

FOLLOW-ME ROBOT:

02/03/2022

Ben Salah Rayan

Rapport Général :

Pendant cette huitième semaine nous avons continué la programmation, et nous avons continué les tests du robot ainsi que l'optimisation de celui-ci.

Répartition des tâches:

Durant cette semaine je me suis occupé de terminer le programme fusionné ainsi que de régler les bugs et les choses qui ne marchaient pas, pendant que Neil a terminé le côté esthétique. Nous avons ensuite fait les derniers tests ensemble et avons travaillé sur l'optimisation du robot.

Finalisation du code:

Lundi je suis donc revenu pour terminer de programmer la voiture. J'ai donc pu régler les problèmes de compatibilité des bibliothèques en me basant sur les bibliothèques que Monsieur Masson avait utilisé. J'ai donc modifié notre programme en fonction de celles-ci. J'ai donc ensuite essayé de faire fonctionner la voiture, après téléversement, tout avait l'air de fonctionner. Les moteurs étaient actionnés quand il le fallait, la pixy détectait bien l'objet, et chose qui ne marchait pas avant en corrélation avec le reste, je recevais bien la distance sur mon téléphone par Bluetooth.

```
testFinal
#include<SoftwareSerial.h>
// #include "Adafruit_VL53L0X.h"
#include <Wire.h>
#include <VL53L0X.h>
#include <Pixy2.h>
#define RX 14
#define TX 15
SoftwareSerial BlueT(RX,TX);

//inclure bibliothèque Pixy2
Pixy2 pixy;
VL53L0X sensor;
//Adafruit_VL53L0X lox = Adafruit_VL53L0X();
//caméra
int signature = 0; //initialisation couleur
int x = 0; //position sur x
int y = 0; //position sur y
int width = 0; //largeur=0
int height = 0; //hauteur=0
int area = 0; //aire de l'objet
int Xmin = 95; //limite Xmin au pixel 95
int Xmax = 200; //limite Xmax au pixel 200
int maxArea = 0; //aire minimum
int minArea = 0; //aire maximum

//moteurA
int ENA = 9; //vitesse moteurA
int IN1 = 4; //sens moteurA
int IN2 = 5; //sens inverse moteurA
//moteurB
int ENB = 10; //vitesse moteurB
int IN3 = 6; //sens moteurB
int IN4 = 7; //sens inverse moteurB

void setup() {
  Serial.begin(115200);
  BlueT.begin(9600);
```

Optimisation:

Notre Robot suivait donc bien notre objet, mais à certains moments il avait un peu de mal ou alors ses déplacements n'étaient pas assez précis. Nous avons donc essayé de réaliser plusieurs tests pour analyser d'où venait le problème. Après de nombreuses minutes à tester et chercher nous avons remarqué une petite différence entre les moteurs, l'un avait un peu plus de pêche que l'autre, d'où le fait qu'en ligne droite la trajectoire se décalait un peu, et en virage également.

Nous avons donc rééquilibré la « puissance » en changeant leur PWM comme ci-contre avec les valeurs précédentes indiquées en commentaire.

```
testFinal
digitalWrite (IN1,HIGH);
digitalWrite (IN2,LOW);
analogWrite (ENA, 90);
digitalWrite (IN3,HIGH);
digitalWrite (IN4,LOW);
analogWrite (ENB, 95);//90 avant
}
void freiner(){
    digitalWrite (IN1,HIGH);
    digitalWrite (IN2,LOW);
    analogWrite (ENA, 0);
    digitalWrite (IN3,HIGH);
    digitalWrite (IN4,LOW);
    analogWrite (ENB, 0);
}
void droite(){
    digitalWrite (IN1,HIGH);
    digitalWrite (IN2,LOW);
    analogWrite (ENA, 45);
    digitalWrite (IN3,HIGH);
    digitalWrite (IN4,LOW);
    analogWrite (ENB, 95);//90 avant
}
void gauche(){
    digitalWrite (IN1,HIGH);
    digitalWrite (IN2,LOW);
    analogWrite (ENA, 90);
    digitalWrite (IN3,HIGH);
    digitalWrite (IN4,LOW);
    analogWrite (ENB, 50);// 45 avant
}
void arret(){
    analogWrite (ENA, 0);
    analogWrite (ENB, 0);
}
```

Recablage et roue folle :

Lors des tests la rallonge s'est débranchée, nous avons donc revissé les bouts de fils dans les emplacements de part et d'autre de telle sorte à ce qu'ils ne bougent plus.

De plus nous avons remarqué un petit blocage au niveau de la roue folle qui empêchait en partie que le mouvement du robot soit précis. Nous avons donc graissé celle-ci avec un produit semblable au WD-40.

Réparation de la fonction d'arrêt:

Depuis le départ notre robot avait du mal à s'arrêter, voire ne s'arrêtait pas du tout lorsqu'il était proche de l'objet et fonçait dedans. Nous avons essayé de régler le problème en modifiant les aires minimum et maximum que la pixy calculait mais en vain. J'ai donc pensé, maintenant que les programmes étaient fusionnés, à utiliser le détecteur de distance pour réaliser cet arrêt à la distance voulue, comme cela devait être fait de base avant changements et complications. J'ai donc ajouté une condition qui disait que :
Soit D la variable stockant la distance, si $D < 20\text{cm}$: on s'arrête.

```

Serial.print(height);
Serial.print(" | area: ");
Serial.print(area);
Serial.print(" | maxArea: ");
Serial.print(maxArea);
Serial.print(" | minArea: ");
Serial.print(minArea);
Serial.println(" ");
delay(200);
if(signature == 1){

    if (x < Xmin){
        Serial.println(" x < Xmin ");
        gauche();
    }

    else if (x > Xmax){

        Serial.println(" x > Xmax" );
        droite(); //faire fonction gauche
    }
    else if (sensor.readRangeContinuousMillimeters()<200){
        arret();
    }

    else if (minArea< area < maxArea){
        avancer();
    }
}
//on rajoute ça:

else if (sensor.readRangeContinuousMillimeters()<200){
    arret();

/*else if (area >minArea){

    Serial.println( "area < minArea ");
    avancer(); //faire fonction avancer
}

```

La condition était donc de base placée ici, or après de nombreux tests cela ne marchait toujours pas, en effet la voiture fonçait encore dans la balle.

Après avoir recherché pourquoi, je me suis dit que c'est peut-être à cause du fait que le code s'exécute de manière linéaire et que le fait que la fonction avancer s'exécute au-dessus, et que la condition est presque toujours remplie, elle prend peut-être le dessus sur la fonction arrêt qui est en dessous. J'ai donc modifié la position de cette condition d'arrêt en la mettant au-dessus de la condition d'avance comme ci-dessous :

```

if (x < Xmin){
    Serial.println(" x < Xmin ");
    gauche();
}

else if (x > Xmax){

    Serial.println(" x > Xmax" );
    droite(); //faire fonction gauche
}
else if (sensor.readRangeContinuousMillimeters()<200){
    arret();
}

else if (minArea< area < maxArea){
    avancer();
}

```

Nous avons donc testé et cela fonctionnait à merveille, le robot s'arrêtait lorsqu'il s'approchait trop et en plus envoyait la distance au téléphone.

UNE vidéo démonstrative est en lien dans un autre fichier du même dossier ou ici : <https://youtu.be/hnjhxxsvHLo>

Réorganisation des fils:

Nous avons enfin réorganisé les fils et nous les avons scotchés entre eux par classe d'appartenance, ceux qui s'occupe de tel ou tel élément entre eux etc... pour faciliter le dépannage en cas de panne et pour l'esthétique. Enfin pour l'esthétique nous avons scotché les câbles entre eux et sur le robot pour qu'aucun ne dépasse comme montré ci-joint :



