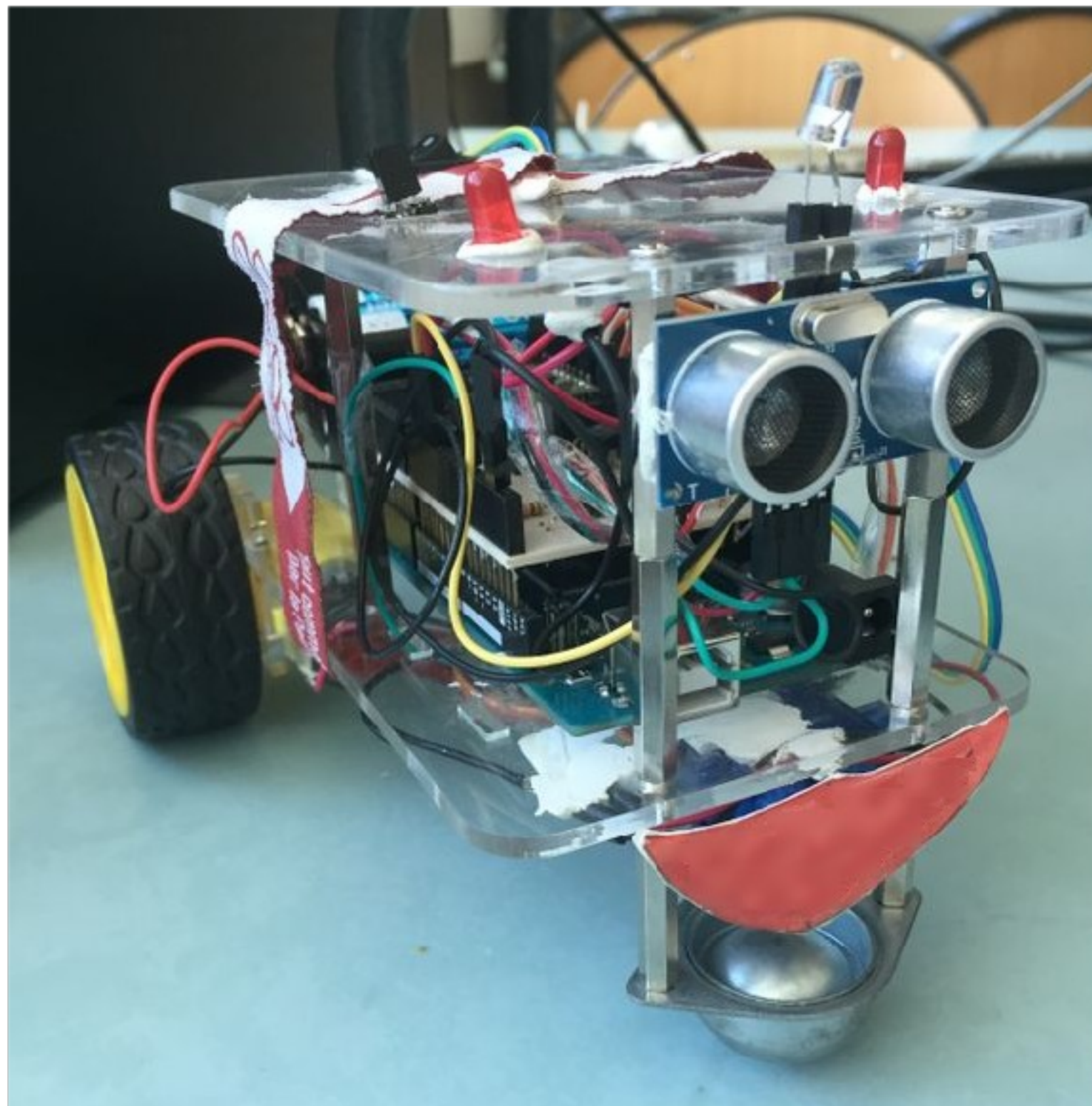


École Polytechnique Universitaire de Lille, classe
préparatoire aux écoles d'ingénieurs Polytech

Robot autonome communiquant



Encadrant : M. Redon

Cahier des charges

- Robot de dimensions inférieures à un cube de 10cm.
- Éviter des obstacles durs
- Communiquer avec les autres robots (émission d'un message IR)
- Reconnaître les signaux de ses amies proies
- Repérer ses prédateurs en détectant leurs signaux IR et tenter de leur échapper
- S'arrêter lorsqu'il est capturé par un robot proie (via un capteur à effet hall)

Matériel utilisé

- Arduino MEGA
- 2 Moteur CC
- 2 LEDs IR
- 1 Récepteur IR
- 1 "Shield" adapté à l'Arduino
- Sonar HCSR04
- Contrôleur de moteur TB6612FNG
- 6x Piles 1.5V
- 1 Capteur à effet hall
- 1 Servomoteur
- 1 Interrupteur

Sommaire

Introduction	1
Journal de Bord	2
Séance 1-4 : Planification	2
Séance 5-10 : Mise en place du Shield	2
Séance 6-7 : Châssis 3D	3
Séance 8-9 : Code – Première partie	4
Séance 10-11 : Conception du châssis 2D	4
Séance 12-13 : Code - Seconde partie	4
Séance 14 : Montage	5
Séance 15-16 : Châssis 2D - Rectification	5
Séance 17 : Soudure	6
Séance 18 : Assemblage	6
Séance 19 : Test	6
Code	6
Fonctionnalités du robot final	8
Conclusion	8

Introduction

Dans le cadre de ce bureau d'études, notre but est de réaliser un robot proie, capable de communiquer avec les autres robots. Le robot est censé évoluer de façon autonome tout en indiquant sa présence via émission IR. Au cours de ses déplacements, il doit savoir détecter les signaux des prédateurs afin de leur échapper. En plus de cela, il doit être capable d'éviter les obstacles durs (murs). Ce wiki relatara les différentes parties de sa réalisation, le raisonnement suivi pour la conception des éléments du robot ainsi que les difficultés que nous avons rencontrées.

Journal de Bord

Séance 1-4 : Planification

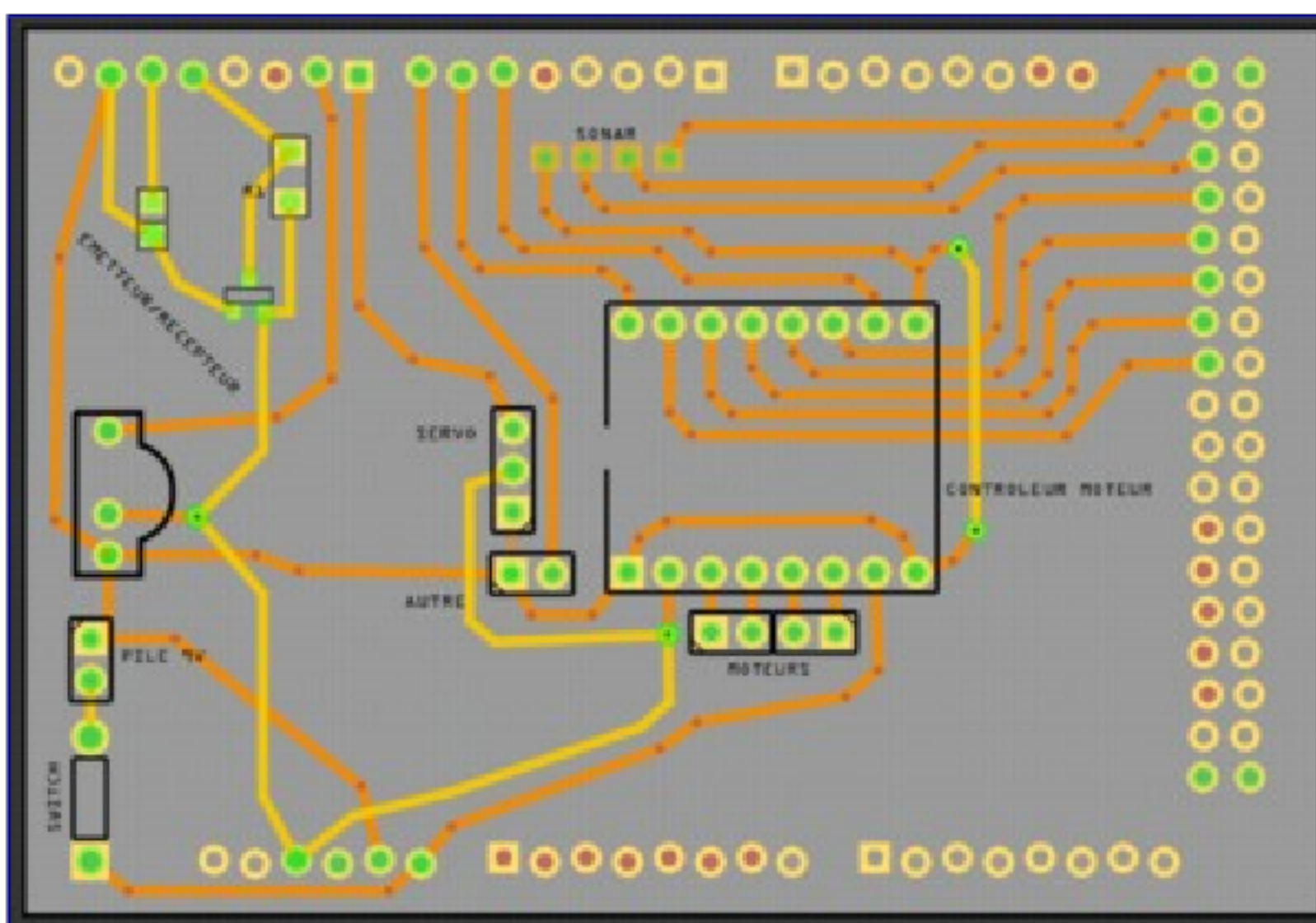
Après avoir décidé de partir sur un robot proie, nous avons dû décider de la forme que prendra celui-ci. Nous nous sommes dit qu'il valait mieux faire un robot de petite taille, agile et rapide. Il nous faudra donc modéliser le châssis. Nous partirions sur un Arduino, que nous avons déjà eu le plaisir de découvrir en cours de spécialité IMA au S3. Afin de rendre notre robot original, nous avons décidé de partir sur un châssis modélisé en 3D, que nous imprimerons au Fabricarium.

Séance 5-10 : Mise en place du Shield

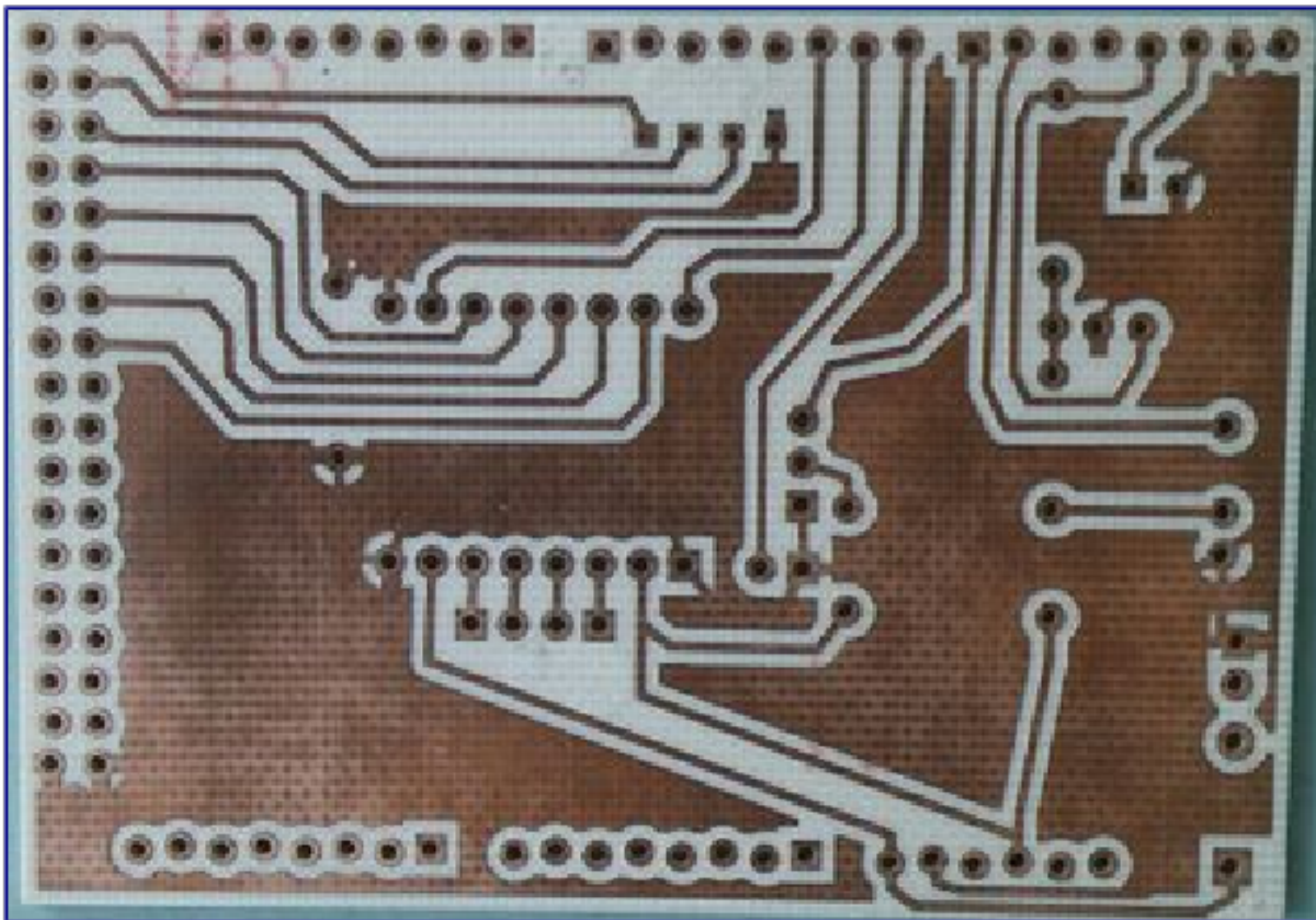
Le robot fonctionnera à l'aide d'une carte Arduino sur laquelle est connectée une carte imprimée que nous allons réaliser totalement adaptée à nos besoins. Pour cela, nous avons utilisé le logiciel Fritzing sur lequel il est très facile de réaliser son circuit et observer la présence, ou non, d'erreurs.

Le fonctionnement du circuit graphique et le lien avec le PCB ont dû être compris et cela est passé par de nombreuses phases de tests. Au final, après avoir déterminé les éléments nécessaires pour notre robot, le circuit a pris forme pour au final être imprimé après validation.

Voici le schéma, autrement appelé PCB, avant les modifications et adaptations pour l'imprimante (ce n'est pas le schéma final) :



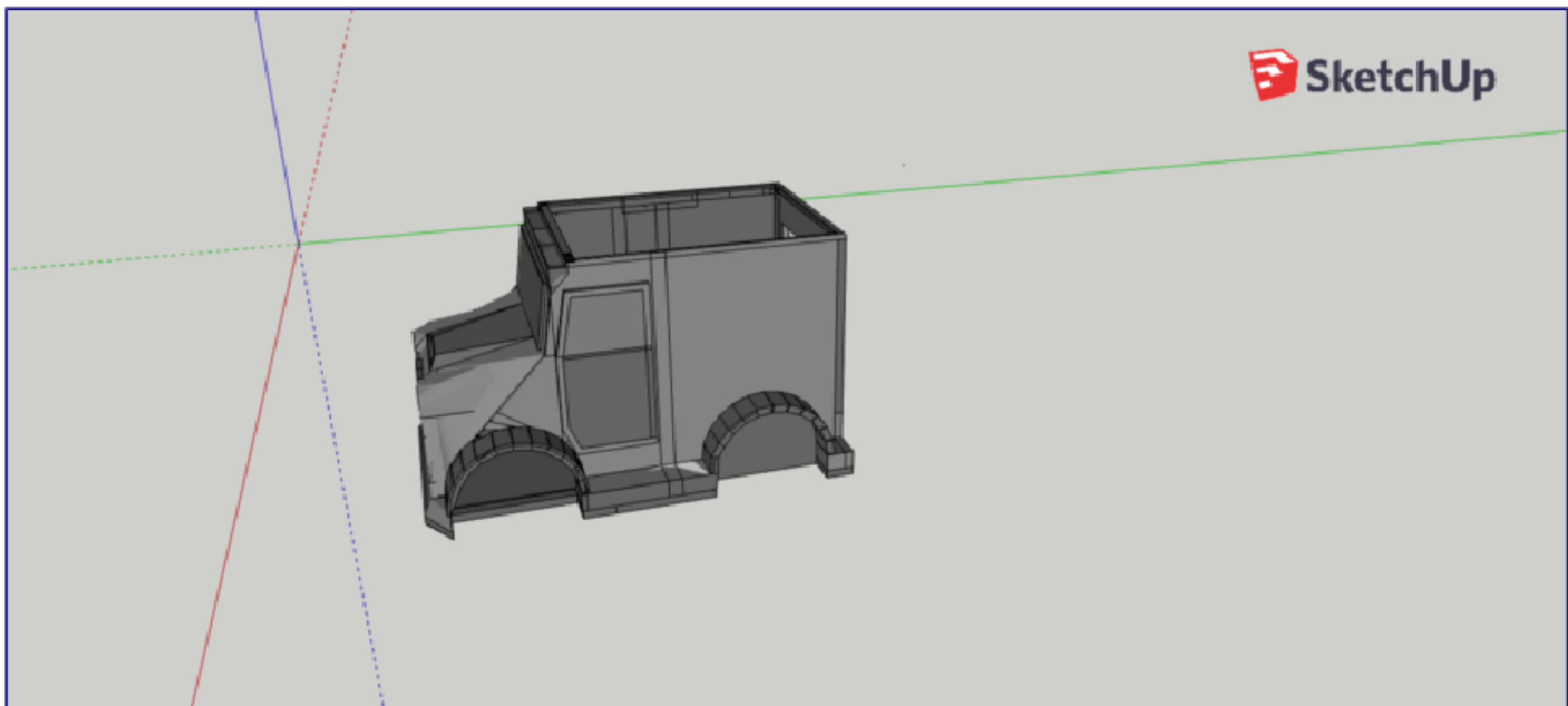
Après impression, le résultat final :



Shield final conçu via Fritzing

Séance 6-7 : Châssis 3D

Partant de notre première planification nous avons modélisé un camion 3D sur Sketchup adapté aux dimensions de notre robot. La prise en main s'est avérée laborieuse mais instructive. Cependant, nous du faire face à de nombreux problèmes. Tout d'abord le modèle était un peu trop complexe pour être imprimable en 3D, il aurait fallu de nombreuses cales et réimpression afin d'avoir un châssis de robot acceptable. Le 2ème problème était dû au fait que la modélisation du châssis avait été conçue en une seule pièce. Ainsi le modèle posait problème avec le logiciel 3D car il considérait que certain solide était positionné au-dessus de certains autres solides creux. De plus, le logiciel, Cura, prévoyait plus de 10h d'impression pour un robot de 20x12x10cm. Ce qui est presque impossible à réaliser avec le matériel du Fabricarium, à moins de lancer l'impression la nuit, auquel cas nous n'aurions pas pu intervenir sur l'imprimante en cas de bug. C'est alors que nous nous sommes tournés vers nos professeurs afin de leur demander conseil. Ils nous ont alors précisé que les impressions 3Ds n'étaient pas très robustes, et qu'il valait mieux partir sur un châssis en plexiglas ou en bois. Après plus d'une dizaine d'heures de travaux personnels pour la modélisation de notre robot, nous avons pris la décision de repartir de zéros et de partir sur un châssis en plexiglas. Le choix a été difficile à prendre car plusieurs fonctionnalités « fun » de notre robot reposaient sur ces modèles 3D.



Modélisation 3D du châssis sur Sketchup

Séance 8-9 : Code – Première partie

Dans une première partie, nous avons déclaré nos ports en tant que constantes, afin de nous faciliter la tâche lors de la programmation de l'Arduino. Ensuite nous nous sommes occupés des fonctions de déplacement du robot : avancer, reculer, tourner à gauche ou à droite. Une fois ces fonctions codées, nous avons relié le sonar à l'Arduino, puis nous avons déterminé une distance minimale raisonnable à partir de laquelle le robot tourne à gauche pour éviter l'obstacle.

Séance 10-11 : Conception du châssis 2D

Après notre échec en modélisation 3D, nous sommes repartis pour la modélisation mais cette fois-ci en 2D. La prise en main du logiciel s'est avérée plus facile que celle de sketchup, et nos bonnes pratiques de mesures et de précision, développées lors de la modélisation 3D, nous ont permis de concevoir un châssis fiable et précis. Nous sommes partis sur une conception de 2 étages superposables, munis de 2 encoches pour les roues. Les piles et les moteurs CC seront fixés en dessous du 1er étage, l'Arduino et le shield au-dessus de celui-ci, et enfin les LEDs et les différents capteurs au-dessus du 2ème étage dans un souci d'esthétique.

Séance 12-13 : Code - Seconde partie

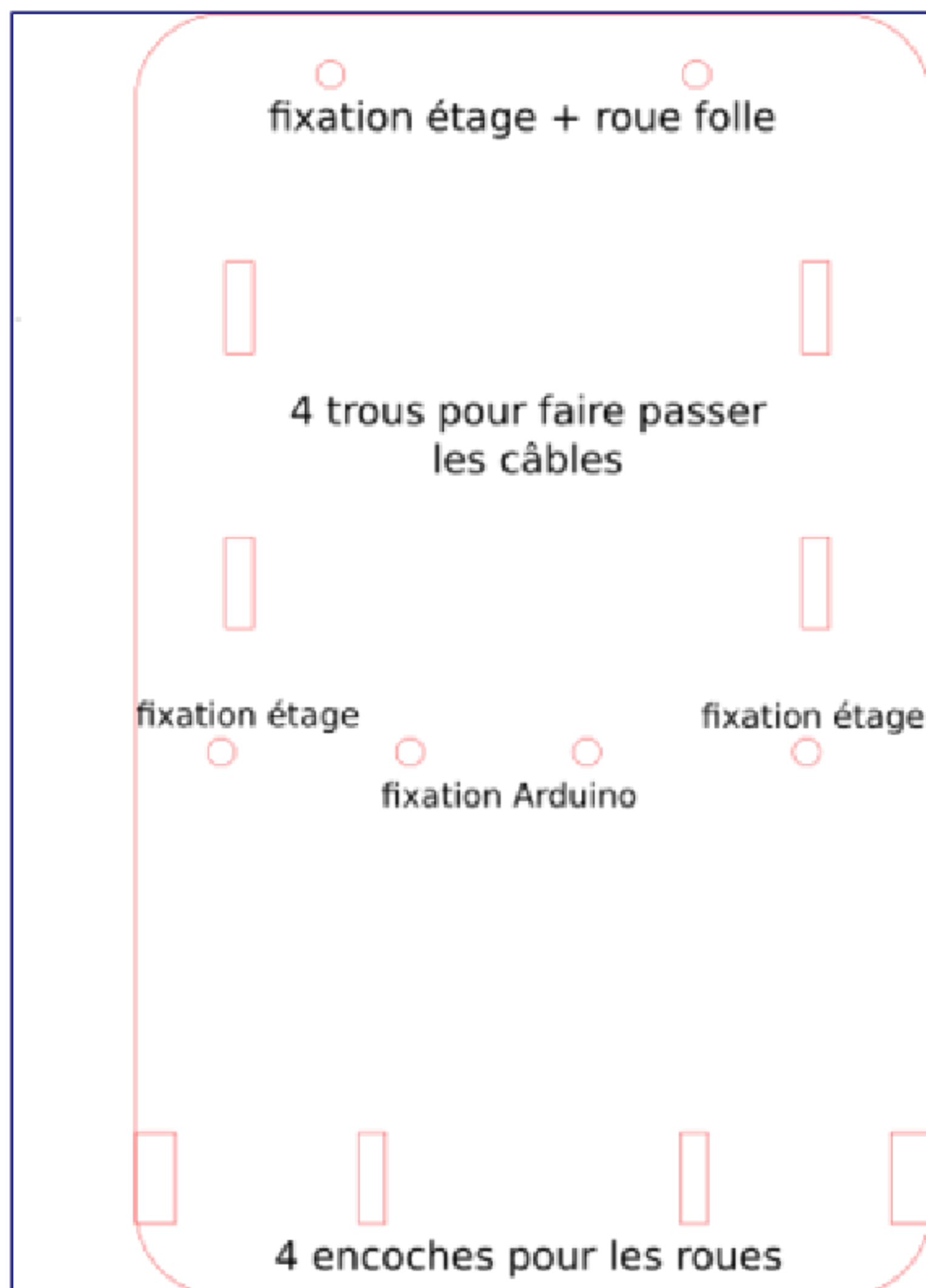
Dans notre deuxième phase de programmation, nous nous sommes occupés du TSOP. Tout d'abord, nous avons codé la fonction de détection du signal, que nous convertissons en hexadécimal dans un second temps. Pour tester notre TSOP, nous avons utilisé une télécommande mise à notre disposition. C'est alors que nous avons eu l'idée de récupérer le code hexadécimal du bouton on/off de la télécommande. Nous avons ainsi programmé notre Arduino pour qu'il s'arrête dès qu'il détecte ce code hexadécimal. Il est aussi capable de reprendre son fonctionnement lorsqu'il le redétecte.

Séance 14 : Montage

Après avoir dessiné nos 2 étages, nous nous sommes rendus au Fabricarium afin de réaliser la découpe. Nous avons appris à régler la machine et à lancer la découpe de A à Z. Cela a été très enrichissant pour nous, et nous a permis de nous rendre compte de la simplicité de réalisation d'un châssis 2D comparé à un modèle 3D. Cependant nous nous sommes aperçus que les moteurs ne tenaient pas très bien sur notre 1er étage.

Séance 15-16 : Châssis 2D - Rectification

Dans un premier temps nous nous sommes dit qu'il valait mieux coller les moteurs sur la plaque de plexiglas, afin d'éviter qu'ils bougent, mais après avoir demandé conseil aux professeurs, nous avons décidé de nous inspirer des robots des sessions précédentes, et de partir sur un système 4 encodons (contre 2 actuellement), tout en réduisant légèrement la taille des pièces de fixation du robot.



Modèle final du châssis en plexiglas

Séance 17 : Soudure

Imprimer une carte ne suffit pas, il faut également s'assurer que tous les éléments que l'on souhaite sont bien connectés. Grâce au schéma initial, on a pu corriger ce qui n'était pas possible à l'impression (notamment croiser deux routes, le double couche n'étant pas possible sur cette imprimante). C'est donc tout naturellement que nous avons été amenés à utiliser le fer à souder. On a également ajouté les connecteurs femelles, permettant de connecter les fils connecteurs (mâle /mâle - mâle/femelle).

Séance 18 : Assemblage

Fort de notre expérience en découpe laser, c'est avec facilité que nous avons découpé le 2ème étage de notre robot. Une fois les 2 étages découpés et le shield soudé, il ne nous reste plus qu'une étape dans la conception mécanique de notre robot : l'assemblage. En une petite demi-heure, nous avons collé, visé... et notre robot pris forme ! Nous avons décidé d'acheter une petite batterie 9V plutôt que de prendre les 6 piles fournies par l'école dans un souci de place et de poids.

Séance 19 : Test

Une fois le robot assemblé nous avons procédé à quelques tests afin de nous assurer que le montage avait bien été fait et que le shield avait bien été soudé. C'est avec joie que nous avons pu voir notre robot rouler de façon autonome dans la classe tout en évitant les murs et les armoires ! Il ne nous restait plus qu'à coder les LEDs et le capteur à effet hall. Cependant nous nous sommes vite aperçus que la batterie 9V se déchargeait beaucoup trop vite, nous avons donc dû repartir sur 6 piles de 1,5 V.

Code – Dernière partie 20 - et plus

Parlons à présent du code, qui a occupé 60% du projet, les corrections étant très régulières.

Comment avons-nous procédé ?

La première chose, primordial, était de définir les PINs sur lesquelles seront connectés tous les éléments.

La première fonctionnalité a été la mise en route des moteurs, et le contrôle de la vitesse et de leur fonction de déplacement.

Pour cela, nous avons codé des fonctions périphériques afin de le rendre plus lisible.

```
void Avant() {  
    digitalWrite (AIN1, HIGH);  
    digitalWrite (AIN2, LOW);  
    digitalWrite (BIN1, HIGH);  
    digitalWrite (BIN2, LOW);  
}
```

```
void Arriere() {  
    digitalWrite (AIN1, LOW);
```

```

    digitalWrite (AIN2, HIGH);
    digitalWrite (BIN1, LOW);
    digitalWrite (BIN2, HIGH);
}

void goForward (int pwm)
{
    Avant();
    analogWrite(PWMA, pwm);
    analogWrite(PWMB, pwm);
}

void goBackward (int pwm)
{
    Arriere();
    analogWrite(PWMA, pwm);
    analogWrite(PWMB, pwm);
}

void rotateRight (int pwm)
{
    analogWrite(PWMA, 2 * pwm);
    analogWrite(PWMB, pwm);
}

void rotateLeft (int pwm)
{
    analogWrite(PWMA, pwm);
    analogWrite(PWMB, 2 * pwm);
}

```

Le robot étant capable de rouler, nous avons ajouté la détection d'obstacle afin qu'il évite de s'abîmer.

```

int Distance1 = sonar.ping_cm();
delay(100);
int Distance2 = sonar.ping_cm();
int Distance = max(Distance1, Distance2);

```

Voici le scénario type que va effectuer le robot :

- Présence d'un champ magnétique ? (effet hall)
- Présence d'une lumière IR ? (TSOP)
- Si aucun des deux : on détecte la distance des obstacles et on avance si elle est supérieure à 20cm.
- Si on détecte un signal IR, alors le robot se place en mode FUIITE : la vitesse des moteurs augmentent, les détections deviennent plus rapides
- Si on détecte un champ magnétique, le robot est considéré comme mort et s'arrête, définitivement.

Pour l'émission du signal IR, nous avons opté pour un code très simple :

```

void Emettre(){
    IRsend.sendNEC(2, 32);
    IRrecv.enableIRIn();}

```


La réception d'un signal IR est également très simple :

```
int IR_Mesure()
{
    irrecv.enableIRIn();
    if (irrecv.decode(&results))
    {
        irrecv.resume();
        return 1;
    }
    else
    {
        irrecv.resume();
        return 0;
    }
}
```

Fonctionnalités du robot final

Après de longues séances de travail, nous sommes arrivé à obtenir un robot fonctionnel, disposant de nombreuses fonctions. Certaines étaient imposées par notre bureau d'étude, mais nous en avons rajouté d'autres, afin d'avoir un robot unique et personnalisé. Désormais nous sommes fiers de vous présenter notre robot qui est capable de :

- Évoluer de façon autonome tout en évitant les obstacles (en tournant sur sa gauche)
- Arrêter et recommencer sa course grâce au bouton d'une télécommande (en détectant le code hexadécimal envoyé par le bouton on/off de la télécommande)
- Commencer/arrêter sa course grâce à un interrupteur mécanique
- S'arrêter lorsqu'il est capturé par un robot proie (via un capteur à effet hall)
- Émettre des signaux infrarouges
- Récupérer des signaux infrarouges grâce à un TSOP
- Se mettre en mode fuite : le robot enclenche plusieurs virages en allant beaucoup plus vite

Conclusion

En conclusion, ce bureau d'études a été un excellent moyen de découvrir ce que le département IMA avait à nous proposer. Nous avons pu découvrir des notions d'électronique, d'informatique, ou encore de modélisation (2D/3 D). Lors de ce BE, nous nous sommes rendu compte que la simple notion d'électronique ou d'informatique ne suffisait pas à la réalisation d'un robot. Nous avons dû lier la pratique à la théorie, l'électronique à la mécanique tout en restant pragmatique. L'esprit critique était aussi de la partie, nous avons dû certes prendre des décisions difficiles, quitte à revoir notre feuille de route, mais nous sommes aujourd'hui fiers du travail que nous avons accompli. C'est donc sur un point positif que nous achevons notre projet qui nous a permis de nous faire une bonne idée du département IMA.