

FOLLOW-ME ROBOT:

09/02/2022

Ben Salah Rayan

Rapport Général :

Pendant cette sixième séance nous avons continué la programmation, et nous avons commencé les tests du robot ainsi que le debuggage .

Répartition des tâches:

Durant cette semaine je me suis occupé des tests de l'émetteur Bluetooth et de l'émetteur laser ainsi que du test du code principal contenant la pixy, j'ai également essayé de résoudre certains problèmes, Neil a lui cherché comment nous pouvions ajouter d'autres fonctionnalités et a réalisé certains tests avec moi.

Test du programme combinant l'émetteur Bluetooth et le détecteur de distance:

Durant ce week-end j'ai donc testé mon programme réalisé lors de la séance précédente, en connectant donc mon téléphone à l'émetteur et en téléversant cette fois-ci le programme sur la carte Uno de notre robot(programme joint ci-dessous).

```
lecture_distance
#define RX 12
#define TX 13
SoftwareSerial BlueT(RX,TX);

Adafruit_VL53L0X lox = Adafruit_VL53L0X();

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

  // wait until serial port opens for native USB devices
  while (! Serial) {
    delay(1);
  }

  Serial.println("Adafruit VL53L0X test");
  if (!lox.begin()) {
    Serial.println(F("Failed to boot VL53L0X"));
    while(1);
  }
  // power
  Serial.println(F("VL53L0X API Simple Ranging example\n\n"));
}

void loop() {
  VL53L0X_RangingMeasurementData_t measure;

  Serial.print("Reading a measurement... ");
  lox.rangingTest(&measure, false); // pass in 'true' to get debug data printout!

  if (measure.RangeStatus != 4) { // phase failures have incorrect data
    Serial.print("Distance (mm): "); Serial.println(measure.RangeMilliMeter);
  } else {
    Serial.println(" out of range ");
  }
  BlueT.print("*D"+String(measure.RangeMilliMeter)+"*");
}
```

J'ai donc testé et cela a très bien marché, résultat ci-dessous :



Mais je me suis rendu compte, en testant, que lorsque l'on débranche la carte de l'ordinateur et qu'on fait fonctionner le tout sur les piles cela ne marche plus. Je n'arrive même plus à me connecter à l'émetteur Bluetooth. J'ai donc tout de suite pensé à un problème d'alimentation, je me suis dit que si ça marchait branché alors c'est peut-être que les composants dont la Carbo2000 sont trop gourmands en énergie et que les piles n'envoient pas assez de « patate ».

Test du programme principal:

J'ai donc par la suite testé le programme principal s'occupant de la pixy et des moteurs ainsi que de la distance. J'ai donc ajouté les bibliothèques nécessaires pour la pixy le détecteur etc... Mais problème rien n'a fonctionné. J'ai donc modifié et le code en changeant certaines fonctions, comme en remplaçant mindist par maxdist, ainsi que les variables de base etc...
(Voir ci-dessous extraits du code que j'ai modifié)

```

projetPIXYfinal
#include <Pixy2.h> //inclure bibliothèque Pixy2
#include "Adafruit_VL53L0X.h"

Adafruit_VL53L0X lox = Adafruit_VL53L0X();

Pixy2 pixy;
//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 dist=0;
int Xmin = 95; //limite Xmin au pixel 95
int Xmax = 200; //limite Xmax au pixel 200
int mindist = 100; //distance minimum
int maxdist=0;

//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); //nombre de baud pour le moniteur série
  pixy.init(); //initialisation Pixy
  pinMode(ENA, OUTPUT); //déclare que ENA est une sortie
  pinMode(IN1, OUTPUT); //déclare que IN1 est une sortie
  pinMode(IN2, OUTPUT); //déclare que IN2 est une sortie
  pinMode(ENB, OUTPUT); //déclare que ENB est une sortie
  pinMode(IN3, OUTPUT); //déclare que IN3 est une sortie
  pinMode(IN4, OUTPUT); //déclare que IN4 est une sortie
  // wait until serial port opens for native USB devices

```

```

projetPIXYfinal
Serial.print(":");
pixy.ccc.blocks[i].print();
x = pixy.ccc.blocks[i].m_x; //
y = pixy.ccc.blocks[i].m_y; //
width = pixy.ccc.blocks[i].m_width; //
height = pixy.ccc.blocks[i].m_height; //
signature = pixy.ccc.blocks[i].m_signature; //

delay(20);
if(signature == 1){

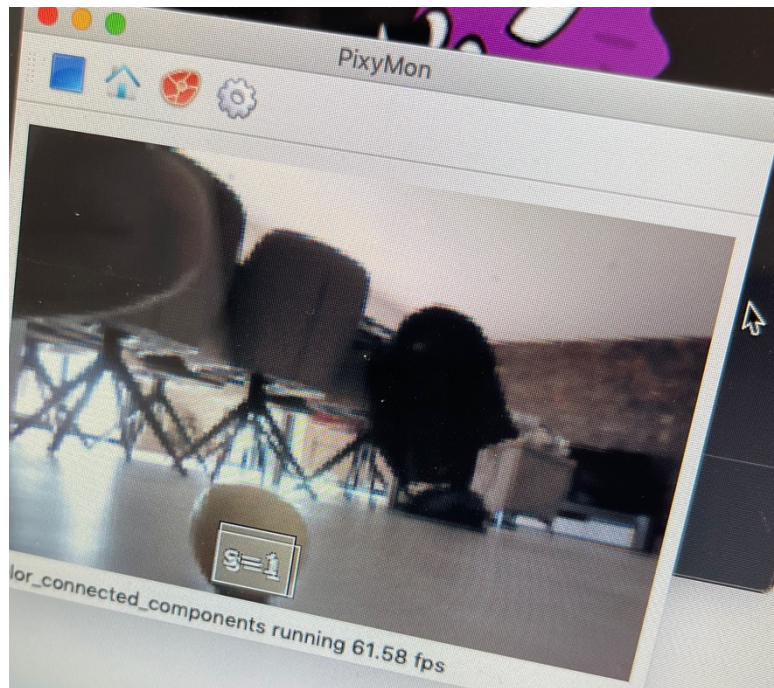
  if (x < Xmin){
    gauche();
  }
  else if (x > Xmax){
    droite();
  }
  else if (measure.RangeMilliMeter > mindist){
    avancer(); //faire fonction avancer
  }
  else if (measure.RangeMilliMeter < mindist){ //j
    freiner();
  }
  else{
    arret(); //fonction arret
  }
}
else{
  arret(); //fonction arret
}
}
}
else {
  Serial.println(" out of range ");
}
}
delay(100);
}
}

```

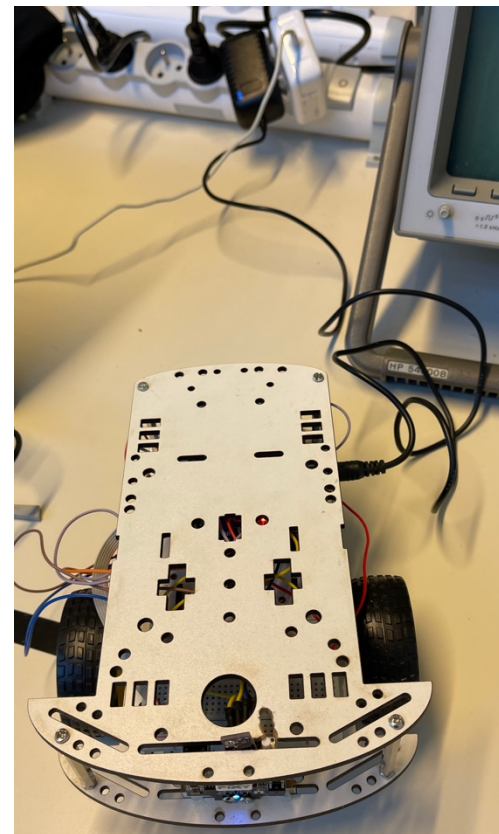
J'ai donc ensuite essayé, ça a l'air de fonctionner mais la voiture est difficile à contrôler, elle est un peu « folle », de plus les moteurs ont du mal, ils ont besoin d'aide. Je remarque aussi qu'il n'y a aucune précision dans les mouvements de la voiture j'ai du mal à analyser d'où cela vient. De plus même lorsqu'elle ne détecte plus l'objet la voiture ne s'arrête pas. J'ai même l'impression qu'elle ne suit pas l'objet, pourtant quand on tient la voiture à la main on a bien les roues qui ne tournent pas à la même vitesse lorsque l'on déplace l'objet de gauche à droite et donc la voiture tourne, mais sur le sol ça ne marche pas.

J'ai également eu la confirmation du gros problème d'alimentation car lorsque l'on débranche de nouveau il n'y a plus rien qui marche,

pourtant la pixy détecte bien l'objet comme on peut le voir ci-dessous.



Pendant la séance j'ai donc parlé de ce problème à Monsieur Masson qui nous a donné une autre alimentation, seul problème c'est un câble qui se branche à une prise donc pour la version de test cela ira en utilisant une rallonge mais il faudra trouver une solution plus durable si l'on développe une version 2.0.



J'ai donc de nouveau testé notre programme Bluetooth/distance et il a enfin fonctionné même en étant déconnecté de l'ordinateur.

Nous avons donc ensuite testé de nouveau le code principal, mais cela n'a encore pas fonctionné, de plus on a remarqué qu'un des moteurs ne marchait pas vraiment vu qu'il n'opposait aucune résistance lorsque l'on pose notre doigt dessus. On teste donc le moteur gauche en le forçant avec des fils pour voir si c'est un problème de notre programme ou de nos branchements, en forçant avec les fils le moteur tourne à fond, ce n'est donc pas un problème de branchements.

Pour confirmer que c'était bien un problème de code, nous avons utilisé une méthode « à l'ancienne » avec un oscilloscope, et on a observé que le moteur était censé recevoir du 240 en PWM, et donc être quasiment constamment à 5V mais là ce n'était pas le cas, donc source du problème identifiée.

Après énormément de temps perdu à essayer de résoudre le problème nous avons enfin réussi à faire un programme qui fonctionnait mais qui nécessitait encore un peu d'améliorations. La voiture fonctionne mais va trop vite et n'a donc pas le temps de repérer l'objet, on a donc réduit les vitesses de rotation.

Un autre problème toujours pas résolu était le fait qu'elle continuait à rouler même sans détecter l'objet. Plus précisément une fois l'objet détecté elle se mettait à rouler mais une fois l'objet hors de sa vue au lieu de s'arrêter, elle continuait, comme si elle restait coincée dans sa boucle de mouvement. On a donc dû résoudre ce problème qui était une modification du code permettant que lors de la non détection elle sorte de sa boucle et exécute la fonction arrêt()).

On a ensuite pris possession d'un nouvel objet servant d'objet suivi car sur le sol cela commençait à devenir compliqué.

Enfin nous avons observé une anomalie lors des tests, on a remarqué que le programme était comme « inversé », c'est-à-dire que lorsque la balle était trop loin le robot s'arrêtait, lorsque la balle était trop proche, presque collée au robot, elle avançait très vite, mais c'était très difficile à voir parce qu'à moyenne distance elle se comportait normalement. Nous avons donc changé dans le programme nos fonctions et surtout les conditions dans nos boucles et en plus d'avoir résolu ce problème cela en a résolu beaucoup d'autres parce que le robot semblait beaucoup plus réactif et précis dans ses mouvements. Lors de cette séance nous avons donc fait une énorme avancée. (Extraits du programme principal ci-dessous)

```
testprojet1
void avancer(){

    digitalWrite (IN1,HIGH);
    digitalWrite (IN2,LOW);
    analogWrite (ENA, 90);
    digitalWrite (IN3,HIGH);
    digitalWrite (IN4,LOW);
    analogWrite (ENB, 90);
}
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, 90);
}
void gauche(){

    digitalWrite (IN1,HIGH);
    digitalWrite (IN2,LOW);
    analogWrite (ENA, 90);
    digitalWrite (IN3,HIGH);
    digitalWrite (IN4,LOW);
    analogWrite (ENB, 45);
}
void arret(){
    analogWrite (ENA, 0);
```

```
testprojet1
Serial.print(minArea);
Serial.println(" ");
delay(20);
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 (area > minArea){

        Serial.println( "area < minArea ");
        avancer(); //faire fonction avncer
    }
    else if (area > maxArea){

        Serial.println("area > maxArea ");
        freiner();
    }
    else{
        arret(); //fonction arret
    }
}
else{
    arret(); //fonction arret
}
}
else{
    arret(); //fonction arret
}
}
```


Et nous sommes très fiers de montrer une première démonstration :

<https://youtu.be/jdbHNYH88h8>