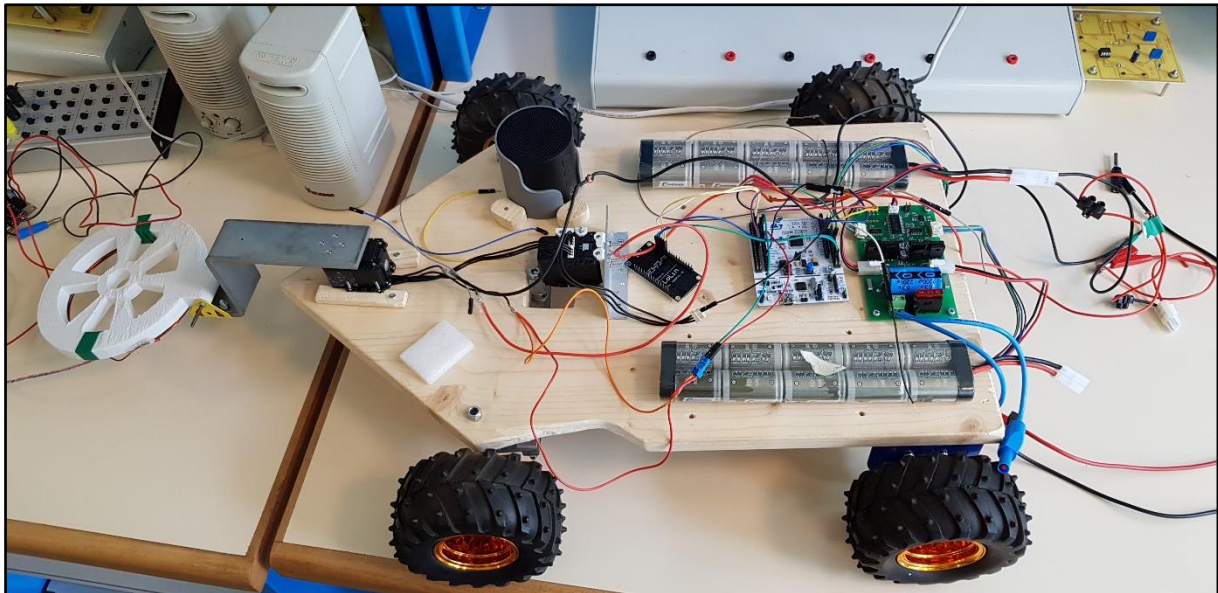


ROBOT DEMINEUR

**Etudiants :****Charles DUCHEMIN****Jonathan CLOAREC****Omar BENNANI****Hugo GEORGES****Kariès CHARLESTON****Tanguy LECLERC****Enseignant-responsable du projet :****M. Fabrice DELAMARE**

Date de remise du rapport : **18/06/2018**

Référence du projet : **STPI/P6/2018 – 21**

Intitulé du projet : **Robot mobile de détection de mines piloté par WIFI à partir d'une tablette.**

Type de projet : **Expérimental**

Objectifs du projet (10 lignes maxi) :

Réalisation d'un robot piloté par WIFI.

Conception du robot sur SolidWorks.

Programmation d'un site Web sous Arduino en HTML.

Programmation en C et dialogue entre les différentes cartes

Mots-clefs du projet (4 maxi) :

SolidWorks, Programmation en C, Site Web, Cartes électroniques

TABLE DES MATIERES

1. Introduction	5
2. Méthodologie / Organisation du travail	6
3. Travail réalisé et résultats	7
3.1. Partie mécanique	7
3.1.1. Modèle Solidworks	7
3.1.2. Montage du robot	8
3.2. Partie informatique	9
3.2.1. Arduino, Nucléo et MD49	9
3.2.2. Carte wifi	11
3.2.3. Les servo-moteurs	12
3.2.4. Le détecteur de métaux.....	13
4. Conclusions et perspectives.....	14
5. Bibliographie	16
6. Annexes.....	17
6.1. Programme principal	17
6.2. Programme wifi	21
6.3. Propositions de sujets de projets (en lien ou pas avec le projet réalisé)	27

1. INTRODUCTION

Étudiants en 2ème année et ayant choisis des thématiques différentes, passionnés par la mécanique pour certains, par l'informatique pour d'autre ou alors ayant tout simplement l'envie de fabriquer un robot, nous nous sommes retrouvés dans le cadre de notre projet de physique (P6) afin de réaliser un robot démineur.

L'objectif final de ce projet était de pouvoir construire un robot piloté en WIFI à l'aide de notre téléphone, et qui nous aiderait à balayer un terrain afin de trouver d'éventuelles mines dissimulées, ou alors de simples métaux. Celui-ci est réalisé sous la supervision de M. Fabrice Delamare, qui s'assure du bon déroulement du travail et qui est présent pour nous aider lors des difficultés.

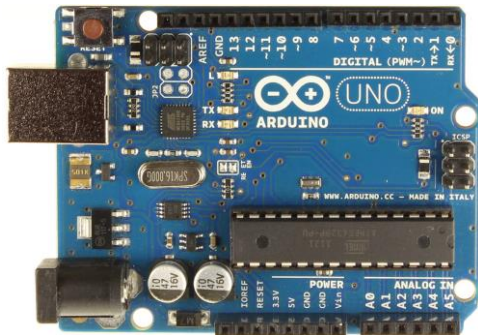
Il nous était évident que la tâche ne serait pas facile puisque de nombreuses notions étaient nouvelles pour la majorité d'entre nous. Mais la volonté de construire ce robot, les conseils donnés par M. Delamare ainsi que les informations présentes sur internet nous ont permis de nous approcher de nos objectifs. Cependant, nous n'avons pas réussi à tous les atteindre.

Dans un premier temps, nous aborderons l'organisation du projet, puis nous parlerons de l'aspect mécanique de notre robot avant d'analyser, dans une dernière partie, la structure informatique de notre projet.

2. METHODOLOGIE / ORGANISATION DU TRAVAIL

Il est clair qu'une bonne organisation était nécessaire afin de mener à bien ce projet. C'est pourquoi, dès le premier jour où nous nous sommes rencontrés, nous avons réalisé deux groupes. Le premier composé de 2 personnes et l'autre de 4 et ce en fonction des connaissances et affinités de chacun. En effet, certains avaient plus de connaissances en mécanique, tandis que d'autres n'étaient pas contre le fait de participer au codage informatique.

Ainsi, Jonathan et Kariès se sont chargés de la CAO ainsi que de l'aspect mécanique. Omar, Charles, Tanguy et Hugo s'attaquèrent quant à eux à la compréhension et au codage des différentes cartes. Il fut décidé que nous prendrions une Arduino UNO R3 comme carte de commandement. Charles ayant déjà des connaissances dans ce langage s'occupa de l'étude de la carte MD49, pendant que les trois autres membres du groupe apprennent la forme du langage Arduino.



Arduino UNO



MD49

Par la suite, le codage de la MD49 fut laissé à Omar, Hugo et Tanguy, tandis que Charles étudiait un nouveau type de composants, à savoir les servomoteurs. C'est alors que compte tenu d'un composant manquant au niveau de l'Arduino, il nous a fallu changer de carte de développement après 4 semaines de travail, rendant ce dernier pratiquement inutile pour notre projet. Nous avons finalement utilisé une Nucleo F401RE. Aussi, suite à de grandes difficultés, tout d'abord dans l'établissement de la communication entre l'Arduino et la MD49 puis entre la Nucleo et la MD49, nous avons abandonné nos tâches pré-établies et avons décidé de nous regrouper afin de résoudre ce problème de communication qui nous empêchait d'avancer pensant que à plusieurs nous trouverions plus rapidement les solutions à nos problèmes. L'évènement qui nous a poussé à changer de carte mère sera expliqué plus en profondeur dans la partie informatique.

3. TRAVAIL REALISE ET RESULTATS

3.1. Partie mécanique

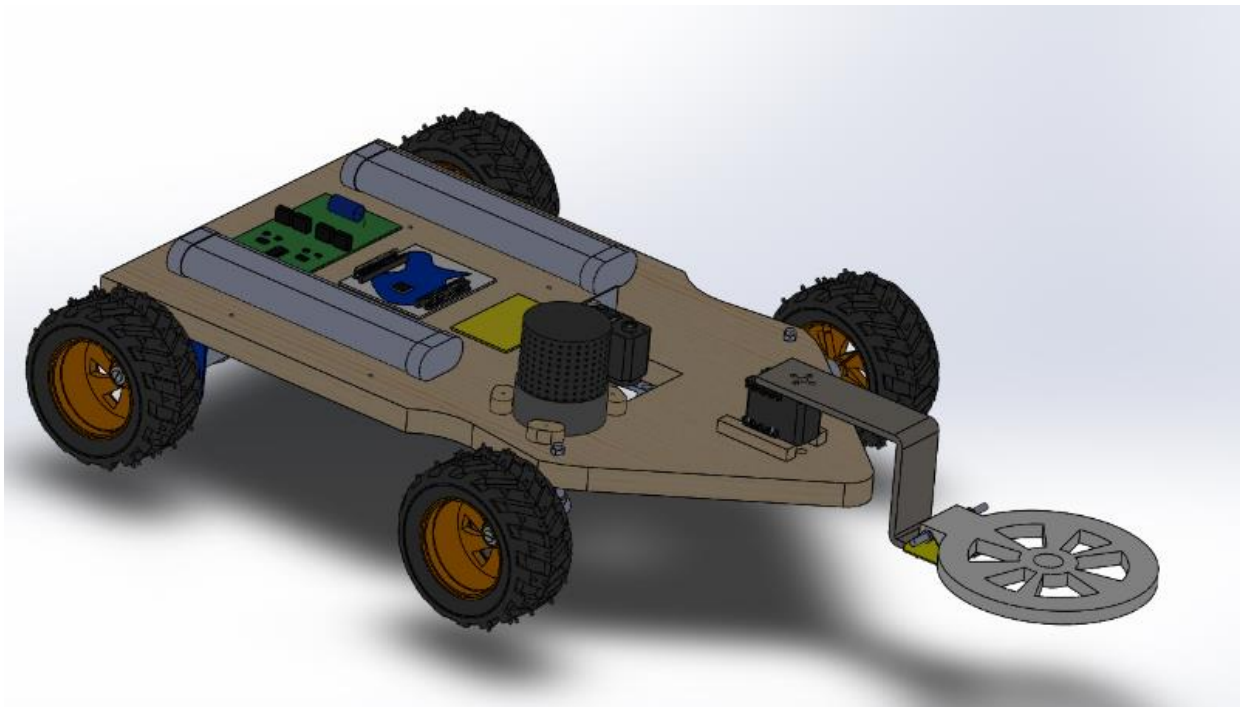
3.1.1. *Modèle Solidworks*

Lors de ce projet, un modèle CAO a été créé. Les pièces modélisées sont principalement des composants fournis au départ du projet et déjà présents sur l'ancien modèle. Cela a demandé du travail en dehors des séances mais de manière raisonnable. La complexité de certaines pièces, notamment les roues, nous a permis de progresser sur le logiciel de CAO SolidWorks présenté au lycée. Nous sommes devenus plus efficaces et avons découvert de nouvelles fonctions.

Ce modèle CAO a ensuite permis de préparer l'assemblage du robot en lui-même. La forme générale de la planche a été vérifiée sur SolidWorks dans l'objectif de définir la meilleure géométrie. Nous avons aussi pu visualiser les composants afin de les placer précisément optimiser le placement des pièces. Cela nous a permis de gagner du temps car nous avons seulement 1h30 par semaine et savoir où placer les composants avant les séances fut bien utile. Ce modèle nous permet également de bien visualiser le robot sous tous ses angles, voire même en cachant certains composants si besoin.

Nous n'avons pas eu besoin d'imprimer des pièces pour ce projet. Nous avons préféré les travailler à partir du bois mais le modèle CAO aurait pu nous être utile pour modéliser certaines pièces complexes si besoin.

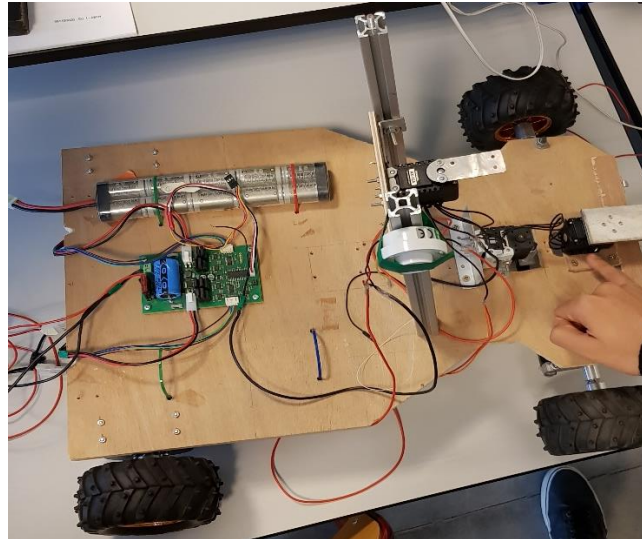
Finalement, il ne manquera que les fils sur ce modèle car ils n'étaient pas importants pour le montage et que cela était trop compliqué à modéliser et non indispensables.



Robot modélisé sur SolidWorks

3.1.2. Montage du robot

Au début de ce projet, nous disposions de la construction réalisée par un groupe précédent. Différents points ne nous convenaient pas dans ce montage notamment au niveau du choix du bois. Il s'agissait en effet d'une planche de bois compressé fragile à l'usinage. Les traces de recherches du projet précédent étaient visibles sous la forme de nombreux trous inutilisés



Robot du groupe précédent

Considérant que le montage n'était pas achevé, nous avons préféré reprendre le projet depuis le début, tout en notant les choix effectués par le groupe précédent afin de d'utiliser leurs expériences.

Nous avons en particulier fait en sorte d'éviter la présence de trous supplémentaires et avons utilisé le même montage pour le train avant. Jonathan s'est occupé de réaliser les pièces en CAO afin de réfléchir aux plans avec un modèle sous les yeux. Il a ensuite participé à l'usinage de la partie mécanique avec Kariès.

Nous avons commencé par décider de la forme du châssis que nous avons choisis en forme de flèche pour un effet d'aérodynamisme. Bien que le robot ne soit pas destiné à un roulage rapide, cela nous semble plus esthétique.

Pour les positionnements des éléments les plus imposant sur le châssis, nous sommes partis de l'équilibrage conventionnel des véhicules de modélisme. A savoir les éléments les plus lourds par symétrie à l'axe médian et, si possible, également une répartition entre les trains de manière à obtenir un centre de gravité légèrement décaler sur l'arrière. Comme cela l'adhérence de propulsion est assurée (la transmission passant exclusivement par les roues arrières) et libérer le train avant afin d'optimiser l'adhérence en tournant même sur terrain glissant.

La place des moteurs était évidente au niveau de l'essieu arrière. Les accus étant les éléments les plus lourds, ils ont été placés de chaque côté du châssis selon le premier des deux équilibres précisés précédemment. Tous les autres éléments sont placés sur l'axe médian. Sur l'arrière, dans l'espace entre les accus, sont placées les cartes électroniques. Le servo de direction et celui contrôlant le bras de détection sont sur l'avant. Le de ces deux éléments combinés à celui du détecteur très en avant de l'essieu avant suffisent à la mise en place du deuxième des équilibres précisé ci-dessus.

3.2. Partie informatique

3.2.1. Arduino, Nucléo et MD49

Comme nous en avons parlé brièvement précédemment, nous avons rencontré de grandes difficultés dans la communication entre la carte de développement et le driver des deux moteurs.

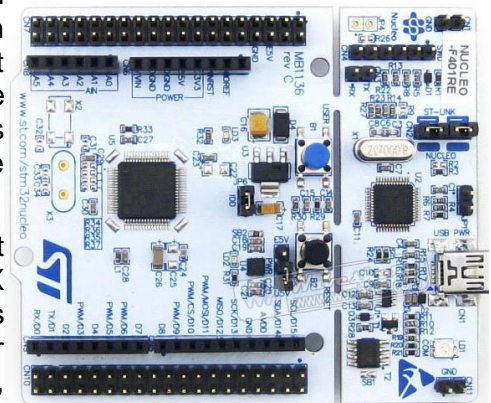
Initialement, nous avons une Arduino Uno R3. C'était la première fois que nous avons à coder une telle carte de développement, et ce en utilisant le langage C/C++ . Nous étions jusqu'alors habitués à coder uniquement en pascal. A l'aide de nombreuses recherches sur internet, des cours qui s'y trouvaient ou d'exemples de codes, nous avons réussi à faire tourner les roues de notre robot avec l'aide de notre professeur. Une partie du programme est présenté ci-dessous.



Carte Arduino Uno R3

Malheureusement, la carte Arduino Uno R3 possède un inconvénient. Pour contrôler les servomoteurs il faut un composant électronique que notre professeur nous pouvait nous obtenir. C'est pourquoi il a décidé de nous fournir une nouvelle carte, la Nucleo F401RE. Cette carte ne devait pas nous poser de soucis puisque le langage est un peu similaire et beaucoup de projets sur internet l'utilisait.

La première grande difficulté était de créer un port série. En effet, le dialogue se faisait à l'aide des PIN RX et TX de la nucléo. Mais ces PINS étant utilisés par l'USB que nous connections à l'ordinateur pour le code, il a fallu en trouver d'autres compatibles avec notre mode de communication, c'est à dire en série.



Carte Nucleo F401RE

Après plusieurs tentatives et assistés par notre enseignant, nous avons finalement réussi à communiquer avec un port série de la carte Arduino et celui de la MD49.



Un autre problème survint également lors du changement de carte : programmer la discussion entre la nucléo et la MD49. Ce problème nous a fait perdre une grande quantité de temps au cours de projet. Au cours de nos nombreuses essais de communications , nous avons constaté avec notre encadrant que la nucléo en notre possession était défectueuse. Nous n'avons malheureusement pas pu cerner l'origine du défaut, la carte ayant été utilisé par d'autres étudiants.

Carte MD49

Finalement, en changeant de carte et en comparant minutieusement les branchements et le code à Arduino, nous sommes enfin parvenus à faire tourner les moteurs principaux à 1 mois de la fin du projet.

Un extrait du code utilisé est consultable ci-dessous :

```

30
31 void avancer() //10
32 {
33     md49_putc(CMD);
34     md49_putc(SET_ACCEL);
35     md49_putc(0x00);
36     md49_putc(CMD);
37     md49_putc(SET_SPEED1);
38     md49_putc(0xE6);
39     md49_putc(CMD);
40     md49_putc(SET_SPEED2);
41     md49_putc(0xE6);
42 }
43 void avancerdroite() //11
44 {
45     md49_putc(CMD);
46     md49_putc(SET_SPEED1);
47     md49_putc(0xFF);
48     md49_putc(CMD);
49     md49_putc(SET_SPEED2);
50     md49_putc(0xE6);
51 }
52 void avancergauche() //12
53 {
54     md49_putc(CMD);
55     md49_putc(SET_SPEED2);
56     md49_putc(0xFF);
57     md49_putc(CMD);
58     md49_putc(SET_SPEED1);
59     md49_putc(0xE6);

```

Les commandes envoyées à la MD49 se font à l'aide de code du type 0xXX, comme illustré dans le tableau suivant :

command	Name	Bytes sent to MD49	Bytes returned by MD49	Description
0x21	GET SPEED 1	2	1	returns the current requested speed of motor 1
0x22	GET SPEED 2	2	1	returns the current requested speed of motor 2
0x23	GET ENCODER 1	2	4	motor 1 encoder count, 4 bytes returned high byte first (signed)
0x24	GET ENCODER 2	2	4	motor 2 encoder count, 4 bytes returned high byte first (signed)
0x25	GET ENCODERS	2	8	returns 8 bytes - encoder1 count, encoder2 count
0x26	GET VOLTS	2	1	returns the input battery voltage level
0x27	GET CURRENT 1	2	1	returns the current drawn by motor 1
0x28	GET CURRENT 2	2	1	returns the current drawn by motor 1
0x29	GET VERSION	2	1	returns the MD49 software version
0x2A	GET ACCELERATION	2	1	returns the current acceleration level
0x2B	GET MODE	2	1	returns the currently selected mode
0x2C	GET VI	2	3	returns battery volts, motor1 current and then motor2 current
0x2D	GET ERROR	2	1	returns a byte within which the bits indicate errors on the MD49
0x31	SET SPEED 1	3	0	set new speed1
0x32	SET SPEED 2 / TURN	3	0	set new speed2 or turn
0x33	SET ACCELERATION	3	0	set new acceleration
0x34	SET MODE	3	0	set the mode
0x35	RESET ENCODERS	2	0	zero both of the encoder counts
0x36	DISABLE REGULATOR	2	0	power output not changed by encoder feedback
0x37	ENABLE REGULATOR	2	0	power output is regulated by encoder feedback
0x38	DISABLE TIMEOUT	2	0	MD49 will continuously output with no regular commands
0x39	ENABLE TIMEOUT	2	0	MD49 output will stop after 2 seconds without communication

Tableau des commandes de la MD49

Avant chaque commande envoyée, il est nécessaire d'envoyer le code 0x00 pour que la MD49 s'apprête à recevoir de l'information. Par la suite on envoie « l'objectif » qu'on lui donne, par exemple SETSPEED1, par le biais de la commande 0x31. On définit finalement la vitesse de rotation du moteur. Dans notre cas, nous utilisons maximum 225 en marche avant, 128 pour l'arrêt et 0 comme pour la marche arrière

3.2.2. Carte wifi

Parmi les fonctionnalités importantes du robot se trouve la capacité de le contrôler à distance. S'approcher d'un robot démineur pendant ses opérations n'est en effet guère conseillé pour la santé !

Le support de dialogue le plus efficace est le wifi. Il nous a donc fallu coder une carte Wifi pour servir d'intermédiaire avec la carte Nucléo et c'est Charles qui s'est attelé à cette tâche.

Nous pensions à l'origine utiliser le module esp8266 pour parler avec la carte Nucléo mais notre renseignant référent nous a remis sur le bon chemin après 2 semaines de recherches.

Nous utiliserons donc une Node Mcu 3V qui se code en Arduino. Il faut cependant installer des modules additionnels au logiciel Arduino de base.

M. Delamare nous a indiqué les adresses pour bien nous renseigner sur son fonctionnement et nous a procurés des codes d'utilisations permettant de connecter la carte à un réseau mais aussi comment coder des sites WEB fonctionnels en HTML.

Il nous faudra une importante quantité de travail personnel (plus de quinze heures en dehors du créneau de P6) pour que finalement notre carte Wifi crée son propre réseau. Nous avons ensuite conçu le site WEB qui servira au contrôle et vérifiés les branchements avec la carte Nucléo.

3.2.3. Les servo-moteurs

La partie concernant les servomoteurs ne fût pas le principal problème de la section « codage » de ce projet. En effet, la documentation fournie sur internet était suffisamment claire pour que les branchements et les premières fonctions soient établies en un temps record.



Servo-moteur AX-12A

Néanmoins, nos codes ne fonctionnaient pas comme prévu. Nous avons dépensé beaucoup de temps et d'énergie à vérifier notre code et nos branchements avant de nous rendre compte que le problème provenait de la bibliothèque du programme. D'une version à l'autre, les bibliothèques sont actualisées et dans le cas des AX12 (le modèle de servomoteurs dont nous disposions), elles ne remplissaient plus leur fonction du fait d'une très récente mise à jour de la bibliothèque Mbed, bibliothèque de base pour coder avec une carte Nucleo. Nous nous sommes intéressés au codage d'une bibliothèque mais ce procédé était d'une difficulté trop élevée. Après en avoir discuté avec notre enseignant référent, celui-ci nous a procuré l'un de ses programmes compatibles avec les AX-12 afin que nous puissions avoir accès à sa bibliothèque.

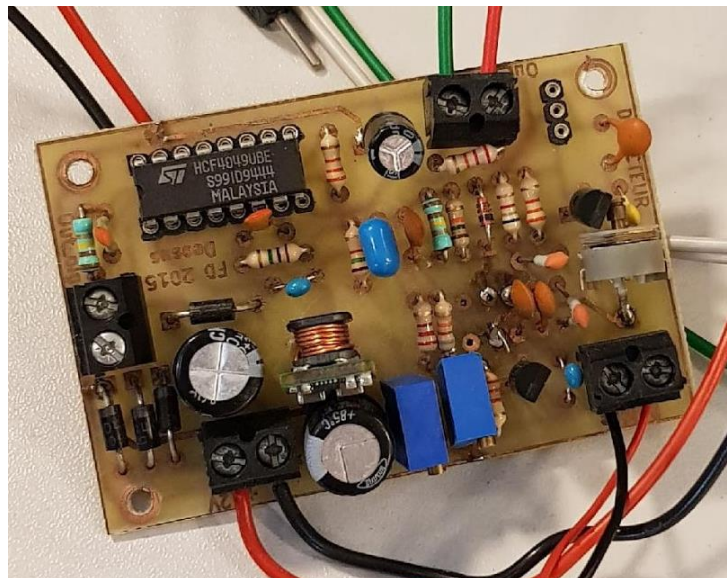
Avec l'aide de M. Delamare, nous avons donc pu faire tourner les AX12. Cependant, après avoir effectué une première rotation, le programme s'arrêtait totalement. De nouvelles recherches nous ont apprises que le problème provenait encore de la bibliothèque. Le programme bloquait en attendant une réponse du servomoteur. Mais configurer la réponse d'un servo est difficile et surtout inutile pour les actions dont nous avons besoin. C'est un élève de 3^{ème} année qui nous fournira finalement une bibliothèque parfaitement fonctionnelle.

Il est maintenant possible de faire une suite d'opérations avec nos servomoteurs, le programme n'ayant plus besoin de la réponse d'un AX-12 pour enchaîner les actions. Après avoir défini les identifiants de chacun des servomoteurs montés en série, il nous est possible de les faire fonctionner simultanément.

Nous avons donc finalement pu achever le codage des servomoteurs. Une simple boucle intégrée dans le premier servo nous permet de faire pivoter le détecteur de métal d'un côté, puis de l'autre. A l'appel de la fonction, appelée « balayage », le détecteur balaye donc une fois, dans un sens, devant le robot pour détecter des mines (ou tout objet métallique

équivalent), puis au second appel, le bras parcourt le chemin en sens inverse. Le 2^{ème} servo est utilisé pour le contrôle du train avant afin de faire tourner robot. Les 3 configurations classiques d'angles de roues (droites, rotation à droite et à gauche) sont codées dans différentes fonctions avec des angles définies pour chacun des mouvements souhaités. Toutes ces fonctions sont appelées lors d'un « switch » (cf annexes). L'utilisateur ne peut faire qu'une opération à la fois, cette dernière se trouve parmi une liste d'actions définies. D'où l'intérêt du switch qui est l'équivalent d'un menu de restaurant.

3.2.4. *Le détecteur de métaux*



Carte du détecteur de métaux

Concernant le détecteur de métaux, nous avons utilisé un détecteur artisanal réalisé par des élèves de l'INSA l'année précédente. Nous avons alors eu besoin de l'aide de notre professeur pour comprendre son fonctionnement car aucune documentation n'était fournie.

La dernière carte électronique à coder sur le robot se trouve être la carte « son ». C'est à dire celle chargée de traiter les relevés de la bobine de détection afin de conclure quant à la présence potentielle d'un objet métallique. Omar et Tanguy ont travaillé sur cette partie avec les indications de notre professeur qui les a mis sur la piste de la comparaison des fréquences renvoyées par la bobine. Charles leur est alors venu en aide et a trouvé la méthode de l'autocorrélation. Cette dernière repose sur la récupération d'un signal témoin, c'est-à-dire dans notre cas celui qui est renvoyé par la carte quand il n'y a pas d'objet métallique à proximité. Il faut alors une technique pour comparer en permanence ce signal à celui fourni défilant à l'instant donné. L'autocorrélation permet que le programme renvoie une valeur importante, un pic, quand les signaux se ressemblent. Il s'agira alors de coder une fonction qui réagit quand il y a moins de corrélations pendant un délai donné. Un pic correspondant à un signal sans objet métallique, une absence prononcée de pics coïncideront de manière certaine avec une perturbation du signal défilant provoquée par la présence d'une mine.

Malheureusement, ce code ne pourra pas être réalisé. En effet, il nous a été impossible de récupérer le signal d'entrée, la communication entre la carte et l'ordinateur ne pouvant être établie avec une qualité satisfaisante. Nous avons néanmoins déjà commencé la réflexion sur ce code et un double problème nous était apparu. Il s'agit de faire comprendre à la carte qu'elle doit en permanence récupérer le signal défilant et récupérer le signal sans objet métallique afin de lancer la fonction de comparaison au signal témoin qui contient la méthode de l'autocorrélation.

4. CONCLUSIONS ET PERSPECTIVES

En fin de compte, nous n'avons pas pu compléter totalement nos objectifs. En effet, à l'issu de ce semestre de travail, notre robot n'est pas capable de s'arrêter automatiquement lorsqu'il détecte un métal. Néanmoins, nous pouvons le contrôler à distance et se fier à notre ouïe lors de la détection de métaux.

Conclusions sur l'apport personnel de cet EC projet :

BENNANI Omar : Ce projet a été pour moi une bonne expérience. Il m'a permis d'apprendre à travailler autour d'un projet pendant une certaine durée, d'apprendre à faire face aux éventuelles difficultés rencontrées, et aussi de découvrir le monde de la robotique. Ayant toujours voulu en faire un, ce projet m'a donné les moyens, que ce soit en termes de connaissances ou matériels, afin de le réaliser.

CLOAREC Jonathan : Ce projet fut pour moi une très bonne manière de découvrir le domaine de la robotique. Travailler sur le modèle CAO et la partie montage mécanique du robot a confirmé mon choix de rejoindre le département mécanique l'année prochaine. J'ai vraiment apprécié travailler sur ce projet. Il m'a encore permis d'apprendre à travailler en groupe sur une longue durée, mais cette fois ci avec plus de personnes.

DUCHEMIN Charles : Voulant travailler plus tard dans la robotique, ce robot démineur fut pour moi une véritable approche très enrichissante. En effet, malgré tous les soucis auquel nous avons été confrontés, ce projet a renforcé mon envie de travailler dans ce domaine qu'est la robotique et les systèmes automatisés. Aussi ayant été confronté au travail en équipe, j'ai pu en apercevoir et les bénéfices, et les limites. Pour finir, ce projet de P6 m'a personnellement apporté des connaissances et aussi une manière de travailler particulière : il n'est pas nécessaire de connaître intégralement un sujet (ou un langage) pour en comprendre ce que nous devons utiliser et comment le faire.

GEORGES Hugo : Je souhaitais découvrir le monde de la robotique, ce projet fût très enrichissant de ce point de vue. Ce travail était pour moi un défi, je suis satisfait du résultat obtenu et du travail accompli sachant que nous étions novices dans ce domaine. Nous avons rencontré de nombreux problèmes sur lesquels il fallait le plus souvent trouver une solution par nous-même en se basant sur une documentation anglaise parfois difficile à trouver. Ce projet m'a apporté des connaissances dans le domaine de la robotique jusqu'alors inconnu mais aussi une meilleure expérience dans le travail de groupe et la recherche d'informations.

CHARLESTON Kariès : La plupart des projets auxquels j'ai eu l'occasion de participer ne comportait qu'une partie expérimentale restreinte et celui-ci m'a permis d'expérimenter le travail de groupe lors de la réalisation d'un système électronique. La répartition des tâches est d'autant plus importante qu'elle doit prendre en considération les pièces dont chacun a besoin pour travailler, alors que cette préoccupation n'est que très peu présente lors d'un travail théorique. La réalisation de ce robot m'a confronté à des problèmes que l'on ne peut prévoir lors de la préparation théorique. Les situations de casses mécaniques, d'absences de pièces ainsi que les incertitudes de placement ou d'usinage ont mises ma créativité à rude épreuve. Je suis relativement satisfait du résultat mais j'espère avoir l'occasion de refaire un projet de ce type afin de me rendre compte à quel point les erreurs commises lors de ce semestre m'ont permis de progresser.

LECLERC Tanguy : Ce projet était ma première expérience dans le domaine de la robotique et m'a permis d'y voir plusieurs aspects. M'étant concentré sur le code, j'ai pu voir les différentes manières de coder et de communiquer avec un robot. Il m'a aussi apporté une nouvelle vision du travail de groupe et de l'organisation afin de gérer de nombreuses difficultés ainsi même que quelques différents au sein du groupe.

Perspectives pour la poursuite de ce projet :

Par la suite, on pourrait penser améliorer le système de détection de notre robot, développer un système de marquage au sol ou de gps et même lui donner la capacité d'être autonome. En effet, l'idéal serait de lui indiquer la zone à balayer puis le laisser faire son travail évitant les zones potentiellement dangereuse tout seul. Enfin, une éventuelle coque (réalisé à partir d'une CAO et d'une impression 3D) afin d'améliorer l'aspect esthétique de notre robot serait un plus.

5. BIBLIOGRAPHIE

<http://www.robot-electronics.co.uk/htm/md49tech.htm> (valide à la date du 10/06/2018).

<https://os.mbed.com/platforms/ST-Nucleo-F401RE/> (valide à la date du 15/06/2018).

<https://os.mbed.com/components/AX-12/> (valide à la date du 15/06/2018).

<http://www.instructables.com/id/Reliable-Frequency-Detection-Using-DSP-Techniques/>
(valide à la date du 15/06/2018).

Crédits d'illustrations :

- Carte Arduino uno R3, pages 9 et 6, www.elektor.fr
- Carte MD49, pages 6 et 10, www.gotronic.fr
- Carte Nucleo, page 9, www.amazon.fr
- Tableau de commandes de la MD49, page 10, www.robotshop.com
- Servo-moteur AX-12 A, page 12, <http://www.eu.diigiit.com>
- Le reste des images ont été prises par les membres du groupe

6. ANNEXES

6.1. Programme principal

```
#include "AX12.h"
#include "mbed.h"

#define CMD                0x00
#define SET_SPEED1        0x31
#define SET_SPEED2        0x32
#define SET_ACCEL         0x33
#define SERIAL_TX         PB_6
#define SERIAL_RX         PB_7
#define Rx_Node           PA_2
#define Tx_Node           PA_3

int requete = 0, n = 10, Longueur = 0; //mode Auto
int Largeur = 0; //mode Auto
float i = 0; //compteur longueur
float j = 0; //compteur largeur
int compteur = 0; //compteur espace
int k=0, m=1; //compteurs balayage

Serial md49(SERIAL_TX, SERIAL_RX, 9600);
AX12 ax_roues(D8, D2, 1); // tx rx, id
AX12 ax_bobine(D8, D2, 2);
Serial Node(Tx_Node, Rx_Node, 115200);

void avancer() //10
{
    md49.putc(CMD);
    md49.putc(SET_ACCEL);
    md49.putc(0x00);
    md49.putc(CMD);
    md49.putc(SET_SPEED1);
    md49.putc(0xE6);
    md49.putc(CMD);
    md49.putc(SET_SPEED2);
    md49.putc(0xE6);
}
void avancerdroite() //11
{
    md49.putc(CMD);
    md49.putc(SET_SPEED1);
    md49.putc(0xFF);
    md49.putc(CMD);
    md49.putc(SET_SPEED2);
    md49.putc(0xE6);
}
void avancergauche() //12
{
    md49.putc(CMD);
    md49.putc(SET_SPEED2);
    md49.putc(0xFF);
}
```

```
md49.putc(CMD);
md49.putc(SET_SPEED1);
md49.putc(0xE6);
}

void reculer() //20
{
md49.putc(CMD);
md49.putc(SET_ACCEL);
md49.putc(0x00);
md49.putc(CMD);
md49.putc(SET_SPEED1);
md49.putc(0x05);
md49.putc(CMD);
md49.putc(SET_SPEED2);
md49.putc(0x05);
}
void reculerdroite() // 21
{
md49.putc(CMD);
md49.putc(SET_ACCEL);
md49.putc(0x00);
md49.putc(CMD);
md49.putc(SET_SPEED1);
md49.putc(0x00);
md49.putc(CMD);
md49.putc(SET_SPEED2);
md49.putc(0x05);
}
void reculergauche() //22
{
md49.putc(CMD);
md49.putc(SET_ACCEL);
md49.putc(0x00);
md49.putc(CMD);
md49.putc(SET_SPEED1);
md49.putc(0x05);
md49.putc(CMD);
md49.putc(SET_SPEED2);
md49.putc(0x00);
}

void arreter() //00
{
md49.putc(CMD);
md49.putc(SET_SPEED1);
md49.putc(0x80);
md49.putc(CMD);
md49.putc(SET_SPEED2);
md49.putc(0x80);
}
void pivotergauche() //32
{
md49.putc(CMD);
md49.putc(SET_SPEED1);
md49.putc(0x00);
md49.putc(CMD);
md49.putc(SET_SPEED2);
md49.putc(0xFF);
}
```

```
void pivoterdroite() //31
{
    md49.putc(CMD);
    md49.putc(SET_SPEED1);
    md49.putc(0xFF);
    md49.putc(CMD);
    md49.putc(SET_SPEED2);
    md49.putc(0x00);
}

void RoueTouteDroite()
{
    //mettre les roues droites comme base de départ pour chaque protocole
    ax_roues.SetGoal(150);
}

void RoueVersDroite()
{
    ax_roues.SetGoal(170);
}

void RoueVersGauche()
{
    ax_roues.SetGoal(130);
}

void Balayage()
{
    while ((k>=0)&&(k<=30)) {
        ax_bobine.SetGoal(k*10);
        k=k+m;
        wait(0.6);
    }
    wait(0.2);
    m=-m;
    if(k==31)
        k=30;
    else if (k==-1)
        k=0;
    //Faire tourner la bobine en fct de n
    //Detection() à faire tous les 10°
    //changer le signe de n à la fin
    //pour changer le sens de rotation de la bobine
}

void Detection()
{
    //la detection de la bobine
}

int main (){

    while(1){ //boucle infinie
        requete = Node.getc();//appelle d'un caractère envoyer par la carte wifi
        requete += Node.getc();
        switch (requete)//si requête vaut :
        {
            case 10://10 alors tu execute ça:
                TournerBobine();
                avancer();
                TournerBobine();
        }
    }
}
```

```
        break;
    case 11:
        TournerBobine();
        avancerdroite();
        TournerBobine();
        //tourner les roues avt vers la droite
        break;
    case 12:
        TournerBobine();
        avancergauche();
        TournerBobine();
        //tourner les roues avt vers la gauche
        break;

    case 20:
        reculer();
        TournerBobine();
        break;
    case 21:
        reculerdroite();
        TournerBobine();
        //tourner les roues avt vers la droite
        break;
    case 22:
        reculergauche();
        TournerBobine();
        //tourner les roues avt vers la gauche
        break;

    case 00:
        arreter();
        TournerBobine();
        break;
    case 31:
        TournerBobine();
        pivoterdroite();
        TournerBobine();
        //tourner les roues avt vers la droite
        break;
    case 32:
        TournerBobine();
        pivotergauche();
        TournerBobine();
        //tourner les roues avt vers la gauche
        break;
    case 42://mode automatique
        Longueur = Node.getc();
        Largeur = Node.getc();
        for (i=4*0,44; i<=Longueur; i = i+ 0,44){

        }
        else {
            for (i = 0; i <= Longueur; i = i + 0,4){
                TournerBobine(n);
                if (Largeur > 0){
                    pivoterdroite();
                    wait_ms(100);
                }
                else if{Largeur < 0){
                    pivotergauche();
                    wait_ms(100);
                }
            }
        }
    }
}
```

```
        }  
        else{  
            stop();  
        }  
        TournerBobine(n);  
    }  
    }  
    i = 0;  
}  
  
    for (){  
    }  
}  
  
    //remettre les roues avant droite pour ne pas gêner les codes suivants  
}  
}
```

6.2. Programme wifi

```
#include <ESP8266WiFi.h>
```

```
const char* ssid = "GetOut!";
```

```
const char* mdp = "1234azerty";
```

```
WiFiServer serveur (80);
```

```
#define RX D7 // celui de la nodemcu v3 à brancher sur le rx de la nucleo:PA_2
```

```
#define TX D4 // de même ! PA_3: sortant droite tout en bas gauche
```

```
String Action = "";
```

```
int Longueur = 1;
```

```
int Largeur = 1;
```

```
void setup () {
```

```
    Serial.begin(115200);
```

```
    delay(10);
```

```
    WiFi.softAP(ssid,mdp);
```

```
    analogWriteFreq(20000);
```

```
    analogWriteRange(255);
```

```
    serveur.begin();
```

```
    Serial.println("Serveur en marche");
```

```
    Serial.println(WiFi.softAPIP());
```

```
}

void loop() {
  WiFiClient client = serveur.available();
  if (!client){
    return;
  }

  Serial.println("Un appareil s'est s'est connecté?");

  while (!client.available()){
    delay(1);
  }

  Serial.println("L'appareil a envoyé des données?");

  //Lire la requête:
  String requete = client.readStringUntil('\r');
  Serial.println(requete);
  client.flush();

  //identification de la requête:
  /*if (requete.indexOf("/A+") > 0){
    //if (Vitesse_D == 0) Vitesse_D= 75; utile ssi on veux controller la vitesse du robot
    //Vitesse_D += 25;
    //if Vitesse_D >= 250) Vitesse_D = 250;
    //Vitesse_G = Vitesse_D;
    analogWrite(TX,10);//la requête
    //analogWrite(TX,); //la valeur si besoin est!

  }*/
  if (requete.indexOf("/A") > 0){
    analogWrite(TX,10);
    Action = "Mise en marche avant";
  }

  if (requete.indexOf("/AD") > 0){
    analogWrite(TX,11);
    Action = "Mise en marche avant droite";
  }
}
```



```
}
```

```
if (requete.indexOf("/AG") > 0){  
    analogWrite(TX,12);  
    Action = "Mise en marche avant gauche";  
}
```

```
if (requete.indexOf("/R") > 0){  
    analogWrite(TX,20);  
    Action = "Mise en marche arriere";  
}
```

```
if (requete.indexOf("/RD") > 0){  
    analogWrite(TX,21);  
    Action = "Mise en marche arriere droite";  
}
```

```
if (requete.indexOf("/RG") > 0){  
    analogWrite(TX,22);  
    Action = "Mise en marche arriere gauche";  
}
```

```
if (requete.indexOf("/S") > 0){  
    analogWrite(TX,00);  
    Action = "Arret Immediat";  
}
```

```
if (requete.indexOf("/P_D") > 0){  
    analogWrite(TX,31);  
    Action = "Mise en pivot vers la droite";  
}
```

```
if (requete.indexOf("/P_G") > 0){  
    analogWrite(TX,32);  
    Action = "Mise en pivot vers la gauche";  
}
```

```
if (requete.indexOf("/Auto") > 0){
    analogWrite(TX,42);
    delay(100);
    analogWrite(TX,Longueur);
    delay(100);
    analogWrite(TX,Largeur);
    Action = "Yolo laisse faire le Robot";
}
if (requete.indexOf("/Lo+") > 0){
    Action = "Plus de Lo";
    Longueur = Longueur + 1;
}
if (requete.indexOf("/Lo-") > 0){
    Action = "Moins de Lo";
    Longueur = Longueur - 1;
}
if (requete.indexOf("/La+") > 0){
    Action = "Plus de La";
    Largeur = Largeur + 1;
}
if (requete.indexOf("/La-") > 0){
    Action = "Moins de La";
    Largeur = Largeur - 1;
}

// Code de la page HTML:
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("");
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head>");
client.println("<meta name='apple-mobile-web-app-capable' content='yes'/>");
client.println("<meta name='apple-mobile-web-app-status-bar-style' content='black-translucent'/>");
client.println("</head>");
client.println("<body bgcolor = \"#e0e6ec\">");
client.println("<hr/><hr>");
```

```
client.println("<h2><i><center> Controle du supposé Robot Démineur via WiFi</i></center></h2>");
client.println("<hr/><hr>");
client.println("<h2><i><center> Duchemin Charles 06/2018 </i></center></h2>");
client.println("<hr/><hr>");

client.println("<br><br>");

client.println("<center>");
client.println("Avancer:");
client.println("<a href=\\\"/A\\\"><button>Avancer</button></a>");
//client.println("<a href=\\\"/A-\\\"><button>Avancer moins vite</button></a>");
client.println("</center>");
client.println("<br><br>");
client.println("<center>");
client.println("Avancer en Tournant vers:");
client.println("<br><br>");
client.println("<a href=\\\"/AG\\\"><button>la Gauche</button></a>");
client.println("<a href=\\\"/AD\\\"><button>la Droite</button></a>");
client.println("</center>");

client.println("<br><br>");

client.println("<center>");
client.println("Reculer tout droit");
client.println("<a href=\\\"/R\\\"><button>Reculer</button></a><br />");
client.println("</center>");
client.println("<br><br>");
client.println("<center>");
client.println("Reculer en Tournant vers:");
client.println("<br><br>");
client.println("<center>");
client.println("<a href=\\\"/RG\\\"><button>la Gauche</button></a>");
client.println("<a href=\\\"/RD\\\"><button>la Droite</button></a>");
client.println("</center>");

client.println("<br><br>");
client.println("<br><br>");

client.println("<center>");
```

```
client.println("Pivoter vers:");
client.println("<br><br>");
client.println("<a href='\"/P_G\"'><button>la Gauche</button></a>");
client.println("<a href='\"/P_D\"'><button>la Droite</button></a>");
client.println("</center>");

client.println("<br><br>");
client.println("<br><br>");

client.println("<center>");
client.println("Arret Immediat");
client.println("<br><br>");
client.println("<a href='\"/S\"'><button>STOP</button></a>");
client.println("</center>");

client.println("<br><br>");
client.println("<br><br>");

client.println("<center>");
client.print("Action en cours: ");
client.println(Action);
client.println("</center>");

client.println("<br><br>");

client.println("<center>");
client.println("Mode Automatique:");
client.println("<br><br>");
client.println("<center>");
client.print("Longueur =");
client.print(Longueur);
client.print("metres");
client.print("          Largeur =");
client.print(Largeur);
client.println("metres");
client.println("<br><br>");
client.println("<center>");
client.println("<a href='\"/Lo+\"'><button>Plus de Longueur </button></a>");
client.println("<a href='\"/Lo-\"'><button>Moins de Longueur </button></a>"); //<br />
```

```
client.println(" ");
client.println("<a href=\\\"/La+\\\"><button>Plus de Largeur </button></a>");
client.println("<a href=\\\"/La-\\\"><button>Moins de Largeur </button></a><br />");
client.println("<br><br>");
client.println("<center>");
client.println("<a href=\\\"/Auto\\\"><button>Activation</button></a>");
client.println("</center>");
client.println("<br><br>");
client.println("center");
client.println("<a href=\\\"/Auto\\\"><button>Oui!</button></a>");
client.println("</center>");

client.println("</html>");
delay(1);
Serial.println("Deconexion");
Serial.println();
}
```

6.3. Propositions de sujets de projets (en lien ou pas avec le projet réalisé)

Pour les sujets de l'année prochaine, une automatisation de notre robot pourrait être un sujet intéressant. Sinon, créer d'autres types de robots peut être très intéressant, comme par exemple en robot éviteur d'obstacle.