

FOLLOW-ME ROBOT:

23/02/2022

Ben Salah Rayan

Rapport Général :

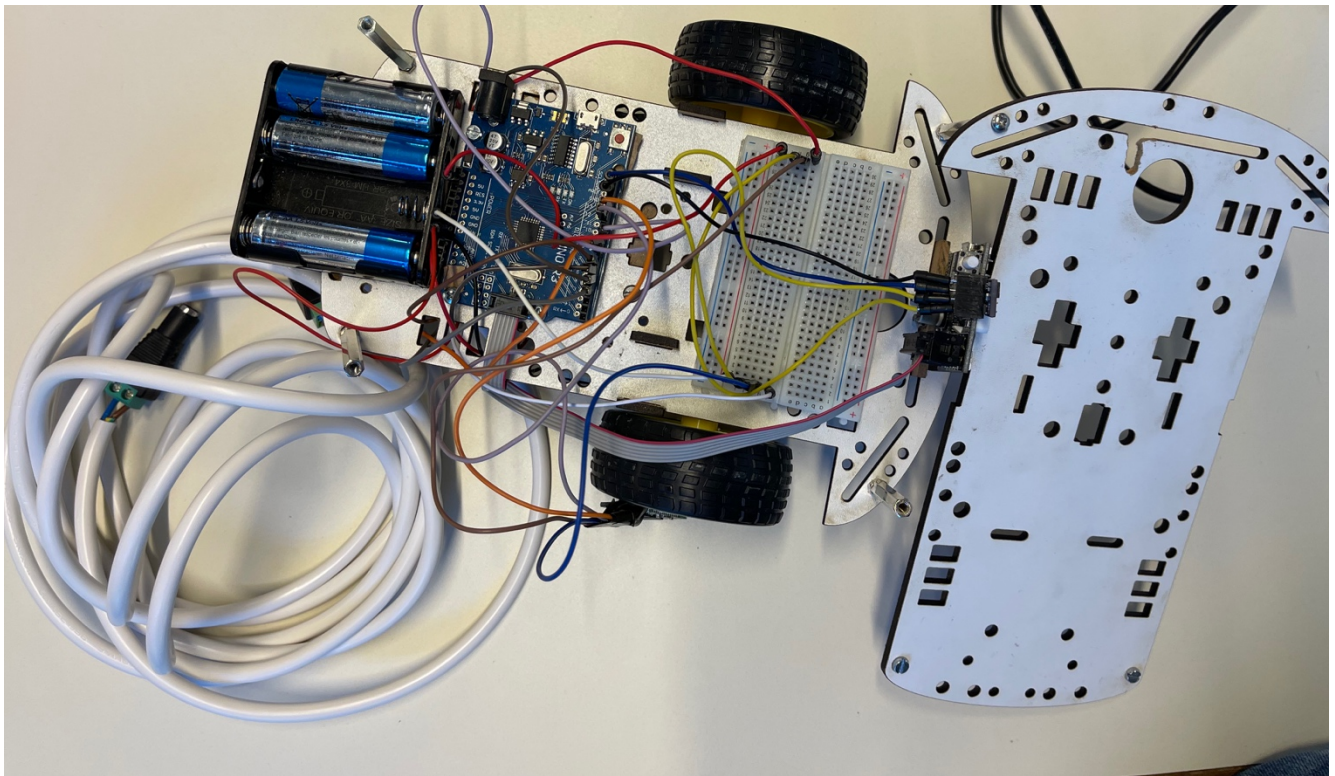
Pendant cette septième séance nous avons continué la programmation, et nous avons continué les tests du robot ainsi que le debuggage.

Répartition des tâches:

Durant cette semaine je me suis occupé de la fusion de programmes, d'une part celui qui s'occupait de la pixy ainsi que du contrôle des moteurs, et d'autre part celui qui s'occupait de l'émetteur Bluetooth et du détecteur de distance. Neil s'est quant à lui occupé du côté esthétique de la Carbo2000.

Création de la rallonge:

Monsieur Masson a eu l'amabilité de ne nous créer une rallonge fait maison, nous lui avons indiqué quelle longueur il nous fallait (pas trop long pour pas trop perdre de « patate »):



Combinaison des deux programmes:

Pendant cette séance j'ai donc essayé de combiner les deux codes de notre Robot, cela a été une tâche rocambolesque.

En effet tout ne s'est pas passé comme prévu.

Comme Monsieur Masson l'avait prévu avant le début des projets, la fusion a « merdouillé » en reprenant ses mots.

En effet lorsque j'ai fusionné plus rien ne marchait, que ce soit le contrôle des moteurs par la pixy ou même la détection de distance et l'envoi de celle-ci au téléphone. J'ai donc essayé de voir si les librairies utilisées : Pixy2.h, SoftwareSerial.h et Adafruit_VL53L0X.h étaient incompatibles entre elles à cause des bus ISP et I2C. Or sachant que softwareSerial et Adafruit marchent ensemble car le programme du détecteur de distance marche, j'ai essayé Software avec Pixy puis Adafruit avec Pixy, et j'ai donc mit en commentaire respectivement tous les éléments de code de chacune des bibliothèques lorsque je les omettais. (Voir ci-joint)

```

TestCodeAvecDist
#include<SoftwareSerial.h>
//include "Adafruit_VL53L0X.h"
#include <Pixy2.h>
#define RX 12
#define TX 13
SoftwareSerial BlueT(RX,TX);

//inclure bibliothèque Pixy2
Pixy2 pixy;
//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);

  pixy.init(); //initialisation Pixy
  pinMode (ENA, OUTPUT); //déclare que ENA est une sortie

```

```

TestCodeAvecDist §
nt IN4 = 7; //sens inverse moteurB

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

  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

  //initialisation module L:: wait until serial port opens for native USB
  /*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 avancer(){
    digitalWrite (IN1,HIGH);
    digitalWrite (IN2,LOW);

```

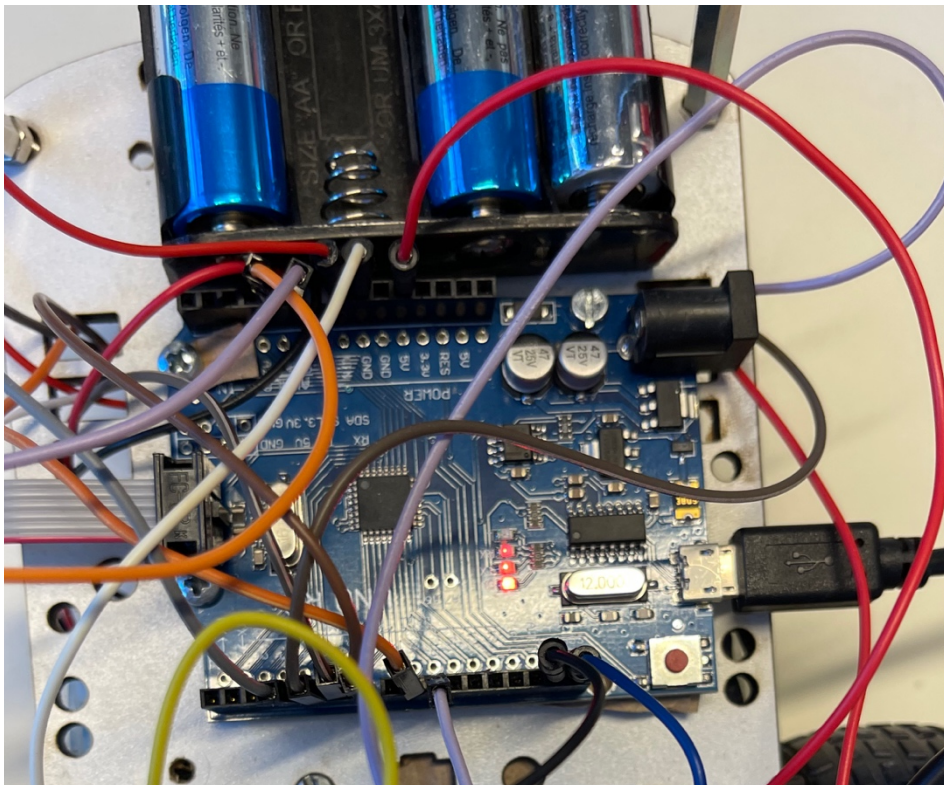
Après Analyse avec Monsieur Masson, il m'a indiqué que c'était donc peut être un problème avec les bus ISP et I2C.

Nous sommes donc repartis avec Monsieur Masson pour un long moment de recherche et de tests sur mes codes et sur les documentations des différents bus. Entre temps Neil a cassé une des plaques de notre Robot, je l'ai donc aidé à retrouver les plans pour effectuer une découpe laser.

Résolution du problème:

Après de longues minutes de recherche, avec Monsieur Masson nous avons trouvé l'erreur qui était en fait une bêtise de ma part, en effet le module Bluetooth était en fait branché sur les prises 12 et 13 or les prises 12 et 13 sont réservées au bus ISP or la pixy est également branchée physiquement avec la broche spéciale au bus ISP directement, il y a donc eu un conflit entre les deux. C'est pour cela que lorsque l'on enlevait l'alimentation du module, on recevait de nouveau des infos de la pixy sur le moniteur série. Le Bus ISP ou SPI en anglais (Serial Peripheral Interface) est utilisé pour communiquer entre des périphériques et la carte sur des courtes distances avec la carte en tant que maître et le périphérique en tant qu'esclave.

J'ai donc changé les branchements en les mettant sur les prises 14 et 15 et j'ai changé le code en accord.



En testant avec le programme de Monsieur Masson sur son PC et ses bibliothèques, cela marchait convenablement, malheureusement sur mon ordinateur avec mon code cela ne marchait toujours pas.

Il m'a donc envoyé son code avec ses bibliothèques et comme la séance se terminait, je n'ai pas encore eu le temps mais je devrais pendant la semaine et jusqu'à la séance prochaine comparer les deux codes et bibliothèques et trouver d'où vient le problème.

Conclusion de la séance:

Ainsi, cette séance m'a paru peu productive surtout comparée aux précédentes mais vu l'étendu des complications qui se sont passés je relativise et regarde le côté positif en pensant que l'on a trouvé une partie de la solution au problème, il restera néanmoins beaucoup de travail pour terminer de le résoudre...