

# Rapport séance 2

Martelli Gino G1

J'ai commencé à écrire le programme pour le projet en commençant par récupérer les données envoyées par la caméra Pixy, puis par faire bouger les moteurs par rapport à ces données pour que la voiture se dirige vers n'importe quel objet rouge.

## Programmation Arduino :

J'ai d'abord commencé par la récupération des données de la caméra Pixy :

Dans le programme écrit à la séance précédente, je pouvais récupérer les données sous cette forme :

```
Detected 1
    block 0: sig: 1 x: 235 y: 100 width: 158 height: 32 index: 121 age: 255
Detected 1
    block 0: sig: 1 x: 236 y: 101 width: 160 height: 32 index: 121 age: 255
Detected 1
    block 0: sig: 1 x: 236 y: 100 width: 160 height: 34 index: 121 age: 255
Detected 1
    block 0: sig: 1 x: 234 y: 99 width: 152 height: 34 index: 121 age: 255
```

Un bloc est un rectangle crée par Pixy qui correspond à l'objet détecté. Il me faut donc les coordonnées x qui correspondent au centre du rectangle, récupérable avec

« pixy.ccc.blocks[0].m\_x ».

```
50 void testPixy() {
51     //// Get blocks from Pixy2
52     pixy.ccc.getBlocks(true, 1);
53     if (pixy.ccc.numBlocks) {
54         pixy.setLamp(1,0); //Allume les 2 leds Pixy
55         if (pixy.ccc.blocks[0].m_x<150){
56             // Tourne le véhicule à gauche pour centrer l'objet
57             turnLeft();
58         }
59         else if (pixy.ccc.blocks[0].m_x>165){
60             // Tourne le véhicule à droite pour centrer l'objet
61             turnRight();
62         }
63         else {
64             //Le véhicule avance
65             forward();
66         }
67     }
68     else {
69         stop();
70         pixy.setLamp(0,0);}
71 }
```

### Fonctionnement de testPixy() :

La caméra Pixy récupère les blocs (ligne 52) avec 1 comme 2<sup>ème</sup> paramètre qui indique qu'il récupère seulement les objets rouges.

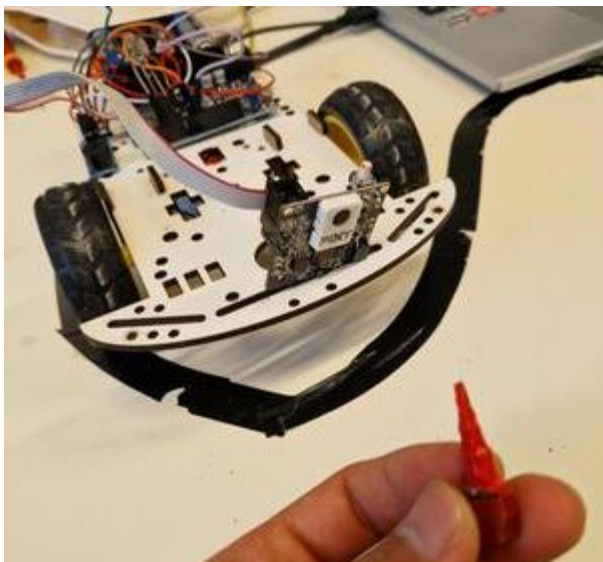
Si le nombre de blocs détectés n'est pas nul (ligne 53), on teste si le premier bloc de la liste est à droite ou à gauche dans le champ de vision de la caméra (ligne 55 et 59). Comme x peut aller de 0 à 315, j'ai défini la gauche comme la zone inférieure à 150 et la droite la zone supérieure à 165. Si l'objet est à droite ou à gauche, le véhicule tourne respectivement à droite ou à gauche (ligne 57 et 61) pour aligner la voiture à l'objet.

Si l'objet n'est ni à droite ni à gauche, il est forcément aligné avec la voiture qui avance donc vers l'objet (ligne 65).

Si la caméra ne détecte aucun objet, la voiture s'arrête (ligne 69). (Dans les prochaines séances, la caméra Pixy sera sur un servomoteur qui balayera une grande zone pour détecter un objet rouge.)

J'ai récupéré les fonctions stop, forward, turnLeft et turnRight qu'a écrit mon camarade pour diriger les moteurs.

On a ensuite testé si la voiture suivait bien les objets rouges. Ça fonctionne très bien ; la voiture se dirige vers l'objet rouge. Même si l'objet est en mouvement, la voiture suit la trajectoire de l'objet.



```
73 void stop(){
74     digitalWrite(IN1,HIGH);
75     digitalWrite(IN2,LOW);
76     digitalWrite(IN3,HIGH);
77     digitalWrite(IN4,LOW);
78     digitalWrite(ENA,0);
79     digitalWrite(ENB,0);
80 }
81
82 void forward(){
83     digitalWrite(IN1,HIGH);
84     digitalWrite(IN2,LOW);
85     digitalWrite(IN3,HIGH);
86     digitalWrite(IN4,LOW);
87     digitalWrite(ENA,90);
88     digitalWrite(ENB,90);
89 }
90
91 void turnRight(){
92     digitalWrite(IN1,LOW);
93     digitalWrite(IN2,HIGH);
94     digitalWrite(IN3,HIGH);
95     digitalWrite(IN4,LOW);
96     digitalWrite(ENA,0);
97     digitalWrite(ENB,90);
98 }
99
100 void turnLeft(){
101     digitalWrite(IN1,HIGH);
102     digitalWrite(IN2,LOW);
103     digitalWrite(IN3,LOW);
104     digitalWrite(IN4,HIGH);
105     digitalWrite(ENA,90);
106     digitalWrite(ENB,0);
107 }
```

J'ai ensuite commencé à rajouter du code pour le sonar et les futures fonctions qu'on va devoir créer.

Je me suis aidé de :

- [docs.pixycam.com](https://docs.pixycam.com)