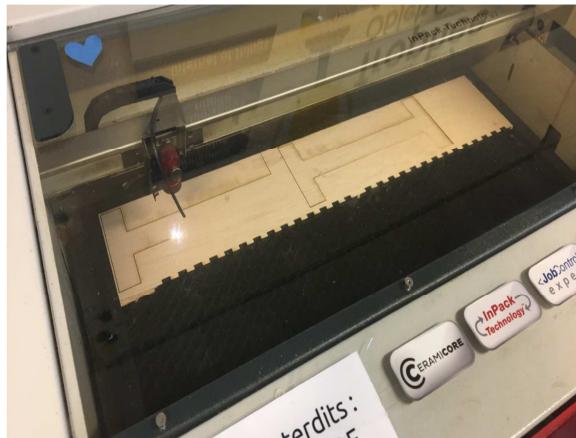


Figure 8 : Découpe du squelette

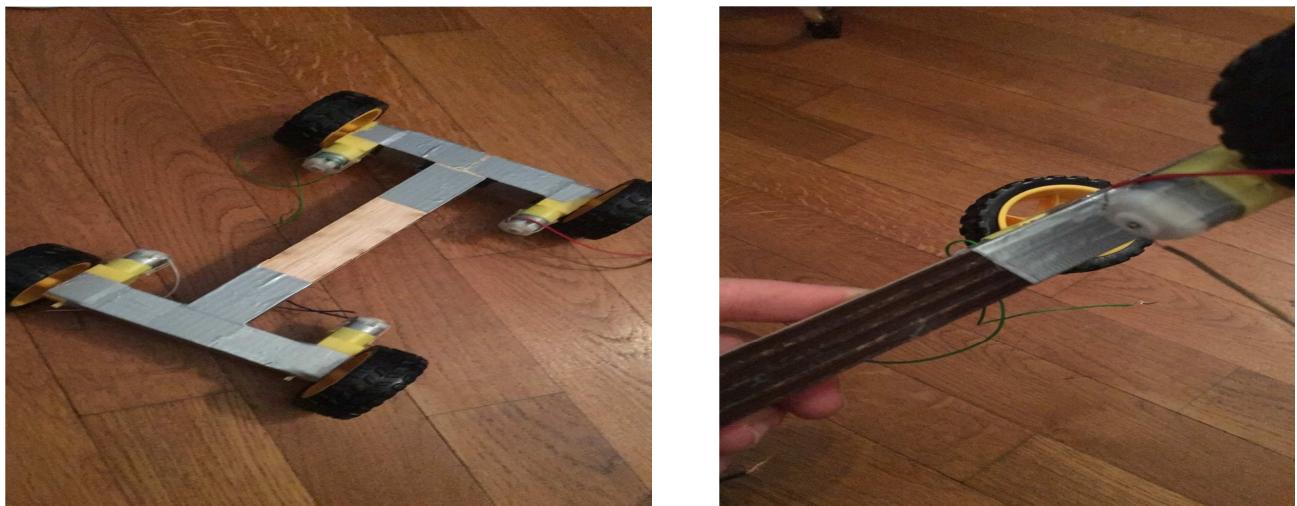


Figure 9 : Assemblage du squelette

Au-dessus de cela, nous avons découpé une planche qui nous permis de poser la breadboard, les capteurs de distances, la carte arduino, les servomoteurs etc.

Le squelette de la voiture est constitué de quatre planches découpées, collées afin d'obtenir une épaisseur convenable (cf Figure 8).

4.2 CONFIGURATION DE L'ACCELEROMETRE

Triple accéléromètre : tensions pour chaque axe X,Y, Z

Cependant nous avons besoin uniquement des axes X et Y

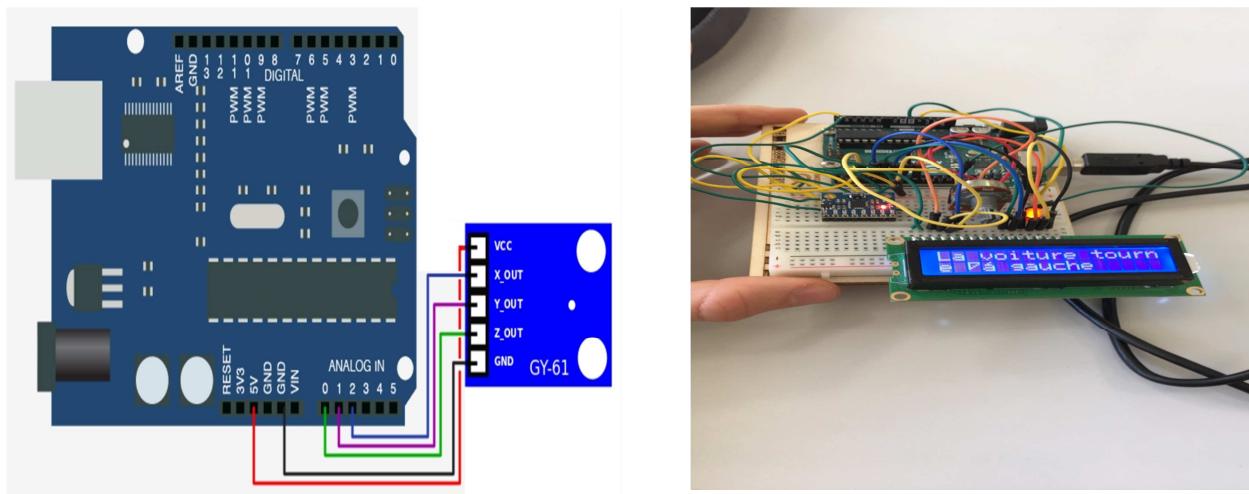


Figure 10 : Exemple de branchement

Plusieurs tentatives ont été nécessaires pour arriver à programmer l'accéléromètre GY-61.

Les premiers essais donnaient un affichage de valeurs négatives constantes (-35, 26°) pour l'axe des x et l'axe des y. Un problème de soudure était à l'origine du problème. En ayant refait la soudure, l'accéléromètre affichait des valeurs normales. De plus nous avons développé un simulateur des actions de la voiture. A l'aide d'un écran LCD et d'un potentiomètre, les différentes actions de la voiture ont été affichées. L'axe des x va déterminer si la voiture avance ou recule, tandis que l'axe des y va déterminer si la voiture tourne à gauche ou à droite.

Voici le lien d'une vidéo youtube réalisée : <https://youtu.be/rGGyJs0EiF4>

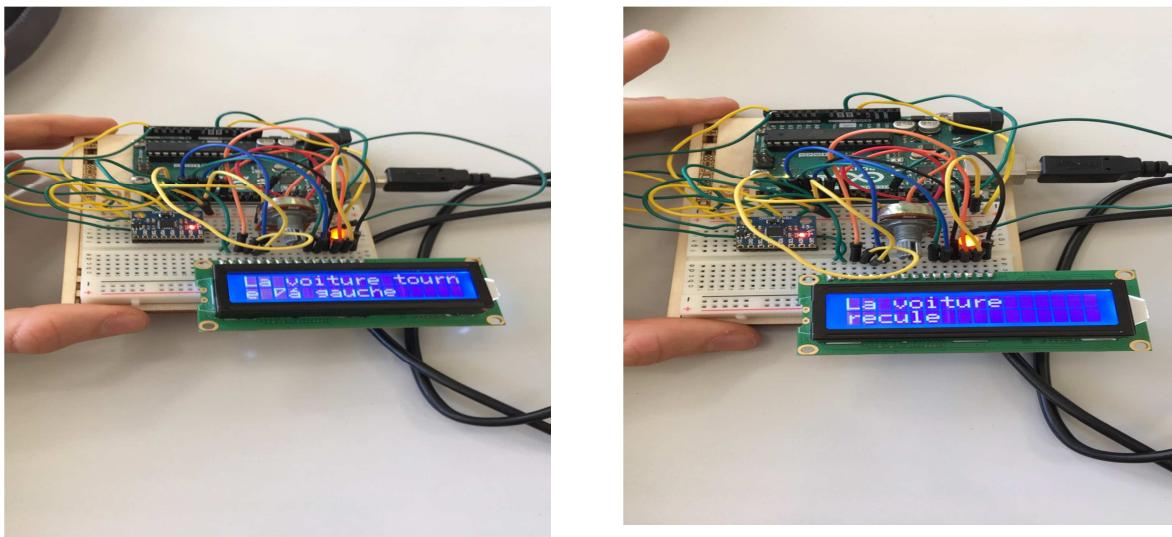


Figure 11 : Simulation de mouvement de la voiture

4.3 EMETTEUR - RECEPTEUR

Le but du projet est d'envoyer les valeurs de l'accéléromètre. Les deux modules HC-12 communiquent entre eux par l'intermédiaire de la fonction « HC12.print » ; après avoir configuré les commandes AT, la transmission des données se fait. L'émetteur envoie les valeurs de l'accéléromètre au récepteur qui les lit et les affiche sur le moniteur série. Le récepteur va ensuite utiliser les valeurs reçues pour faire marcher les moteurs, leur puissance dépendant de l'angle de l'accéléromètre.

Après avoir implanté les programmes des moteurs donnés par Mr Masson nous sommes arrivés à envoyer les valeurs de l'accéléromètre, et à faire évoluer ces valeurs.

4.4 CONFIGURATION DU CAPTEUR DE DISTANCE

Nous voulions reproduire le bruit sonore que fait une voiture lorsque l'arrière de la voiture s'approche d'un mur. C'est à dire que si la voiture s'approche d'un obstacle, elle va émettre un « bip » sonore qui aura une fréquence plus ou moins élevée selon la distance à laquelle se trouve la voiture par rapport à l'obstacle. A l'aide d'un buzzer et d'un capteur de distance, nous pouvons réaliser cette fonction. Nous avons fait

une simulation des capteurs de distance qui seront positionnés sur l'avant et l'arrière de la voiture. Lorsque la voiture est à une distance de moins de 10cm devant un obstacle, la voiture s'arrête. Comme la voiture n'était pas prête, afin de pouvoir tester la fonctionnalité désirée, nous avons fait une simulation avec une LED.

Un premier programme avait été effectué, celui-ci émet un « bip » lorsque la distance que reçoit le capteur de distance était inférieure à 10 cm.

Il a été ensuite modifié pour que le buzzer émette des « bip » selon des fréquences différentes. Enfin, lorsque la voiture va reculer, les feux arrière seront allumés.

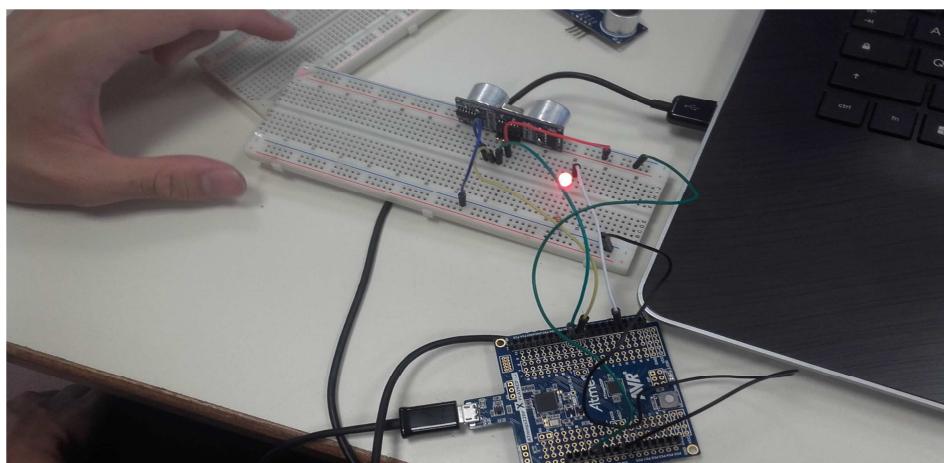


Figure 12 : Simulation alerte de distance avec Buzzer

4.5 L298N ET MOTEURS

L298N : quadruple demi-pont en H

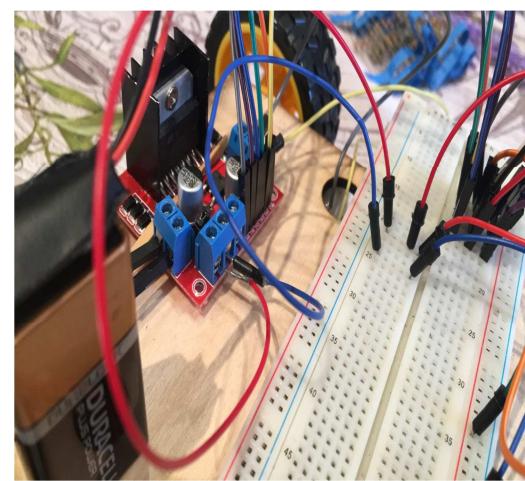
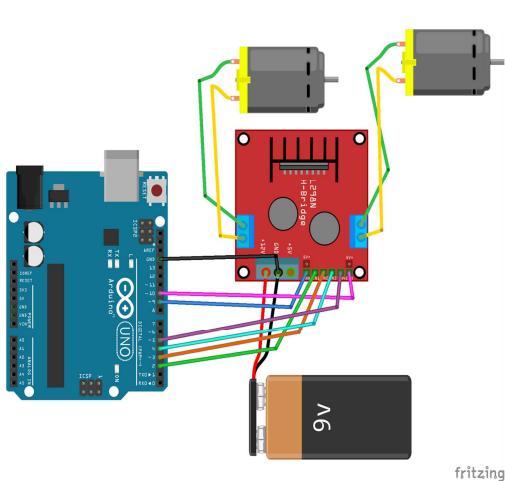
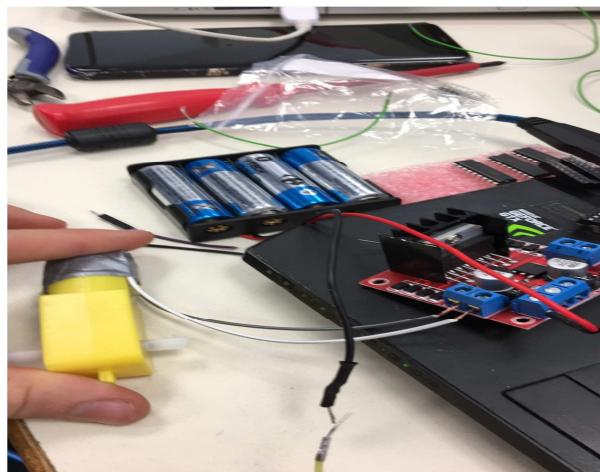
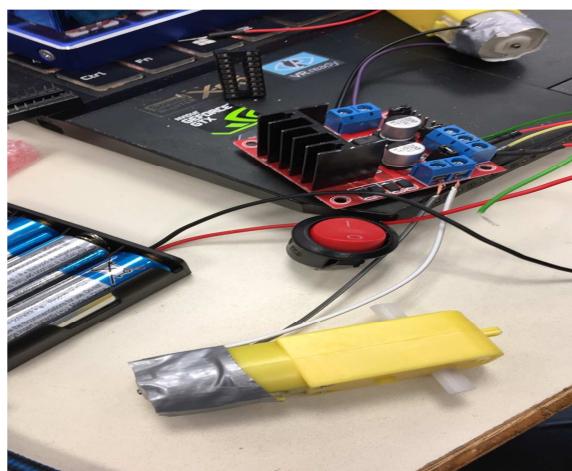


Figure 13 : Exemple de branchement

A l'aide du cours nous avons réussi à commander 2 moteurs. Tout d'abord nous avons commencé par souder les bornes de chaque moteur (comme on peut le voir dans l'image ci-dessous), puis la partie branchement a suivi. On a réalisé cette opération deux fois afin d'avoir nos quatre roues et moteurs.

**Figure 14 : Soudure des bornes du moteur**

Un interrupteur sera utilisé pour commander les moteurs.

**Figure 15 : Commande par interrupteur**

5. LA REALISATION

- ➡ Collage et fixation des premiers éléments sur le plateau.
- Fixation des modules L298N, de la breadboard et de la carte Arduino Uno.

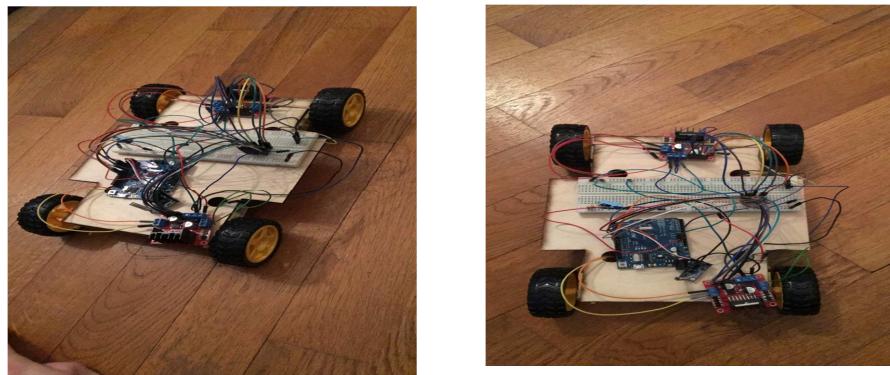


Figure 16 : Rendu de la voiture, avant l'installation des derniers éléments

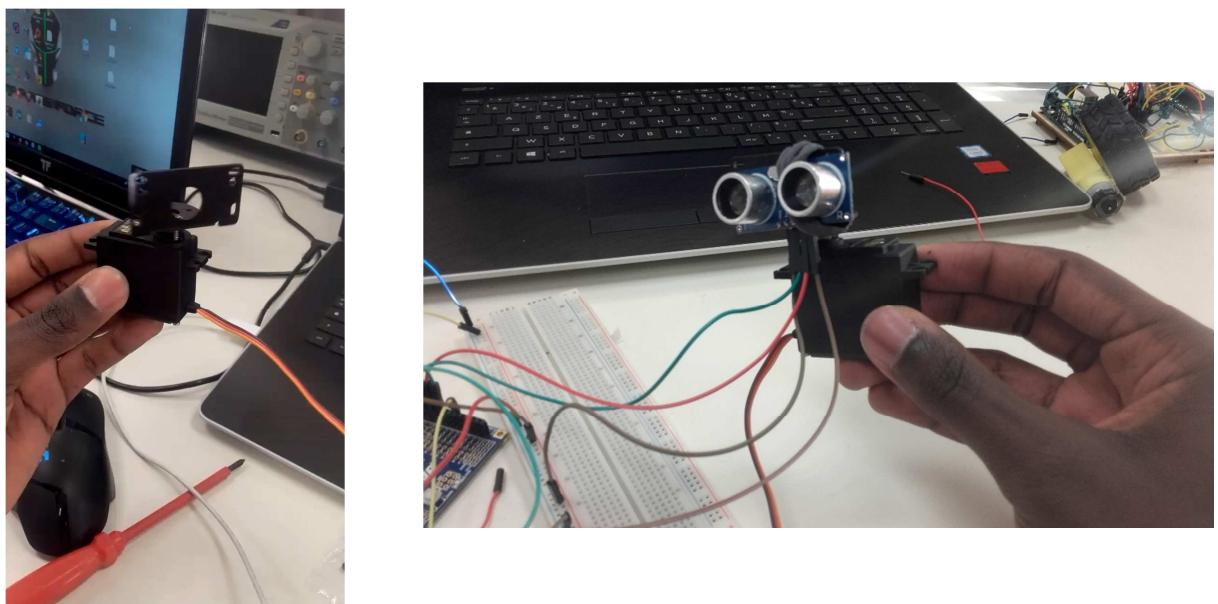


Figure 17 : test du Buzzer

6. LES PRINCIPALES DIFFICULTES

6.1 MANIPULATION DE L'ACCELEROMETRE

A l'aide de programmes trouvés sur internet, et les programmes déjà implantés dans le logiciel Arduino, nous avons réussi à afficher sur le moniteur série les coordonnées de l'accéléromètre selon les trois axes. Cependant, lorsque nous essayons de faire varier les coordonnées de l'accéléromètre en le faisant pivoter, nous n'avons pas obtenu de changements de variations pour les angles, ce qui est au cœur de notre projet. De plus, lorsque la première difficulté a été résolue, nous avions des variations de la valeur des angles de l'accéléromètre en ne bougeant pas. C'est pour cela que le contrôle de la voiture était parfois hasardeux.

6.2 FONCTIONNEMENT DU MODULE HC-12

Comprendre le fonctionnement du module HC-12 au niveau du code et l'utilisation des librairies.

Au début du projet, nous avions choisi le RF 433mhz. La principale tâche que nous devions réussir était d'envoyer les données de l'accéléromètre avec le RF 433mhz. Mais nous avons eu du mal à communiquer entre l'émetteur et le récepteur. Nous avons essayé plusieurs librairies, mais rien ne fonctionnait. C'est pour cela que nous avons décidé de changer d'émetteur/récepteur, pour utiliser un HC-12.

En utilisant la librairie Software Serial, nous avons réussi à communiquer entre deux arduinos (à l'aide des moniteurs séries). C'est à dire que nous avons réussi à écrire sur un moniteur série, et recevoir le message sur l'autre moniteur série. Pour réaliser le projet, nous devions envoyer les données de l'accéléromètre, ce qui fût une tâche plus compliquée. Après plusieurs tentatives de programmation, nous sommes arrivés à afficher les valeurs de l'angle x, sur le moniteur série (de l'arduino qui a le récepteur), uniquement lorsqu'on envoie un caractère quelconque sur le moniteur série (de l'arduino qui a l'émetteur). Nous n'arrivions pas au but puisque nous devions envoyer des données en continu sans avoir à envoyer un caractère. En plus de cela, la valeur que l'on recevait restait fixe. Même en bougeant l'accéléromètre nous n'arrivions pas à afficher d'autres valeurs. Ce qui était problématique.

Nous sommes enfin arrivés à nos fins, après plusieurs tests, plusieurs tentatives de

programmation, grâce au programme donnés par Mr Masson.

6.3 TENSION DES L298N

Difficulté de trouver le voltage et ampérage à appliquer au niveau des L298N (soit trop élevé, soit trop faible).

6.4 ABANDON DE L'IDEE DU « GANT »

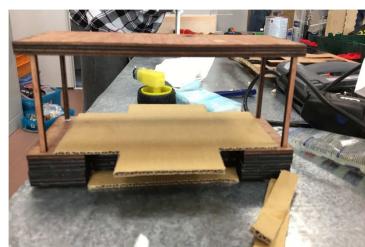
Notre projet initial était de commander notre voiture avec un gant. Pour cela il fallait connecter un émetteur et un récepteur RF 433MHz afin d'établir une communication. Nous avions pour but d'établir une communication entre l'émetteur (qui sera positionné sur le gant) et le récepteur (qui sera positionné sur la voiture). Nous avons réussi à éteindre et allumer une LED à l'aide de RF 433, puis nous avons essayé de réaliser un programme qui permet d'envoyer les valeurs d'un accéléromètre d'une carte arduino à une autre. En réalisant ceci, cela nous permettra de piloter la voiture. Nous avons décidé de répartir le travail en deux.

- ✚ Partie émetteur (gant)
- ✚ Partie récepteur (voiture).

6.5 ESTHETIQUE.

L'esthétique de la voiture nous a fait perdre un peu de temps dans la réalisation de notre projet. En effet l'idée de départ étant de réaliser la voiture avec trois plaques. Le « tronc » où la majorité des composants électroniques seront positionnés, et où les quatre roues seront fixées. Un deuxième plateau, où seront positionnés les capteurs arrière, le servomoteur et le capteur à l'avant. Et enfin, un troisième plateau au-dessus du deuxième plateau surmonté par 4 « pics ». Mais le tout n'était pas très stable et la voiture était très haute, le résultat ne ressemblait pas du tout à une voiture. Nous avons donc réfléchi à une autre solution.

Figure 18 : voiture initiale



7. CONCLUSION

Nous nous sommes heurtés à de nombreuses difficultés qui nous ont obligés à modifier notre cahier des charges. Le projet a donc été simplifié à la commande d'une voiture.

Certaines difficultés rencontrées ont été surmontées et d'autres contournées. Ces difficultés proviennent de problèmes de compatibilité de librairies, de conversion de variables et de librairies, mais aussi d'organisation du projet. Ces difficultés ont une pour conséquence de nous faire perdre du temps et par manque de temps nous avons été contraints à modifier notre cahier des charges.

8. REFERENCES

- <https://youtu.be/rGGyJs0EiF4>
- <https://youtu.be/C9oaY5EBLyM>

9. PROGRAMMES ET ALGORITHMES

9.1 EMISSION DES DONNEES

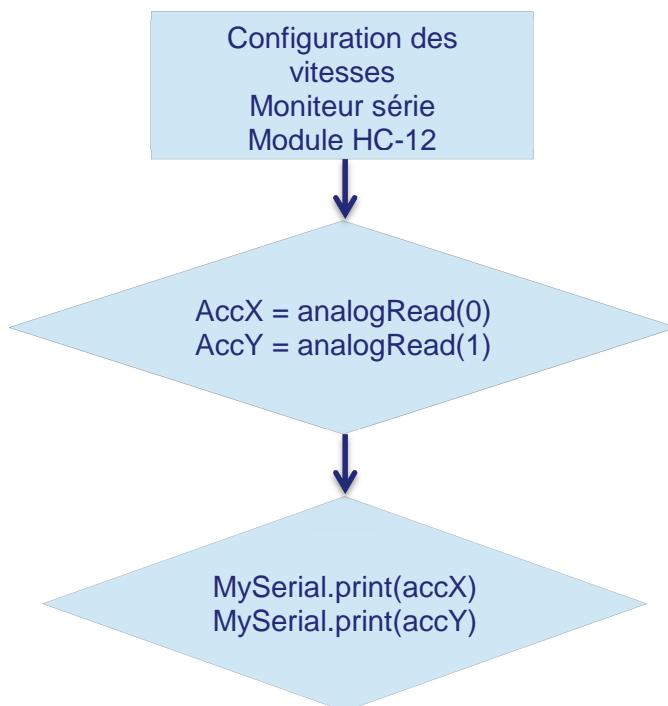


Figure 19 : Algorithme (émetteur)

9.2 RECEPTION DES DONNEES

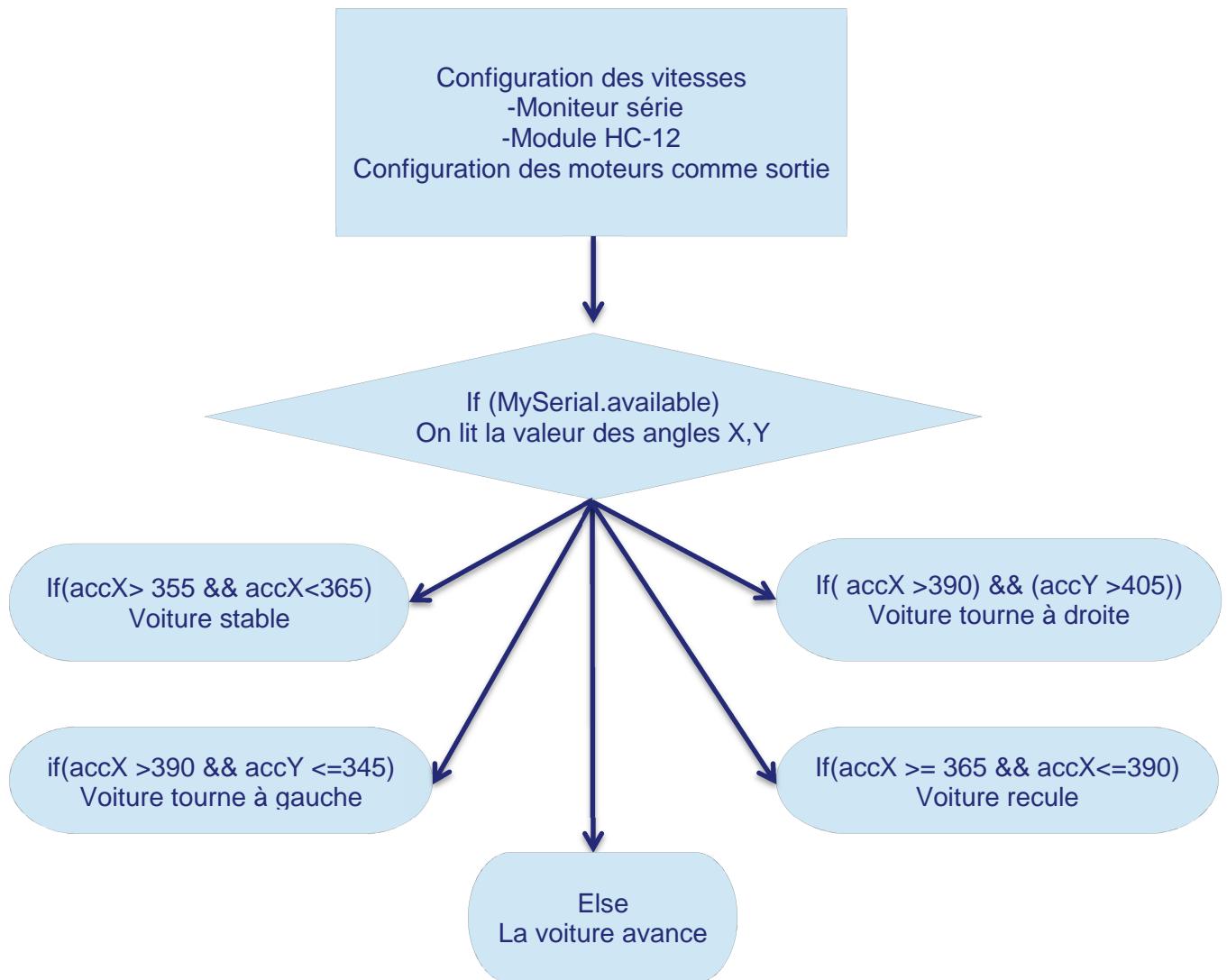


Figure 20 : Algorithme (récepteur)