

Nous avons commencé par essayer de faire communiquer les 2 modules bluetooth et avons demandé l'aide de M.Masson, la communication entre les 2 modules était dans le but que la carte Arduino reliée à la caméra communique les coordonnées de la balle à la carte Arduino reliée au moteur, mais finalement M.Masson nous a dit qu'il valait mieux tout faire sur une seule carte Arduino et tout relier par des fils car le temps de communication par bluetooth aurait été trop long. Et comme il est nécessaire d'avoir de la télécommunication dans le projet, on fera une communication entre la carte Arduino et un téléphone plus tard. Etant donné que nous n'avons plus qu'une seule carte Arduino qui doit tout gérer, je m'occupe de fusionner les programmes du moteur et de la caméra. Une fois la fusion des programmes effectuée, j'ai fait le lien entre la coordonnée de la balle qui nous intéresse et le nombre de Pas que devra faire le moteur pour que le gardien se trouve à la même coordonnée que la balle. Voici donc l'état actuel du programme :

```
#include <Pixy2.h> //inclure bibliothèque Pixy2
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 area = 0; //aire de l'objet
int Xmin = 95; //limite Xmin au pixel 95
int Xmax = 200; //limite Xmax au pixel 200
int minArea = 0; //aire minimum
int maxArea = 0; //aire maximum

const int Pas = 3; // Un pas du moteur
const int Dir = 2; // Sens de la rotation (Direction)
int PosG = 103; // Position du Gardien
int nbPx = 0; // Nombre de Px à ann faire
int nbPas = 0; // Nombre de pas à faire

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

  pixy.init(); //initialisation Pixy

  pinMode(Pas,OUTPUT);
  pinMode(Dir,OUTPUT);
  digitalWrite(Dir,HIGH);
}

void loop() {
  int i;
  pixy.ccc.getBlocks();
  area = width * height; //calcul de l'aire
  maxArea = 4500; //aire max
```

```
minArea = 3500; //aire min
```

```
if (pixy.ccc.numBlocks){  
  Serial.print("Detected "); //ecrit sur le moniteur série qu'un block est détecté  
  Serial.println(pixy.ccc.numBlocks);
```

```
  
  for (i=0; i<pixy.ccc.numBlocks; i++){  
    Serial.print( "block ");  
    Serial.print(i);  
    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;  
    //Infos de la caméra sur le moniteur série Arduino  
    delay(20);  
  }  
}
```

```
digitalWrite(Dir,HIGH); //remise du sens de base  
//réception de la position de la balle par la pixycam ici c'est la coordonnée "y" qui nous  
intéresse  
nbPx = y-PosG; //calcul du nombre de Px "nbPx" à faire en fonction de la position de la  
balle "y" et de celle du gardien "PosG"  
PosG = y; //la position du gardien PosG est désormais celle de la balle "y"  
nbPas = round(nbPx*2.8); //nombre de pas à faire "nbPas" sachant que pour faire un Px il  
faut 2.8 pas
```

```
  
if(nbPas<0){ //changement du sens de rotation  
  digitalWrite(Dir,LOW);  
  nbPas = -nbPas ;  
}  
for(int x=0;x<nbPas;x++){ //rotation de nb pas  
  digitalWrite(Pas,HIGH);  
  delayMicroseconds(500);  
  digitalWrite(Pas,LOW);  
  delay(1);  
}  
}
```

Normalement le programme permet donc au moteur de réagir aux déplacements de la balle et de mettre le gardien en face. Mais pour le tester nous avons besoin d'une structure, l'objectif de la semaine prochaine est donc de débiter la construction de la structure de notre projet.