03/02/2020

BERIHUETE ALIANNA

Rapport de séance

Après avoir tenté d’utiliser Processing pour repérer la balle avec OpenCV j’ai décidé de passer sur Python car je trouvais beaucoup plus de documentation sur Python que sur Processing.

Grâce à la librairie OpenCV sur python j’ai réussi à détecter une couleur (le jaune, mais on peut changer les valeurs HSV) et à faire réagir les servomoteurs avec Arduino par l’intermédiaire de la communication série.

J’ai d’abord écrit un programme qui après avoir détecté la balle (grâce à la librairie cv2 = OpenCV) dessine un cercle autour de la cible (jaune), trace un trait au centre de l’image et finalement un trait entre le milieu de ce trait et le centre du cercle. Ainsi on calcule la distance entre le centre de l’image et la cible. Lorsque cette distance est à gauche le programme envoie une lettre à la carte et le programme Arduino fait tourner le servomoteur principal vers la gauche (idem pour la droite). Mais ensuite je n’ai pas réussi à faire en sorte que le robot se place précisément devant la balle.

J’ai ensuite trouvé un exemple de code de détection faciale avec lequel le robot suivait parfaitement le visage. J’ai essayé de reproduire la même structure avec le code que j’avais déjà et OpenCV, qui dessinait des cercles autour de la cible. J’ai donc seulement utilisé les coordonnées x et y. Donc, pour chaque cercle tracé, le programme récupère les coordonnées x et y grâce auxquelles il effectue des calculs ;

a=int((2\*x)/2)

b=int((2\*y)/2)

x1=int(a/3.66)

y1=int(b/2.55)

puis envoie la valeur x1 à la carte arduino

ser.write(struct.pack('>BB',x1,y1))

et cette valeur sera stockée dans un tableau et indiquée aux servomoteurs (coordonnée x pour le servomoteur de la base et coordonnée y pour le servomoteur de gauche).

Le code fonctionne bien (surtout lorsque la lumière n’est pas trop forte et que l’on possède un objet assez jaune mais c’est dû à la qualité de la caméra).

Durant la séance de Lundi, on a commencé à compléter le code pour qu’une fois que le robot est face à la balle, il se penche et la ramasse.

Comme j’avais déjà le calcul de la longueur entre le centre de l’image et la cible dans le code Python, j’ai rajouté la condition que la longueur ne soit pas 0 :

length = ((640/2)-(int(x)))

if length!=0 :

a=int((2\*x)/2)

b=int((2\*y)/2)

x1=int(a/3.66)

y1=int(b/2.55)

ser.write(struct.pack('>BB',x1,y1))

et si la longueur est effectivement nulle (donc si la balle est au centre de l’image) on envoie la lettre ‘S’ à Arduino :

if length==0:

a='S'

ser.write(a.encode())

et lorsque Arduino reçoit ‘S’ il fait se pencher le robot.

Voici le code Arduino pour l’instant :

#include <Servo.h>

Servo middle, left, right, claw ;

int data[1];

void setup()

{

Serial.begin(9600);

middle.attach(11);

left.attach(10);

right.attach(9);

claw.attach(6);

}

void loop() {

right.write(30);

int a=1;

int c;

while (a==1){

while (Serial.available()>=2){

for (int i =0; i<2;i++){

data[i]=Serial.read();

}

middle.write(data[0]);

//left.write(data[1]);

}

}

delay(1000);

while(Serial.available()){

c=Serial.read();

if (c=='S'){

a=0;

pencher(30);

}

}

}

void pencher(int p){

for(int i=0;i<80;i++){

right.write(i);}

//left.write(50);

delay(p);

}

Mais ceci ne fonctionne pas.