



Année 2019-2020

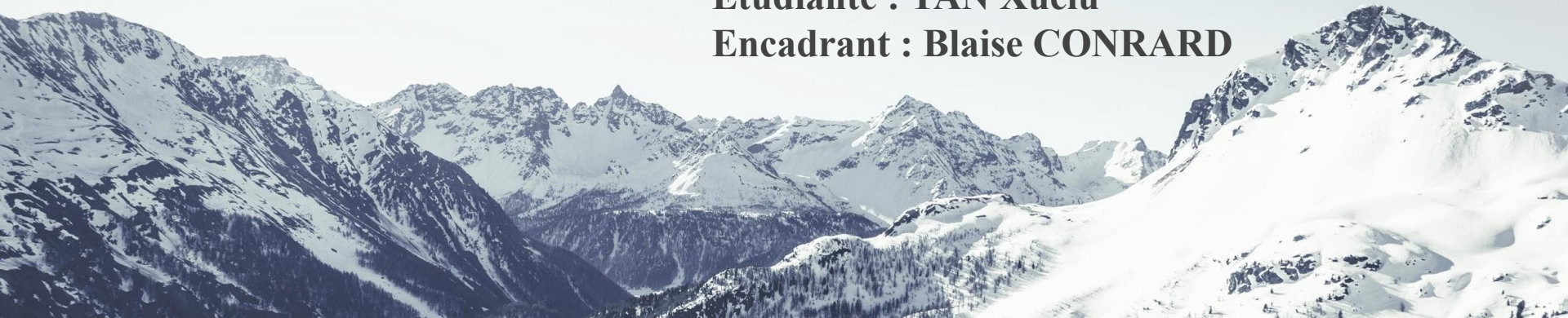


Soutenance de PFE IMA5

Labyrinthe à bille autonome

Etudiante : YAN Xuelu

Encadrant : Blaise CONRARD





Cahiers de charges

- **Objectif**
- **Fonctionnement**
 1. La bille se déplace en douceur
 2. Déterminer les coordonnées de la bille
 3. La bille se déplace selon une trajectoire spécifiée



Sommaire

Caméra

Détection d'objets

Tester avec Arduino

Mécanique

Structure

Labyrinthe

Réalisation

Electronique

Tester les moteurs

Dessiner la carte

Carte électronique

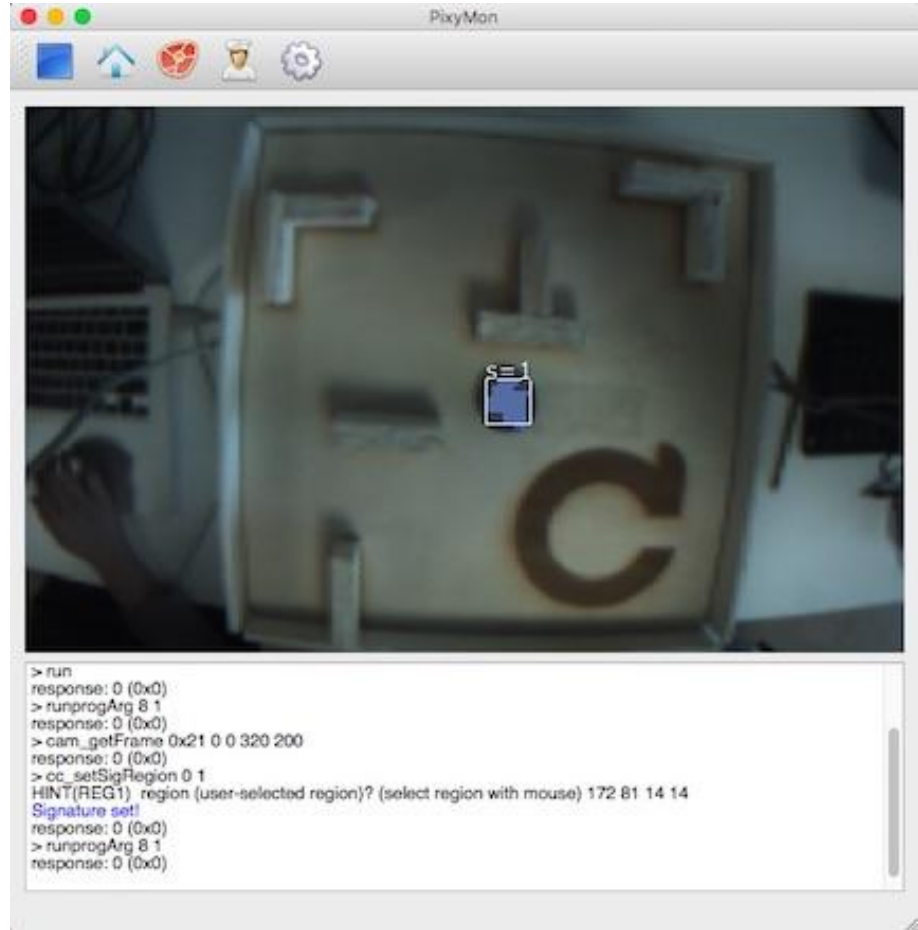
Programmation

Contrôleur P

Trajectoire spécifiée

Détection d'objets

- Logiciel :
Pixymon
- Fonction :
Détecer et localiser l'objet en couleur appris.
- Principe :
Algorithme de filtrage de teinte



Tester avec l'Arduino

Méthode de communication :

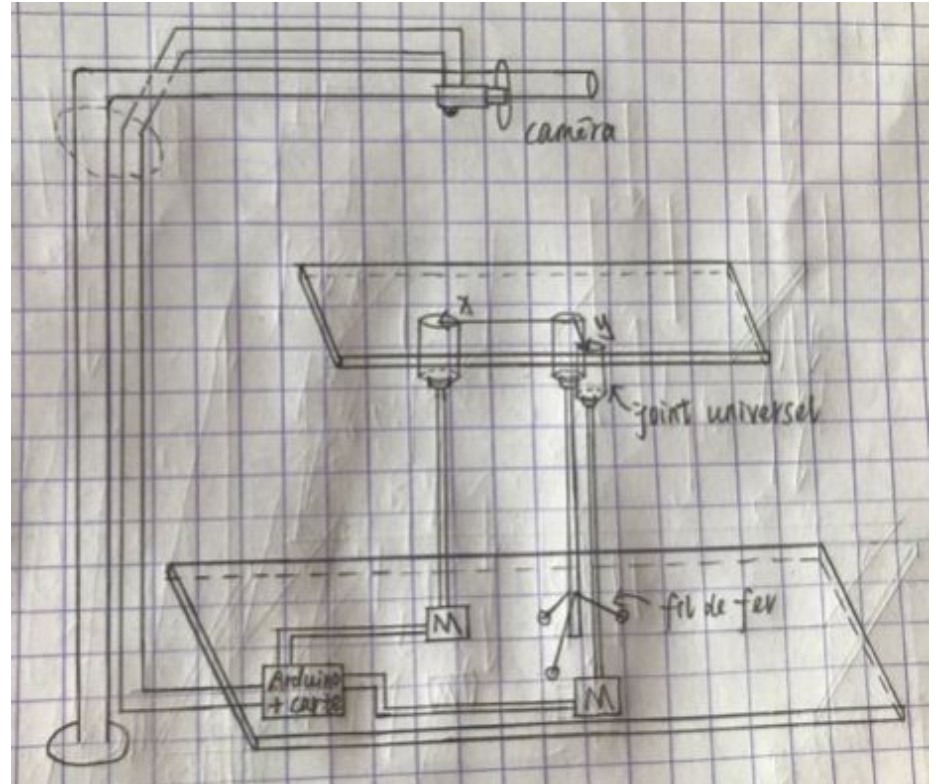


Arduino SPI

```
Starting...
22:42:53.984 -> Detected 1:
22:42:53.984 -> block 0: CC block! sig: 12 (10 decimal) x: 151 y: 115 width: 126 height: 107 angle -69
22:42:58.057 -> Detected 1:
22:42:58.057 -> block 0: CC block! sig: 12 (10 decimal) x: 153 y: 108 width: 141 height: 110 angle -51
22:43:01.516 -> Detected 1:
22:43:01.516 -> block 0: CC block! sig: 12 (10 decimal) x: 145 y: 107 width: 138 height: 110 angle -41
22:43:04.921 -> Detected 1:
22:43:04.921 -> block 0: CC block! sig: 12 (10 decimal) x: 143 y: 110 width: 128 height: 108 angle -54
22:43:08.424 -> Detected 1:
22:43:08.424 -> block 0: CC block! sig: 12 (10 decimal) x: 139 y: 111 width: 135 height: 108 angle -52
22:43:12.403 -> Detected 1:
22:43:12.403 -> block 0: CC block! sig: 12 (10 decimal) x: 138 y: 111 width: 128 height: 109 angle -68
22:43:16.077 -> Detected 1:
22:43:16.077 -> block 0: CC block! sig: 12 (10 decimal) x: 137 y: 112 width: 136 height: 107 angle -70
```

Structure

- Deux planches
- Universal joint
- Deux servomoteurs
- Caméra



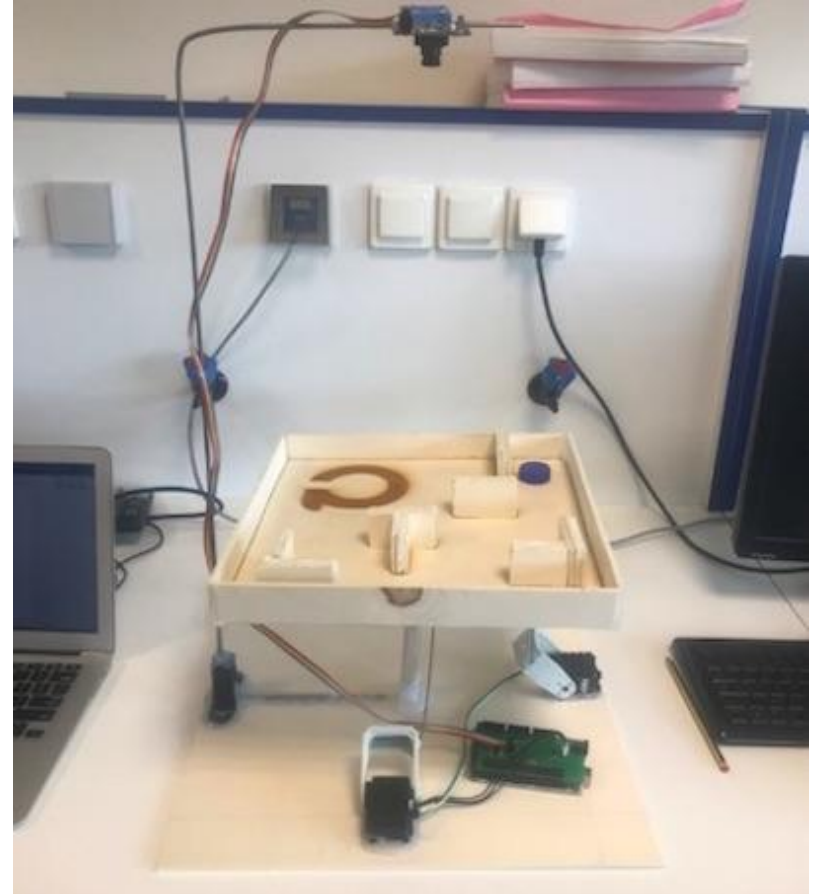
Labyrinthe

- La taille du labyrinthe
30 x 30 cm
- Obstacles
Méthode : gravure + mûrs



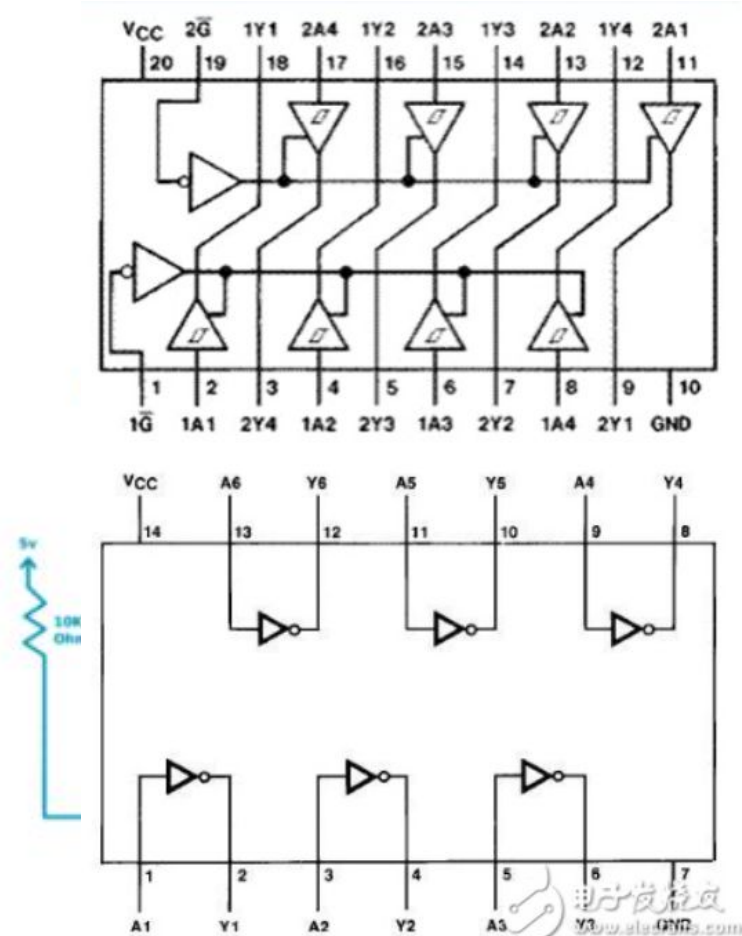
— / —

Réalisation



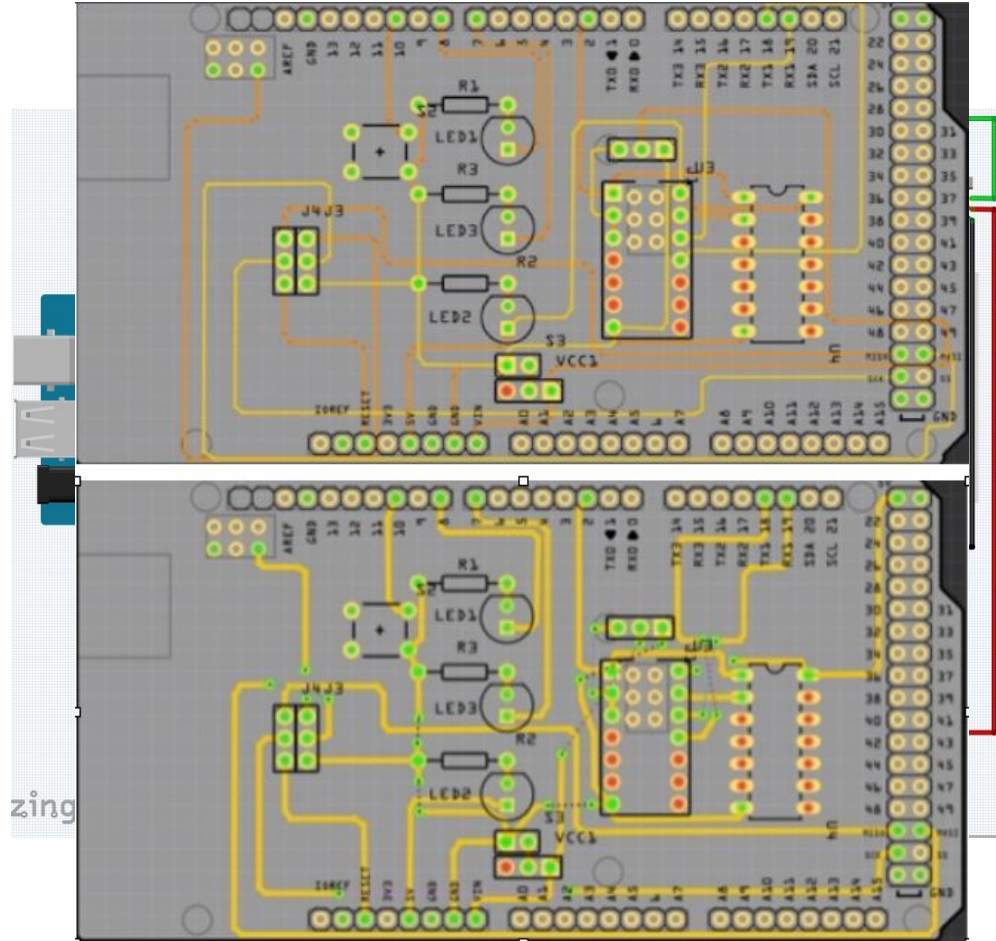
Tester les servomoteurs

- Connexion : Ports RX et TX de l'Arduino
TX -- non inverseuse
RX -- inverseuse
- Buffer :
74LS241N
74LS244N (+ 74HC04N)



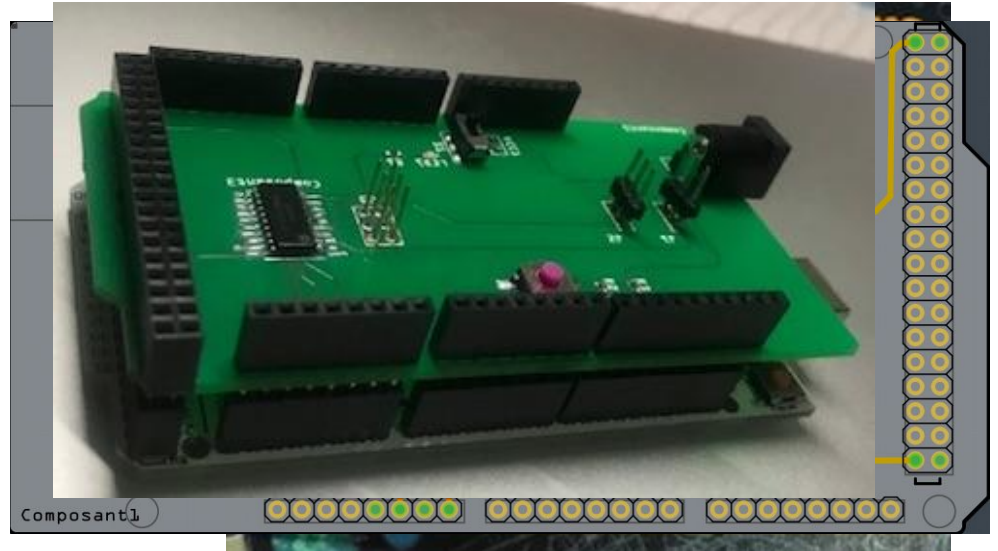
Dessiner la carte

- Interrupteur
Contrôler le système
- Alimentation
Arduino VIN - 9V
- Deux face -- **Pas réalisable**
Seule face : pistes plus larges
cavaliers



Carte électronique

- Méthode 1 (74LS244N)
Le plaque d'essai
- Méthode 2 (74LS241N)
PCB



Contrôleur P (proportion)

- Principe : $U(t) = K_p * Err(t)$
L'écart du système est reflété de manière proportionnelle.
- Le coefficient de l'inclinaison :
Pente faible -- Coefficient grand
Pente haute -- Coefficient petit
- La plage de rotation du moteur :
Limiter la rotation du moteur et protéger la structure.

```
float Pid_controlClass :: PID_realizeX(float Po
pi.SetPositionX=PositionX;
pi.errX=pi.SetPositionX-pi.ActualPositionX;
```

```
Pid_control.pi.ActualPositionX = x;
Pid_control.pi.ActualPositionY = y;
```

```
Serial.println(Pid_control.PID_realizeX(Px));
Serial.println(Pid_control.PID_realizeY(Py));
//Serial.println(Pid_control.pi.voltageX);
Dynamixel.moveSpeed(1,Pid_control.PID_realizeX(Px),400);
Dynamixel.moveSpeed(2,Pid_control.PID_realizeY(Py),400);
```

```
if (pi.angle1<450) pi.angle1=450;
return pi.angle1;
}
```

Démonstration



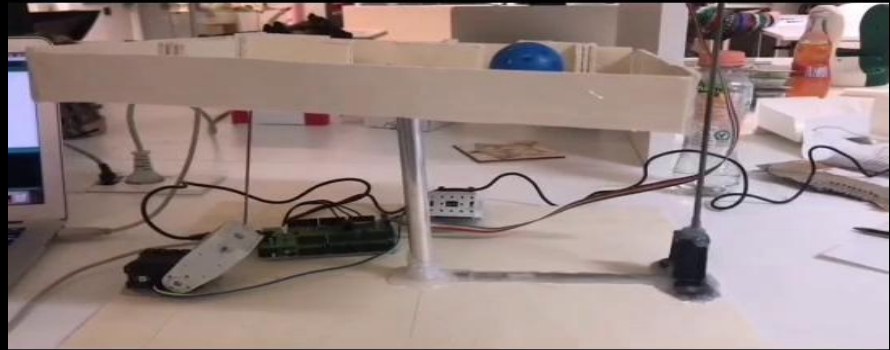
Trajectoire spécifiée

```

void f...
    i...
        if (pi.i==0|pi.i==3){
            pi.angleX=600.7;
        }else
        {
            pi...
                if(pi.voltageX>=0){
                    pi.angleX=644.7-5*pi.voltageX;
                }
                else if(pi.voltageX<0){
                    pi.angleX=644.7-2*pi.voltageX;
                }
            }
        }
    }
    pi.volta...
}

```

Démonstration



Conclusion



Merci pour votre attention !

