

Rapport de projet: ARDUIBOT



31 mai 2018

Table des matières:

1	Pourquoi ce choix ?	3
2	Définition des objectifs	4
3	Matériel utilisé 1) Impression 3d 2) Composants	5-6
4	Réalisation 1) Assemblage 2) Code	7-9
5	Problèmes rencontrés au cours du projet	10-11
6	Améliorations futures	12
	Conclusion	13

1 Pourquoi ce choix ?

Lorsque nous regardions lors de la première séance les différents projets déjà réalisés sur internet, nous voulions tous les deux réaliser un robot. Nous trouvions que c'était ce qui était le plus intéressant et amusant à réaliser pour un projet comme celui-ci. Nous avons donc visionné de nombreuses vidéos sur les robots réalisés grâce à Arduino et nous sommes tombés d'accord pour réaliser celui-ci. Ce qui nous a tout de suite attiré était son design minimaliste et très jovial, ainsi que ses caractéristiques faciles à comprendre et à utiliser. Nous trouvions que c'était un jouet parfaitement adapté à des enfants, ce qui nous a tout de suite motivé car ce genre d'objet est donc très populaire chez les personnes extérieures.



LittleBot 3D Printed Arduino Robot Introduction

LittleArm Robot • 14 k vues • il y a 1 an

For more information visit <http://www.littlearmrobot.com/> The LittleBot is an Arduino-based 3D printed robot for beginners. He can ...



Assembly and Use of the LittleBot Arduino Robotics Kit

LittleArm Robot • 2,9 k vues • il y a 9 mois

This video details how to assemble and use the android app for the LittleBot arduino robotics kit. This 3D printed robot kit can be ...



LittleBot Walkthrough

LittleArm Robot • 771 vues • il y a 1 an

This is an informal walkthrough of the 3 LittleBot robots that are available on our Kickstarter For more information visit ...



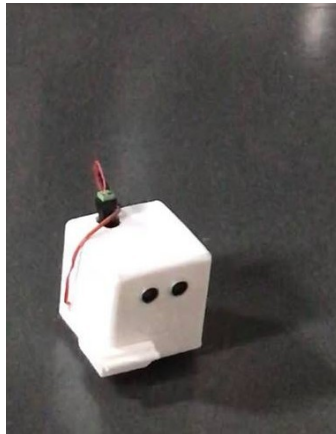
LittleBot: The Best Arduino Robot for Beginners

LittleArm Robot • 8,5 k vues • il y a 1 an

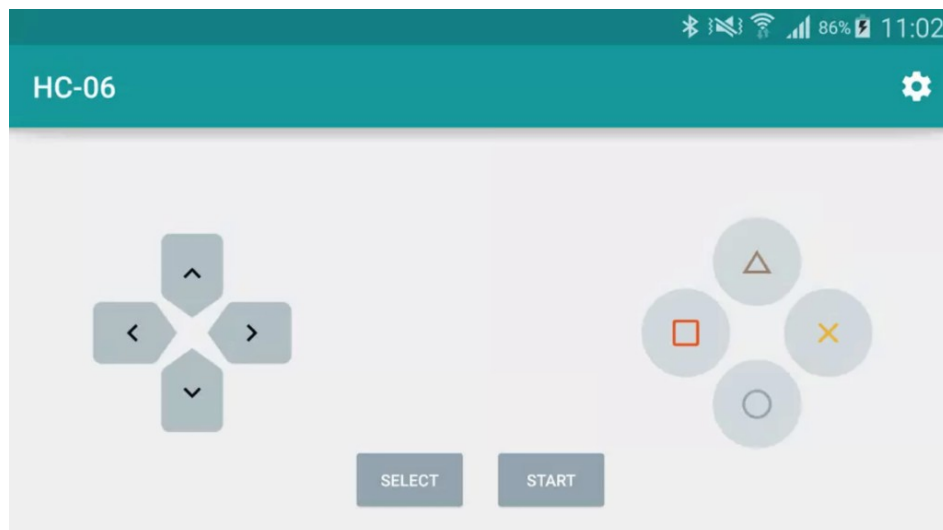
The Littlebot s available on Kickstarter or Indiegogo. He is a fully 3D printed arduino robot that can be controlled with a bluetooth ...

2- Définition des objectifs

Nous avons ensuite dressé le cahier des charges de notre robot. Il y avait plusieurs caractéristiques sur lesquelles nous étions certains, et d'autres sur lesquelles nous hésitions. Par exemple, nous savions dès le début que notre robot posséderait un moyen de repérer les distances grâce à des capteurs qui feraient office d'yeux.



Nous voulions aussi pouvoir le diriger manuellement mais nous hésitions encore à l'époque sur un contrôle par application android ou par joystick (l'application android serait facile à télécharger mais un joystick serait plus instinctif à prendre en main pour un enfant). Pour débiter nous avons donc utilisé des applications déjà faites, puis nous avons trouvé une application vierge que nous avons personnalisé. Cette application s'appelle arduino bluetooth controller.



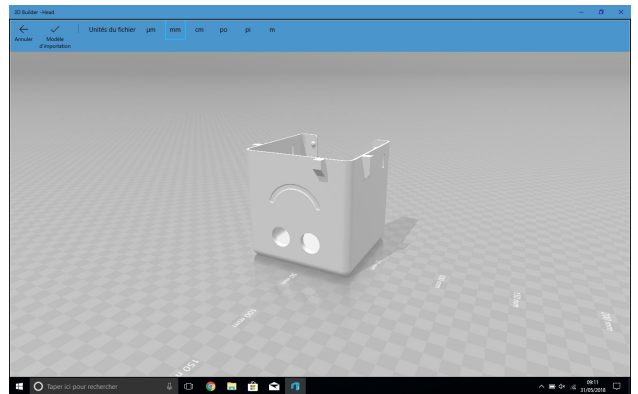
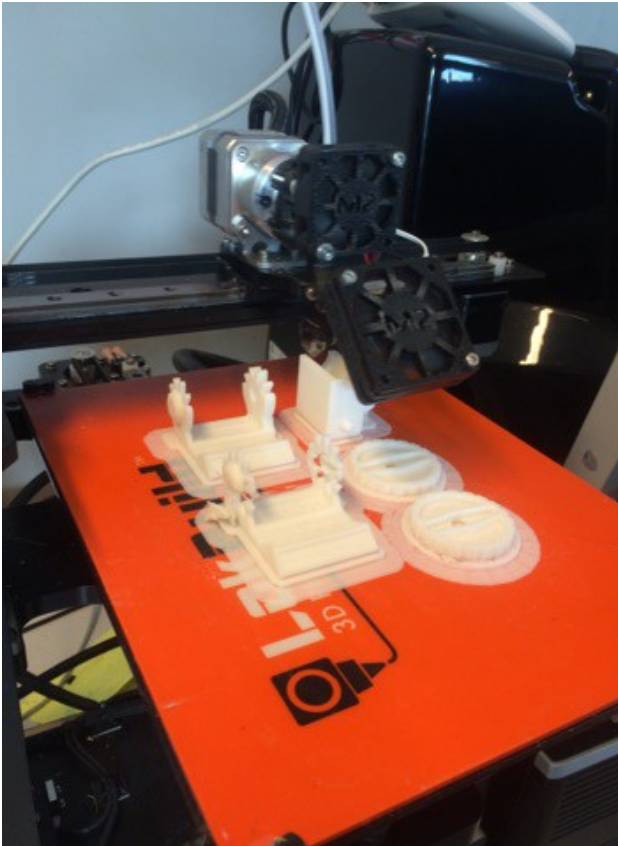
Nous voulions également que notre robot émette des bruits grâce à un buzzer afin de ressembler à un véritable petit Android (notre inspiration était les bruits de r2d2 et de bb8).

Nous voulions enfin si possible créer plusieurs modèles différents afin de les faire interagir entre eux lors de combats de sumo ou de courses de vitesse par exemple.

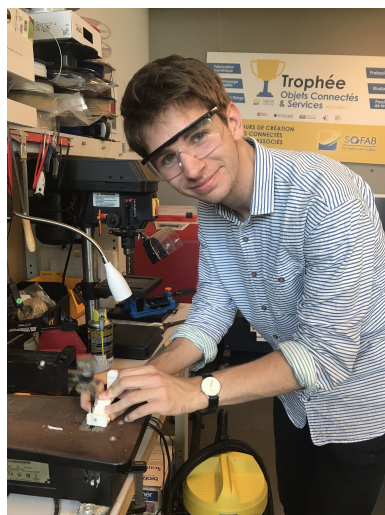
3 Matériel utilisé

1 Impression 3d

La première étape a été d'imprimer le corps de notre robot, ainsi que ses différentes parties. Nous avons pour cela fait appel au Fablab du site des Templiers et de leur imprimante 3d.



Le design du robot et ses courbes simples ont rendu la réalisation et la modification de certaines pièces plutôt faciles. Nous avons eu quelques problèmes sur certaines pièces qui étaient trop fines à certains endroits pour la précision de l'imprimante 3d, nous avons donc du modifier de nombreuses pièces que nous avons par la suite modifiées à la main (ponçage, perçage, etc).



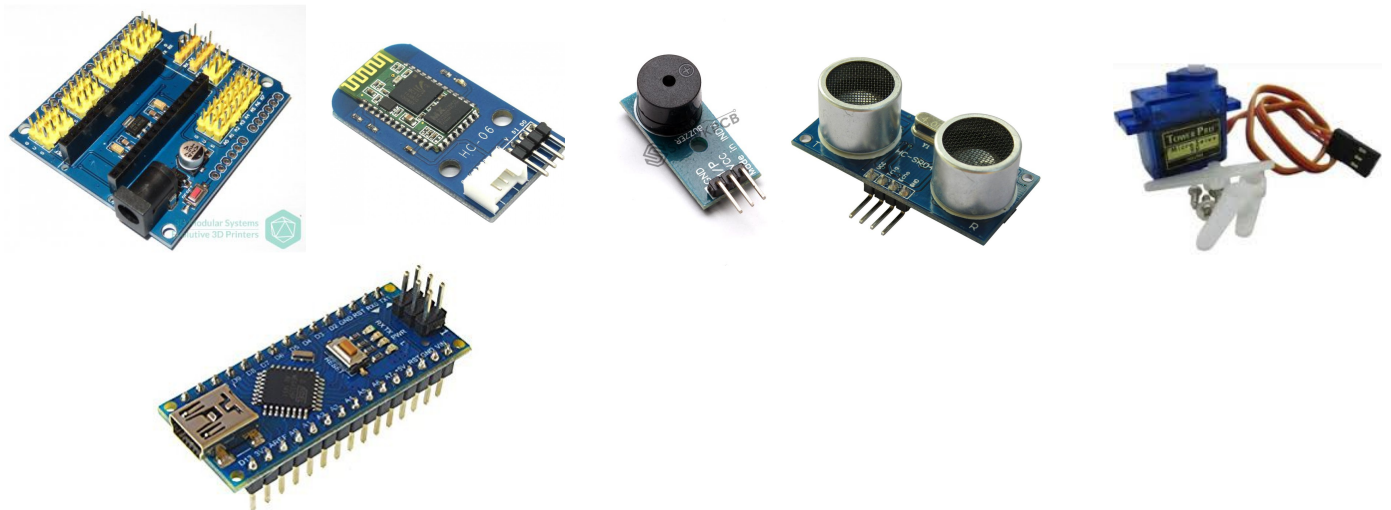
Le corps du robot est donc composé de:

- Sa tête, qui est le carré supérieur avec deux trous pour les yeux.
- Un châssis, permettant de stocker le circuit à l'intérieur.
- 2 roues indépendantes de chaque côté de la tête.
- Un support de pince permettant de rattacher la pince au reste de la tête.
- Une pince composée de deux doigts tournant simultanément et permettant d'attraper des objets.

2 Composants

Notre robot utilise les composants suivants:

- Un module bluetooth permettant de recevoir des informations depuis l'application android.
- 2 servo moteurs à rotation continue indépendants permettant de contrôler les roues.
- 1 servo moteur permettant d'actionner la pince grâce à des rotations d'angles définies.
- 1 buzzer émettant des bruits grâce à des variations de fréquences.
- Une carte arduino nano, plus petite qu'une arduino uno donc plus facile à stocker à l'intérieur du corps du robot.
- Une control board permettant de connecter tous les composants à la carte arduino nano.
- Un boîte de 4 piles en série permettant d'alimenter l'arduino avec une tension de 6,4V
- Un connecteur jack mâle qui vient se brancher sur le port jack femelle de l'arduino pour l'alimenter.



De gauche à droite: Arduino nano expansion board, module bluetooth HC-06, Piezo buzzer, capteur de distance hc-sr01, 1 servo-moteur et 2 servo-moteurs à rotation continue, arduino nano.

3 Réalisation

1 Assemblage

Le nombre de pièces étant relativement faible, nous pouvons nous permettre d'assembler et de démonter le robot comme bon nous semble. Il suffit juste d'enlever sa tête et nous pouvons accéder à l'intérieur, au châssis, très facilement. L'assemblage étant donc facile, nous avons pu nous concentrer tout de suite sur la prise en main des composants. Nous avons donc d'abord travaillé sur chaque composant un à un.

Premièrement, nous avons connecté les servo moteurs. Nous avons réalisé quelques fonctions simples comme par exemple avancer, reculer, tourner à droite, tourner à gauche..

Nous avons ensuite connecté et configuré un module bluetooth HC-06 afin de recevoir les informations qui seront envoyées par l'application android.

Nous avons ensuite connecté les capteurs de distance, permettant au robot de détecter les obstacles en face de lui.

Puis nous avons utilisé un buzzer pour permettre à notre arduibot de produire des sons.

Nous avons réalisé le circuit sur une plaque à trou miniature, puis nous avons compacté les fils au maximum à l'aide d'élastiques et nous avons collé le tout à une paroi interne de la tête du robot afin de pouvoir le fermer.

2 Code

Nous avons donc créé un programme dépendant des valeurs envoyées par l'application android. A chaque bouton de l'application correspond une valeur qui, lorsqu'elle est envoyée à l'arduino, déclenche une boucle séparée du reste du programme à l'aide d'un if. Par exemple, l'envoi d'un « 1 » par l'application entraîne la fonction avancer. L'envoi d'un 2 entraîne la fonction reculer, etc. Ainsi, nous avons pu gérer les différents cas et développer plusieurs modes différents.

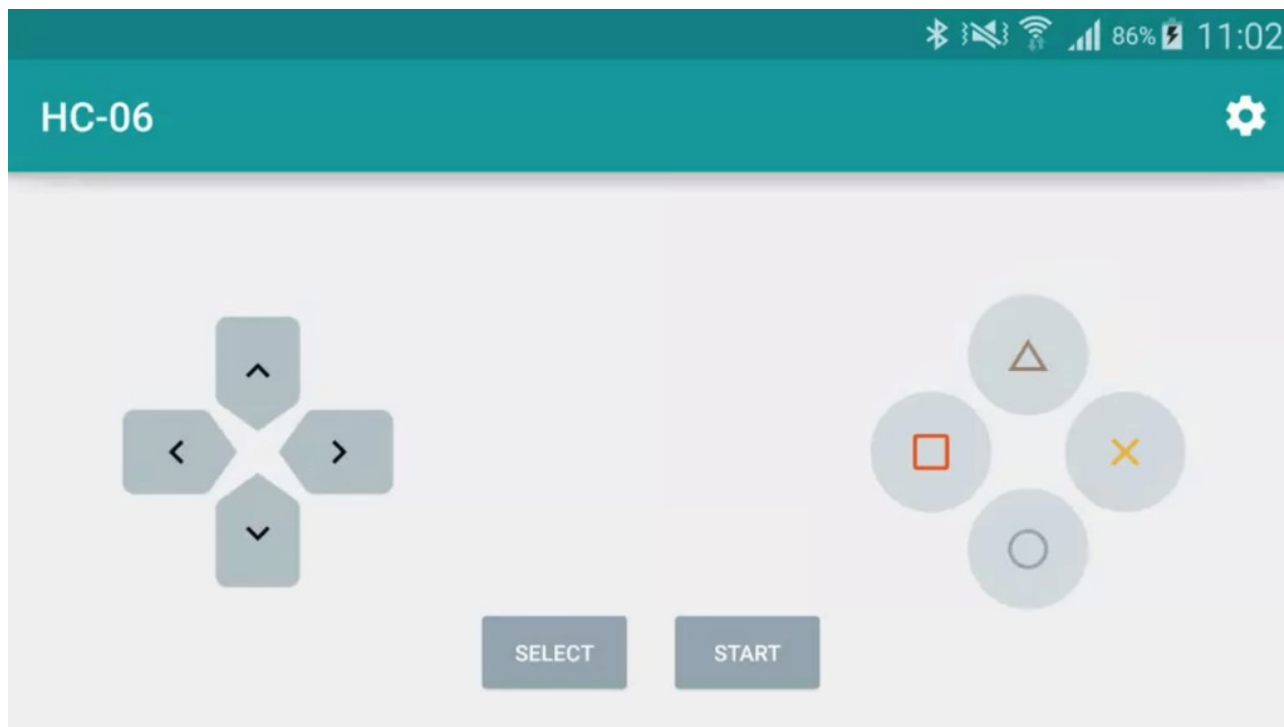
```
app | Arduino 1.6.5
Fichier Édition Croquis Outils Aide

app pitches.h

void loop() {

  //CONTROLLER MODE
  if (mySerial.available())
  {
    command=(mySerial.read());
    if (command=="0") {
      servo1.write(88);
      servo2.write(88);
      Serial.println("pause");
      delay(500);
      autonomous = false;
    }
    if (command=="1") {
      servo1.write(140);
      servo2.write(0);
      Serial.println("front");
      delay(100);
      autonomous = false;
    }
    if (command=="2") {
      servo1.write(40);
      servo2.write(180);
      Serial.println("back");
      delay(500);
      autonomous = false;
    }
    if (command=="3") {
      servo1.write(0);
      servo2.write(0);
      Serial.println("left");
      delay(500);
      autonomous = false;
    }
    if (command=="4") {
      servo1.write(180);
      servo2.write(180);
      Serial.println("right");
      delay(500);
      autonomous = false;
    }
  }
}
```

Pour recréer le programme, il faut donc d'abord configurer les valeurs qu'envoie l'application en cliquant sur le rouage en haut à droite. Les nôtres sont les suivantes:



Flèche vers le haut:	envoie la valeur « 1 »,	= avance
Flèche vers le bas:	envoie la valeur « 2 »,	= recule
Flèche vers la gauche:	envoie la valeur « 3 »,	= tourne à gauche
Flèche vers la droite:	envoie la valeur « 4 »,	= tourne à droite
Bouton select:	envoie la valeur « 0 »,	= arrêt
Bouton carré:	envoie la valeur « 5 »,	= serrer la pince
Bouton triangle:	envoie la valeur « 6 »,	= desserrer la pince
Bouton croix:	envoie la valeur « 7 »,	= joue la musique 1
Bouton rond:	envoie la valeur « 8 »,	= joue la musique 2
Bouton start:	envoie la valeur « 9 »,	= active le mode autonome

Nous avons également développé un mode autonome, utilisant uniquement les capteurs de distance. L'arduibot mesure la distance qui le sépare d'un obstacle devant lui à l'aide de ses capteurs et, quand la distance devient trop faible, il change de direction en ayant un comportement aléatoire. Le comportement paraît aléatoire car nous avons en fait créé 8 façons qui permettent à l'arduibot d'éviter l'obstacle et qui lui donnent donc un côté spontané dans sa démarche. Nous avons fait une moyenne de 5 mesures de distance afin d'empêcher une réaction lors du passage rapide d'un objet ou d'une personne qui ne constitue donc pas réellement un obstacle.


```
app | Arduino 1.6.5
Fichier Edition Croquis Outils Aide

app | pitches.h

if (allReads/5 < 150){
  int randomNumber = random(8);
  if (randomNumber == 1){
    Serial.println("turn right");
    servo1.write(180);
    servo2.write(180);
    delay(200); // delay so that the robot has time to turn far away from obstacle
  }
  if (randomNumber == 2){
    Serial.println("turn left");
    servo1.write(0);
    servo2.write(0);
    delay(200); // delay so that the robot has time to turn far away from obstacle
  }
  if (randomNumber == 3){
    Serial.println("back up");
    servo1.write(80);
    servo2.write(80);
    delay(200); // delay so that the robot has time to turn far away from obstacle
    servo1.write(0);
    servo2.write(180);
    delay(1000);
  }
  if (randomNumber == 4){
    Serial.println("long right");
    servo1.write(180);
    servo2.write(180);
    delay(500); // delay so that the robot has time to turn far away from obstacle
  }
  if (randomNumber == 5){
    Serial.println("long left");
    servo1.write(0);
    servo2.write(0);
    delay(500); // delay so that the robot has time to turn far away from obstacle
  }
}
```

```
app | Arduino 1.6.5
Fichier Edition Croquis Outils Aide

app | pitches.h

/* *****
 * Public Constants
 * ***** */

#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 76
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 195
```

Pour le buzzer, nous avons commencé par définir une banque de fréquences correspondant à différentes notes, puis nous avons définies des mélodies et des bruitages que l’arduibot jouera dans certains cas précis.

5 Problèmes rencontrés au cours du projet

Nous avons eu de nombreux problèmes au cours du développement du projet.

- nous utilisions une control board pour carte arduino nano qui n'était conçue que pour y connecter des servo-moteurs (la control board n'avait donc que des prises mâles), or nous avions différents capteurs et composants à connecter en femelle.

Nous avons donc du utiliser une board en supplément que nous avons collé à l'intérieur de la paroi interne de l'arduibot et où nous avons réalisé un mini circuit permettant à notre programme de fonctionner.

- nous avons eu de nombreux problèmes lors de l'impression 3d des pièces car certaines d'entre elles étaient creuses ou trop fines, et l'imprimante qui a été mise à notre disposition ne pouvait réaliser que des formes pleines et épaisses.

Nous avons donc du modifier certains fichiers sur place, et dans certains cas modifier post impression les pièces à l'aide de fraiseuses ou grâce à différents limages. Tout cela a été réalisé grâce aux outils du Fablab.

- nous voulions vraiment que l'arduibot soit contrôlable par joystick car c'était un de nos premiers objectifs, mais nous avons dû y renoncer car il aurait fallu utiliser un joystick filaire ce qui détruisait tout l'intérêt sans fil de notre projet. Nous aurions pu utiliser un joystick sans fil avec un autre module bluetooth HC-05 pour envoyer des informations, mais nous avons préféré abandonner cette idée pour nous concentrer sur le développement du code restant.

- nous souhaitions que l'arduibot effectue ses mouvements tout en produisant des bruits de robot, or nous n'avons pas réussi à obtenir les deux simultanément. Notre code étant composé à base de boucles et de if afin de traiter les informations envoyées par l'application, la juxtaposition d'une commande d'un servo-moteurs et d'un envoi de tension au buzzer se faisait obligatoirement avec un certain délai. Nous avons donc soit d'abord tout le mouvement, puis ensuite tout le son, soit tout le son puis ensuite tout le mouvement. Notre code se présentait donc comme suit :

action A

delay

Action B

delay

or nous aurions aimé qu'il soit:

action A + action B

delay

Nous n'avons donc pas réussi à obtenir le deuxième schéma d'action car le module bluetooth envoyait des valeurs fixes ce qui ne nous permettait pas de produire deux actions différentes en même temps.

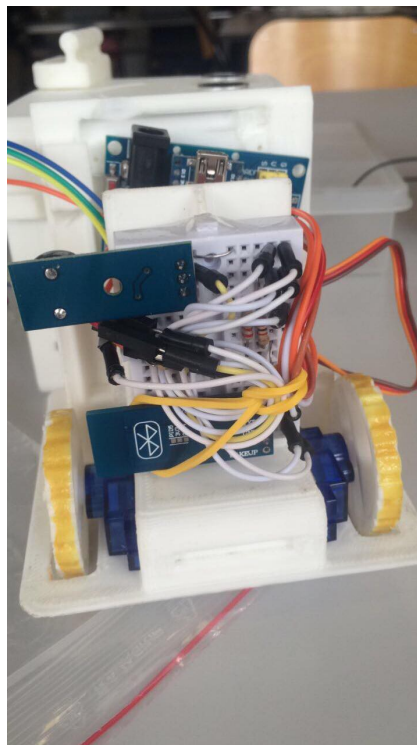
Nous avons donc trouvé qu'avoir des bruits de robots dénués de mouvement ne représentait pas un grand intérêt, et nous avons donc décidé de les remplacer par une fonction « boîte à musique » permettant à notre arduibot de jouer des airs reconnaissables, comme par exemple la marche

impériale, zelda, still dre.. (Ajouter des sons est très facile car nous avons développé une galerie de notes prédéfinies, donc il serait très rapide de rajouter de nouvelles musiques ou des bruits de robots par le futur).

image

-nous avons aussi eu de multiples problèmes de connexion sur notre mini circuit à l'intérieur du robot, car nous n'avions pas assez de câbles femelle mâle pour connecter en une fois la control board aux composants. Nous avons donc du utiliser 2 fils connectés entre eux pour chaque connectique, ce qui s'est vite transformé en un joli méli mélo une fois la control board remplie. De plus, nous ne disposions que d'un espace très restreint à l'intérieur du robot pour stocker tous les câbles.

Nous avons donc été contraints de créer des boucles de fils et de les sécuriser à des éléments fixes sur les parois afin de compacter le tout et pouvoir fermer le corps de l'arduibot. Cependant, quelques minutes de transport ont suffi à déconnecter plusieurs câbles dans le réseau de boucle, ce qui nous a contraint à recommencer une nouvelle fois.



-nos roues ayant été réalisées sur l'imprimante 3d, elles n'étaient parfaitement lisse et n'adhéraient donc sur quasiment aucune surface. Le robot patinait et n'avancait donc presque pas.

Nous avons donc utilisé une résine collante afin de créer une sorte de pneu et fournir aux roues une meilleure adhérence.

6 Améliorations futures

Le premier point que nous aurions aimé améliorer avec plus de temps est la coordination de deux événements, le son et le mouvement, car cela aurait rendu notre arduibot encore plus amusant et attachant.

Nous aurions aussi aimé développer une application plus complète avec, à défaut d'avoir un joystick physique, au moins un joystick sur l'application afin de pouvoir déplacer notre robot dans toutes les directions au lieu de seulement 4 avec notre application actuelle.

De plus, nous aurions voulu améliorer son apparence en limant certaines pièces qui ont été mal finies ou trop abîmées lors des multiples transports. Nous aurions aussi voulu le peindre dans une couleur attirante afin de le rendre plus voyant et plus chaleureux qu'une couleur blanche neutre.

Pour finir, nous aurions aimé créer d'autres arduibot afin de développer des programmes d'interactions, comme par exemple des combats de sumo ou des courses de vitesse et renforcer son côté social.

Conclusion:

Ce fut un projet très enrichissant car nous avons acquis de nombreuses compétences comme le travail sur le long terme ou encore la planification, mais aussi des connaissances techniques comme l'utilisation d'une imprimante 3d ou la manipulation d'outils de fraisage. Le produit fini est à la hauteur de ce que nous esperions, même si certaines modifications et certains sacrifices ont été requis. Nous sommes malgré tout fier de notre résultat final, car nous avons réussi à contourner nos problèmes avec les moyens dont nous disposions et à développer de nouvelles fonctionnalités. Nous nous sommes aussi beaucoup amusés lors des différents tests, le plus mémorable étant sans doute la première avancée sans connexion à l'ordinateur.



Remerciements:

Nous tenons à remercier nos professeurs Mr. Masson et Mr. Ferrero pour leurs conseils sur le développement du projet et sur l'utilisation des composants, ainsi que le Fablab pour son aide précieuse sur la création et le perfectionnement des pièces ainsi que sur de nombreuses idées d'améliorations.