Rapport de séance n°4

Le 09/01/2024

Aimée BOUKAMBA

ROBO3

La séance n°4 est majoritairement axée autour de l'écriture des codes informatiques. Pour le mouvement du robot, le rapport bibliographique annonce une cartographie et la recherche de l'humain le plus proche. Le programme que j'ai écrit ne permet pour l'instant que de se balader aléatoirement en évitant les obstacles.

CODE D'AVANCE DU ROBOT

```
#include <AFMotor.h>
AF_DCMotor motorsG(1);
AF_DCMotor motorsD(2);

#define trigPinAvant 9 // defini les broches du capteur central Avant
#define echoPinAvant 10

#define trigPinDroit A5 // defini les broches du capteur DROIT
#define echoPinDroit A4

#define trigPinGauche A3 // defini les broches du capteur GAUCHE
#define echoPinGauche A2

void setup() {
Serial.begin(9600); // pour afficher des données à l'écran si besoin

pinMode(trigPinAvant, OUTPUT);// envoie les ultra-sons AVANT CENTRAL

pinMode(echoPinAvant, INPUT);// recoit l'echo AVANT CENTRAL

pinMode(trigPinDroit, OUTPUT);// envoie les ultra-sons DROIT
pinMode(echoPinDroit, INPUT);// recoit l'echo DROIT

pinMode(trigPinGauche, OUTPUT);// envoie les ultra-sons GAUCHE
pinMode(echoPinGauche, INPUT);// recoit l'echo GAUCHE
motorsD.run(RELEASE);
motorsG.run(RELEASE);

// initialise la broches Analog In A0 en pin OUTPUT pour faire du son
pinMode(A0, OUTPUT);

delay (3000); // POUR PAS QUE LE ROBOT DEMARRE dès la mise sous
tension

 for (int compteur=0;compteur<=100; compteur=compteur+1) // ALARME
SONORE de démmarage
  {
  digitalWrite(A0, 255);
  delay(1);
  digitalWrite(A0, LOW);
  delay(1);
  digitalWrite(A0, 255);
  delay(1);
  digitalWrite(A0, LOW);
  delay(1);
  }
}

void loop() {

long duration, distance; // MESURE OBSTACLE FRONTAL AVANT
digitalWrite(trigPinAvant, LOW);
delayMicroseconds(2);
digitalWrite(trigPinAvant, HIGH);
delayMicroseconds(10);
digitalWrite(trigPinAvant, LOW);
duration = pulseIn(echoPinAvant, HIGH);
distance = (duration/2) / 29.1;// converti la distance en cm

if (distance < 18)/* TESTE SI OBSTACLE FRONTAL AVANT A MOINS DE 18CM */
{
  Serial.println ("DETECTION OBSTACLE FRONTAL" ); // si besoin d'afficher
infos
  Serial.print ("DISTANCE " );
  Serial.print ( distance);
  Serial.print ( " CM!");
  Serial.println ("CORRIGER TRAJECTOIRE");
  motorsD.run(RELEASE);
  motorsG.run(RELEASE);

  digitalWrite(trigPinGauche, LOW);
  duration = pulseIn(echoPinGauche, HIGH);
  distance = (duration/2) / 29.1;// converti la distance en cm

  if (distance < 40)/* TESTE SI OBSTACLE A GAUCHE A MOINS DE
40 CM */ {
  Serial.println ("DETECTION OBSTACLE à gauche" );
  Serial.print ("DISTANCE " );
  Serial.print ( distance);
  Serial.print ( " CM!");
  Serial.println ("NE PAS TOURNER A GAUCHE-FAIRE MARCHE ARRIERE +
DEMI TOUR");
  delay(500);
  motorsD.run(BACKWARD); // RECULE LENTEMENT
  motorsG.run (BACKWARD);
  motorsG.setSpeed(100);
  motorsD.setSpeed(100);
  delay(1500);

  motorsD.run(FORWARD); // FAIT DEMI TOUR
  motorsG.run (BACKWARD);
  motorsG.setSpeed(200);
  motorsD.setSpeed(200);
  delay(1500); // AJUSTER DUREE du DELAY SELON TENSION ACCU ET
TESTURE DU SOL
  }
  else { // TOURNE A GAUCHE
  motorsD.run(FORWARD);
  motorsG.run (BACKWARD);
  motorsG.setSpeed(200);
  motorsD.setSpeed(200);
  delay(800);// AJUSTER DUREE du DELAY SELON TENSION ACCU ET
TESTURE DU SOL
  motorsD.run(RELEASE);
  motorsG.run(RELEASE);
  delay(1500);
  }
 }
 else { // TOURNE A DROITE
  motorsD.run(BACKWARD);
  motorsG.run (FORWARD);
  motorsG.setSpeed(200);
  motorsD.setSpeed(200);
  delay(800);// AJUSTER DUREE du DELAY SELON TENSION ACCU ET TESTURE
DU SOL
  motorsD.run(RELEASE);
  motorsG.run(RELEASE);
  delay(1500);
  }
}
else { // PAS OBSTACLE DEVANT DONC MARCHE AVANT
// MESURE OBSTACLE A DROITE AVEC AFFECTATION COEFF MOTEURS GAUCHE !
(CMG)
  delay(5);
  long duration, distance; // MESURE OBSTACLE A DROITE
  digitalWrite(trigPinDroit, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPinDroit, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPinDroit, LOW);
  duration = pulseIn(echoPinDroit, HIGH);
  distance = (duration/2) / 29.1;// converti la distance en cm
  if (distance > 13) {
  motorsG.run(FORWARD);
  motorsG.setSpeed(200);
  }
  else {
  float CMG = ((4+(distance-7))/10);
  motorsG.run(FORWARD);
  motorsG.setSpeed(200*CMG);
  }

  for (int compteur=0;compteur<=100; compteur=compteur+1) // ALARME
SONORE OBSTACLE
  {
  digitalWrite(A0, 255);
  delay(1);
  digitalWrite(A0, LOW);
  delay(1);
  digitalWrite(A0, 255);
  delay(1);
  digitalWrite(A0, LOW);
  delay(1);
  }
  delay(5);
  long duration, distance; // MESURE SI OBSTACLE A DROITE
  digitalWrite(trigPinDroit, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPinDroit, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPinDroit, LOW);
  duration = pulseIn(echoPinDroit, HIGH);
  distance = (duration/2) / 29.1;// converti la distance en cm

  if (distance < 40)/* TESTE SI OBSTACLE A DROITE A MOINS DE 40CM
*/ {
  Serial.println ("DETECTION OBSTACLE à droite" );
  Serial.print ("DISTANCE " );
  Serial.print ( distance);
  Serial.print ( " CM!");
  Serial.println ("NE PAS TOURNER A DROITE");
  delay(500);

  long duration, distance; // MESURE SI OBSTACLE A GAUCHE
  digitalWrite(trigPinGauche, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPinGauche, HIGH);
  delayMicroseconds(10);
  }

// MESURE OBSTACLE A GAUCHE AVEC AFFECTATION COEFF MOTEUR DROITS
!(CMD)
  delay(5);
  //long duration, distance; // MESURE OBSTACLE A GAUCHE
  digitalWrite(trigPinGauche, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPinGauche, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPinGauche, LOW);
  duration = pulseIn(echoPinGauche, HIGH);
  distance = (duration/2) / 29.1;// converti la distance en cm
  if (distance > 13) {
  motorsD.run(FORWARD);
  motorsD.setSpeed(200);
  }
  else {
  float CMD = ((4+(distance-7))/10);
  motorsD.run(FORWARD);
  motorsD.setSpeed(200*CMD);
  }
 }
}
```

CODE DE RECONAISSANCE FACIALE EN PYTHON CORRELE AVEC ARDUINO

(Le code résulte d'un travail en commun avec le robot Militech.)

```python
import numpy as np
import serial
import time
import sys
import cv2


#Setup Communication path for arduino (In place of 'COM9')
arduino = serial.Serial('COM9', 9600)
time.sleep(2)
print("Connected to arduino...")

#Importing the Haarcascade for face detection
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

#To capture the video stream from webcam.
cap = cv2.VideoCapture(0)
cv2.namedWindow('img', cv2.WINDOW_NORMAL)

#Read the captured image, convert it to Gray image and find faces
while 1:
    ret, img = cap.read()
    cv2.resizeWindow('img', 500,500)
    cv2.line(img,(500,250),(0,250),(0,255,0),1)
    cv2.line(img,(250,0),(250,500),(0,255,0),1)
    cv2.circle(img, (250, 250), 5, (255, 255, 255), -1)
    gray  = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3)
```

```python
#detect the face and make a rectangle around it.
    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),5)
        roi_gray  = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]

        arr = (y:y+h, x:x+w)
        print (arr)

        print ('X :' +str(x))
        print ('Y :'+str(y))
        print ('x+w :' +str(x+w))
        print ('y+h :' +str(y+h))

# Center of roi (Rectangle)
        xx = int(x+(x+h))/2
        yy = int(y+(y+w))/2
        print (xx)
        print (yy)
        center = (xx,yy)

# sending data to arduino
        print("Center of Rectangle is :", center)
        data = "X{0:d}Y{1:d}".format(xx, yy)
        print ("output = '" +data+ "'")
        arduino.write(data)

#Display the stream.
    cv2.imshow('img',img)

#Hit 'Esc' to terminate execution
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break
```

```ino
#include<Servo.h>

Servo servoVer; //Vertical Servo
Servo servoHor; //Horizontal Servo

int x;
int y;

int prevX;
int prevY;

void setup()
{
  Serial.begin(9600);
  servoVer.attach(5); //Attach Vertical Servo to Pin 5
  servoHor.attach(6); //Attach Horizontal Servo to Pin 6
  servoVer.write(90);
  servoHor.write(90);
}

void Pos()
{
  if(prevX != x || prevY != y)
  {
    int servoX = map(x, 600, 0, 70, 179);
    int servoY = map(y, 450, 0, 179, 95);

    servoX = min(servoX, 179);
    servoX = max(servoX, 70);
    servoY = min(servoY, 179);
    servoY = max(servoY, 95);

    servoHor.write(servoX);
    servoVer.write(servoY);
  }
}

void loop()
{
  if(Serial.available() > 0)
  {
    if(Serial.read() == 'X')
    {
      x = Serial.parseInt();
      if(Serial.read() == 'Y')
      {
        y = Serial.parseInt();
        Pos();
      }
    }
    while(Serial.available() > 0)
    {
      Serial.read();
    }
  }
}
```