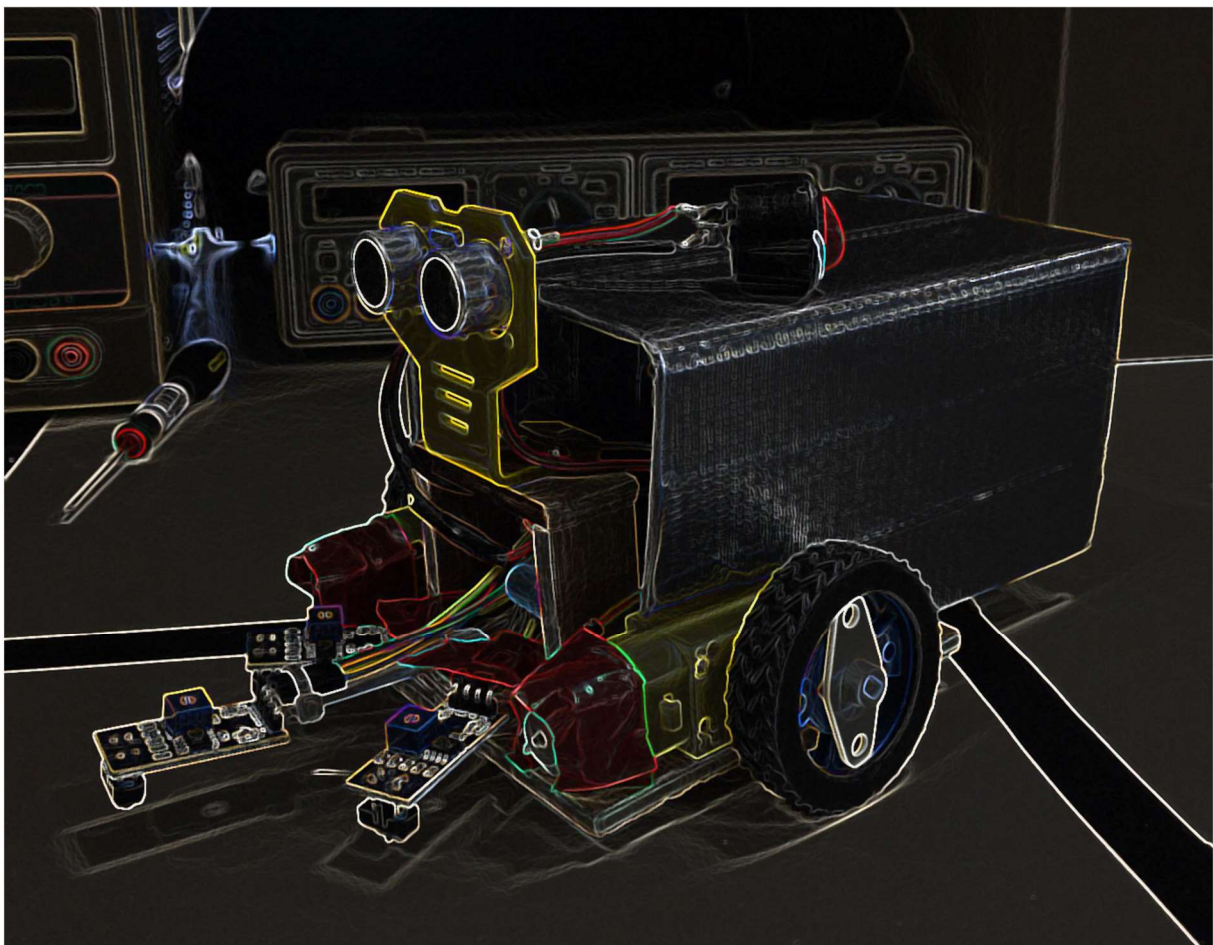


RAPPORT DE PROJET ROBOTLABYRINTHINATOR



SOMMAIRE

Introduction

- I) Matériel et ressources utilisées
 - 1) Le matériel
 - 2) Le rendu final du projet

- II) Fonctionnement du robot

- III) Planning et difficultés rencontrées
 - 1) Planning et diagramme de Gant
 - 2) Les difficultés rencontrées

Conclusion

Bibliographie

Introduction

Nous avons fait ce projet dans le cadre du cours d'électronique arduino de deuxième année de cycle préparatoire intégré à l'école Polytech Nice-Sophia. Nous avons eu un peu plus de trois mois pour réaliser ce projet de l'idée jusqu'au prototype. Le but de ce projet était de nous donner le goût d'innover dans les nouvelles technologies, de nous former au travail en équipe et de nous mettre dans les conditions dans lesquelles nous serons durant notre métier d'ingénieur.

N'ayant pas trouvé de sujet qui nous intéressaient sur Internet, nous avons décidé de réfléchir à des événements de notre vie qui nous ont plus ou moins marqués tout les deux. Il s'avère que nous avons un souvenir chacun concernant les labyrinthes. L'un de nous deux était restés coincés entre deux portes car il ne connaissait pas la réponse à la question « En ... j'ai découvert l'Amérique ». L'autre était resté coincé dans un labyrinthe jusqu'à ce qu'on lui dise qu'il fallait utiliser la règle de la main droite.

Et c'est sur cette dernière règle qu'aït né le Robotlabyrinthinator. En effet, nous avons conçus un petit « robot chercheur » avec deux modes d'utilisation possible. Le premier mode est le mode automatique qui permet au robot de trouver tout seul la sortie de n'importe quel labyrinthe sans se perdre. Il utilise la règle de la main droite, règle infallible pour sortir d'un labyrinthe. Le deuxième mode est manuel et c'est à nous de le télécommander.

Ainsi, la problématique de notre sujet est : Comment sortir à coup sûr d'un labyrinthe ?



1) Matériel et ressources utilisées

1) Le matériel

En ce qui concerne notre robot, nous pouvons découper en trois parties le matériel que nous avons utilisé.

Premièrement, il y a la structure du robot et du labyrinthe, qui se compose :

- D'un support en bois où reposent les composants électroniques,
- D'une coque en carton, scotchée avec du scotch gris de professionnel, donnant un aspect industriel au robot,
- D'une demi-balle de ping-pong, sous le support en bois, pour assurer la stabilité du robot et éviter qu'il ne penche en avant ou en arrière,
- D'une grande planche fine en bois,
- De scotch noir d'électricien acheté à Castorama.

Ensuite, il y a la partie la plus importante de notre robot, la partie électronique qui compte de nombreux composants :

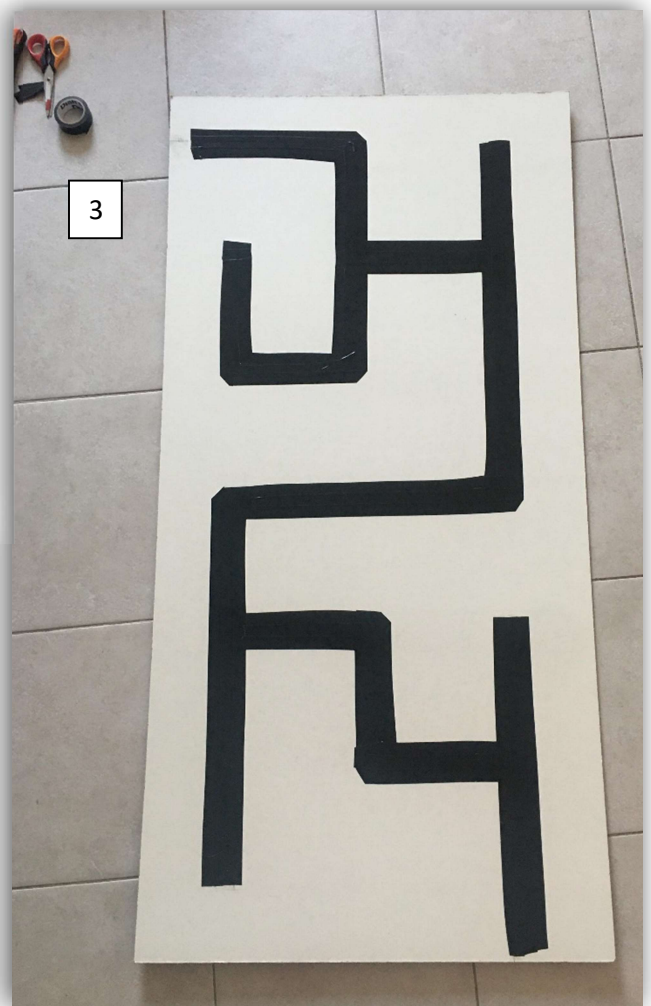
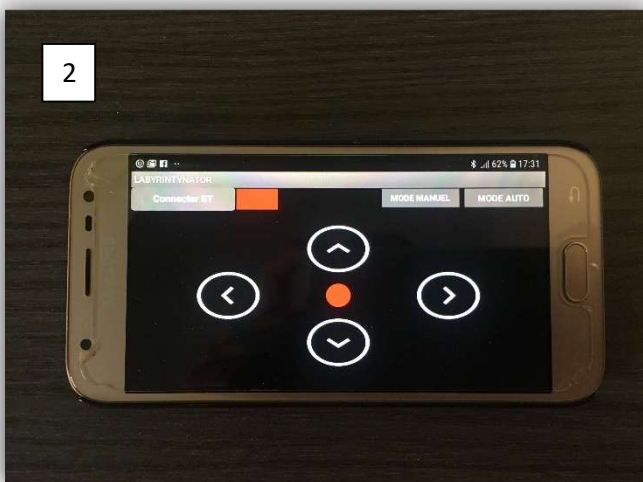
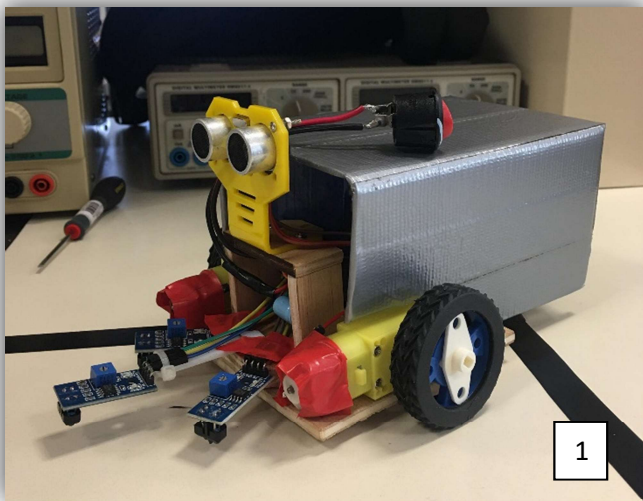
- Deux moteurs CC 5V où sont accrochées les deux roues de notre robot,
- Une carte Arduino UNO,
- Un support à pile comprenant 4 piles de 1,5V soit un total de 6 V
- Un quadruple demi-pont en H qui permet de pouvoir activer les deux moteurs en même temps et de les faire aller dans le même sens (permettant ainsi au robot d'avancer droit),
- Un capteur de distance capable de détecter la présence d'obstacles sur le chemin,
- Trois détecteurs de lignes qui, au contact d'un scotch isolant d'électricien absorbant les infrarouges, s'activent,
- Un bouton poussoir ON-OFF qui permet d'alimenter le robot ou de l'éteindre,
- Une petite plaque à trous que l'on a isolée et où sont branchés la plupart des fils.

Pour finir, il nous reste la partie Bluetooth qui comprend un module Bluetooth câblé sur la petite plaque à trous. Ayant tous les deux des iPhones, il a fallu que l'on se procure un téléphone Samsung pour pouvoir communiquer avec notre module Bluetooth. De plus, nous avons créé une application sur le site du MIT app Inventor.

2) Le rendu final du projet

Dans notre rendu final, ce n'est pas forcément le matériel que nous avons pris au départ qui s'y trouve. En effet, il nous a été arrivé de changer certaines pièces notamment les moteurs avec les roues, les détecteurs de lignes, les piles, les fils et le module bluetooth.

C'est avec tout ce matériel que nous avons maintenant ce rendu final :



- 1 Robot chercheur
- 2 Télécommande de guidage manuel (créée avec MIT app Inventor)
- 3 Labyrinthe

11) Fonctionnement du robot

1) Programme

Le programme final du Robot est le fruit de la combinaison astucieuse des programmes du mode Manuel et du mode Automatique. Nous allons ici en aborder les principaux aspects.

```
void setup() {  
  Serial.begin(9600);  
  
  //bluetooth  
  BlueT.begin(9600);  
  delay(200);  
  
  //deplacements  
  pinMode(ENA,OUTPUT);  
  pinMode(ENB,OUTPUT);  
  pinMode(IN1,OUTPUT);  
  pinMode(IN2,OUTPUT);  
  pinMode(IN3,OUTPUT);  
  pinMode(IN4,OUTPUT);  
  analogWrite(ENA,0);  
  analogWrite(ENB,0);  
  
  //sens rotation moteur D  
  digitalWrite(IN1,HIGH);  
  digitalWrite(IN2,LOW);  
  
  //sens rotation moteur G  
  digitalWrite(IN3,HIGH);  
  digitalWrite(IN4,LOW);  
  
  //capt distance  
  pinMode(trig,OUTPUT);  
  pinMode(echo,INPUT);  
}
```

```
String lireBT(){  
  int n;  
  char cBT;  
  String rBT="";  
  String rBT2="";  
  if (BlueT.available()){  
    if ((char)BlueT.read()=='X'){  
      while (true){  
        cBT = (char)BlueT.read();  
        if (cBT=='X') {  
          break;  
        }  
        rBT+=cBT;  
      }  
    }  
    Serial.println(rBT);  
    return rBT;  
  }  
}
```

```
void loop() {  
  data=lireBT();  
  
  if(data=="m"){  
    i=2;  
  }  
  if(data=="o"){  
    i=1;  
  }  
}
```

Sur ces photos nous avons, à droite, la déclaration des variables du programme dont, le Bluetooth, les capteurs de lignes, les moteurs, le capteur de distance. Tous les Pins propres à chaque appareil y sont référencés.

A gauche, nous avons le Setup qui initialise le dialogue, le Bluetooth, les moteurs et le capteur de distance.

```
//bluetooth  
#include<SoftwareSerial.h>  
const int RX=12;  
const int TX=13;  
SoftwareSerial BlueT(RX,TX);  
String data;  
int i=0;  
  
//moteur droit  
int ENA=10;  
int IN1=16;  
int IN2=17;  
  
//moteur gauche  
int ENB=11;  
int IN3=18;  
int IN4=19;  
  
//capteur avant, gauche et droit  
int captg=4;  
int capta=3;  
int captD=2;  
int captG=0;  
int captA=0;  
int captD=0;  
  
//capteur de distance  
const int trig=6;  
const int echo=5;  
float distance=0;  
float temps=0;
```

Sur cette image ci-dessous on peut voir la fonction lireBT() qui se charge de récupérer tout ce que reçoit le Bluetooth.

Juste ici le Loop démarre avec le morceau de code qui permet au robot d'initialiser une variables en fonction du mode choisis pour le robot.

Le programme suivra des chemins différents en fonction de la valeur de cette variable.

```

while(i==1){
  captG=digitalRead(captg);
  captA=digitalRead(captA);
  captD=digitalRead(captD);

  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  temps=pulseIn(echo, HIGH);
  distance=temps*0.034/2;
  delayMicroseconds(60);
  if (distance<=10){
    analogWrite(ENA,0);
    analogWrite(ENB,0);}
  else if (captG==0 && captA==0 && captD==1){
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,LOW);    //robot va à DROITE
    analogWrite(ENA,160);
    analogWrite(ENB,0);}

  else if (captG==0 && captA==1 && captD==0){
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,LOW);    //robot va tout droit
    analogWrite(ENA,160);
    analogWrite(ENB,190);}

```

Sur ces images nous avons, à droite, un morceau du mode Auto, on voit que le robot attend de recevoir de nouvelles infos via le Bluetooth et réagit en conséquence.

A gauche, on voit un bout du mode Manuel qui permet au robot d'évoluer seul dans le labyrinthe. On remarque que si le robot fait face à un obstacle a moins de 10cm il s'arrête. C'est comme cela que nous modélisons la fin du labyrinthe.

```

while(i==2){
  data=lireBT();
  if(data=="h"){
    Serial.println("Avance");
    data="a";
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,LOW);
    analogWrite(ENA,160);
    analogWrite(ENB,190);
  }

  if(data=="b"){
    Serial.println("Recule");
    data="a";
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,HIGH);
    digitalWrite(IN3,LOW);
    digitalWrite(IN4,HIGH);
    analogWrite(ENA,160);
    analogWrite(ENB,190);
  }
}

```

A la fin de l'exécution des deux modes on peut réinitialiser la variable qui oriente le programme sur le mode Auto ou Manuel et ainsi changer de mode.

2) Application dédiée au robot



Nous avons mis au point une application, grâce au MIT App Inventor, qui permet de diriger le robot en mode Auto ou Manuel et aussi de changer de mode quand cela est nécessaire.

3) Résultats concrets

Après de nombreux tests, nous avons remarqué que notre programme n'était pas au meilleur de son potentiel car le module Bluetooth a tendance à recevoir des caractères étranges et inutiles en plus de ceux envoyés avec l'application MIT.

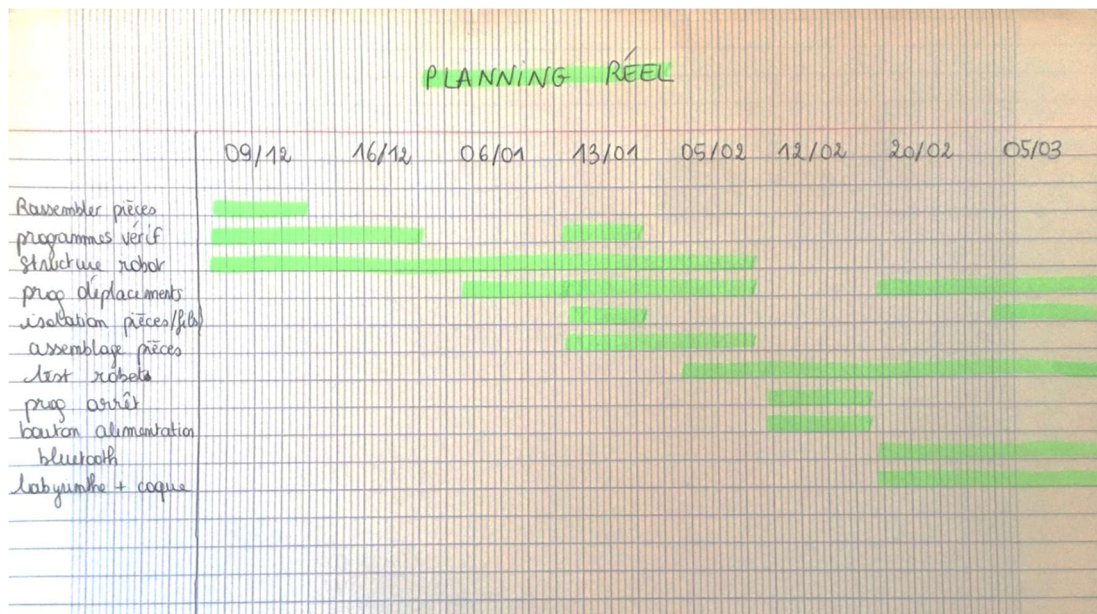
Ces caractères ne sont pas compris par le programme et celui-ci a donc tendance à mal comprendre les ordres que l'application MIT lui envoie, ce qui ralentit ou saccade son exécution.

III) Planning et difficultés rencontrées

1) Planning et diagramme de Gant

Notre planning initial était assez bien réparti et nous laissait du temps les deux dernières séances pour régler les petits détails. Nous devions commencer progressivement par la structure du robot, puis le programme de déplacement et d'arrêt du robot et enfin finir avec le Bluetooth (partie qui nous semblait la plus compliquée).

Cependant, le temps était plus court qu'on ne l'imaginait et il nous restait encore beaucoup de choses à faire lors de la dernière séance... En effet, comme on peut le voir sur le diagramme de Gant suivant, diagramme du déroulement réel de nos séances, il nous restait encore 5 parties non terminées concernant notre projet, principalement le Bluetooth sur lequel nous avons passés plusieurs semaines mais encore l'esthétique du robot ainsi que le programme des déplacements.



2) Difficultés rencontrées

Les principales difficultés rencontrées sont liées au Bluetooth et aux branchements. En effet, nous avons eu beaucoup de mal à faire la connexion Bluetooth entre le téléphone et le module Bluetooth, celui-ci ne nous renvoyait rien alors qu'il était bel et bien connecté. Nous avons dû demander de l'aide, refaire une nouvelle application, ... Nous avons enfin réussi à résoudre le problème trois jours avant l'oral. Le principal défaut de notre robot est son alimentation, il est sous-alimenté ce qui fait qu'il n'active pas tout le temps ses moteurs quand on lui demande notamment pour les demi-tours et également lors du mode manuel avec le bluetooth. En revanche, lorsqu'il est branché à l'ordinateur, tout marche parfaitement. Ce problème étant survenu la dernière semaine de projet, il nous a été impossible de trouver une autre alimentation à brancher.

Conclusion

En conclusion, ce projet qui pour nous aura été une grande première nous a appris que l'organisation et la répartition des tâches ainsi que le travail d'équipe sont les clés du succès d'un projet. Mais ce projet nous aura aussi apporté des connaissances techniques, dans le domaine du bricolage, tel que la soudure, l'utilisation de la disqueuse à métaux, ou de la scie électrique à bois.

Nous avons, de plus, appris à nous responsabiliser face à la charge de travail, tant bien en cours qu'à la maison, que représentait ce projet. Nous l'avons mené à bien et nous en sommes fiers. Fiers de notre petit Robotlabyrinthinator qui a énormément progressé en quelques mois à peine.

Malgré ces prodigieux progrès nous ne sommes pas parvenus au terme de nos attentes initiales.

En effet une grande perspective d'avenir pour notre robot serait non seulement de régler les problèmes d'alimentation qu'il rencontre mais aussi de lui coder une « mémorisation » du parcours accompli dans le labyrinthe pour que celui-ci puisse le refaire d'une traite en évitant les chemins inutiles qu'il aurait emprunter la fois d'avant.

Pour finir, nous souhaitons remercier Mr Masson pour toute l'aide qu'il a pu nous apporter et tout le matériel qu'il nous a fourni tout au long de notre projet.

Bibliographie

Lien de l'image du labyrinthe d'Alice au Pays des Merveilles :

https://www.bing.com/images/search?view=detailV2&id=8BF6DE17CF78A339408BB5065AE2F42C8A6D0335&thid=OIP.JZOZm0T88-GOkDG99IgIPgHaEK&mediaurl=http%3A%2F%2Fi.skyrock.net%2F5500%2F90835500%2Fpics%2F3213866273_1_12_EPbOtdXk.jpg&exph=427&expw=760&q=labyrinthe+alice+au+pays+des+merveilles&selectedindex=0&ajaxhist=0&vt=0&eim=1,6

Site du MIT app Inventor : <http://appinventor.mit.edu/>

Différentes autres sources, aides :

- <https://www.tubefr.com/robot-labyrinthe-solveur.html>
- <https://forum.arduino.cc/index.php?topic=420618.0>
- <http://perso.iut-nimes.fr/fgiamarchi/wp-content/uploads/2011/02/Projet-Line-Maze-Robot.pdf>
- <https://www.maths-cours.fr/methode/algorithmes-de-dijkstra-etape-par-etape/> (non utilisé mais intéressant si l'on souhaite améliorer le robot)

Le site de l'enseignement pour le bluetooth et les moteurs :

<http://users.polytech.unice.fr/~pmasson/Enseignement-arduino.htm>