جامعة مولاي إسماعيل
UNIVERSITÉ MOULAY ISMAÏL

كلية العلوم
FACULTÉ DES SCIENCES

Master Intelligence Artificielle et Analyse des Données

ROBOTIQUE

COMPTE RENDU De TP 5

Réalisé par :

HALIMA DAOUDI
HAJAR BOUCHAMA
MARYEM ANFEDOUAK
NOHAYLA ANOADA
MERYEM ILLA
ASMAE ANEMROUD

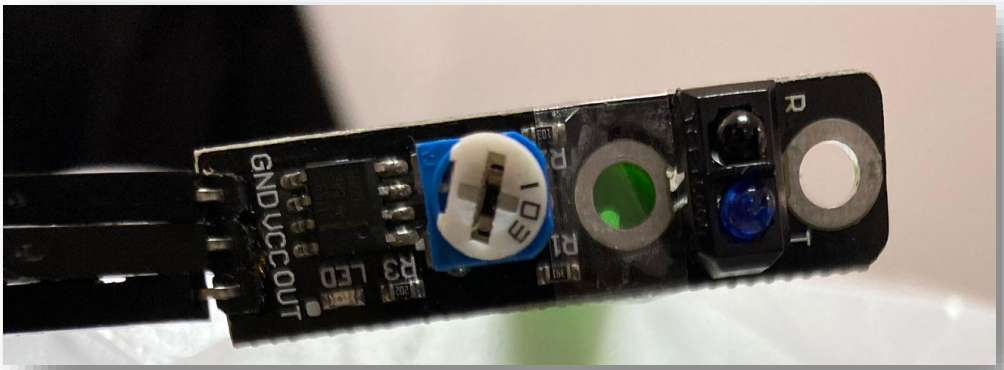
Année universitaire : 2023 – 2024

Introduction :

La manipulation réalisée considère un robot qui peut suivre une ligne noire d'une façon autonome à l'aide d'un capteur. Il est très utilisé dans les différentes applications dans le domaine industriel.

C'est quoi un capteur de suiveur de ligne ?

Un capteur de suivi de ligne est un capteur détecte si une surface réfléchissant ou absorbant la lumière se trouve devant le capteur. Ce qui est actuellement le cas, le module sort sur sa sortie numérique, comme il est montré dans les images ci-dessous. Ce comportement peut être utilisé dans les systèmes de contrôle, comme ils le sont par exemple dans les robots, afin de suivre une ligne de manière autonome.



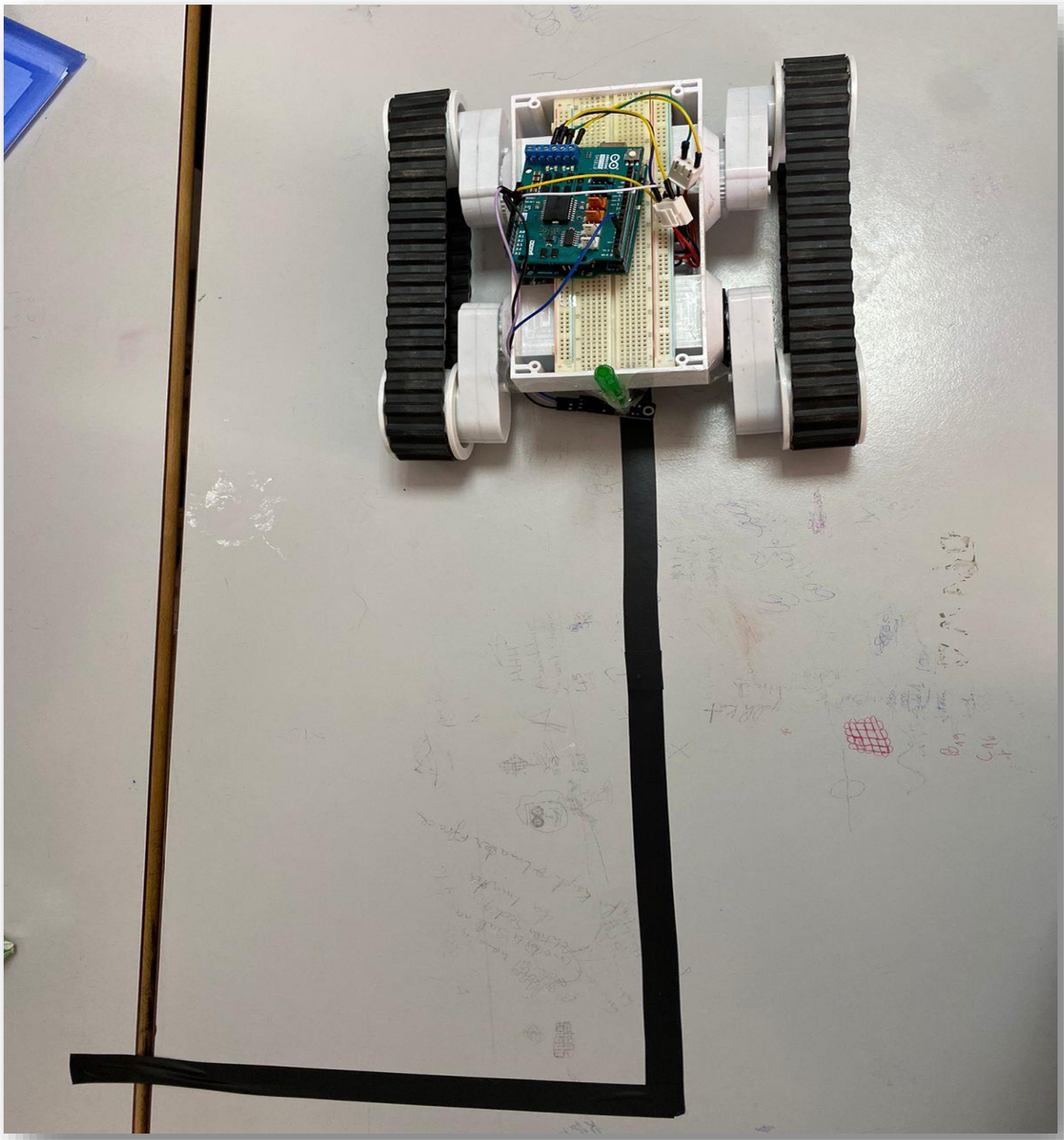
Comment marche un suiveur de ligne ?

Le robot suit une ligne noire sur un arrière-plan blanc tracé au sol qui représente le chemin à suivre, et pour faire cela le robot a besoin d'un capteur suiveur de ligne qui distingue la ligne noir de l'arrière-plan blanc.

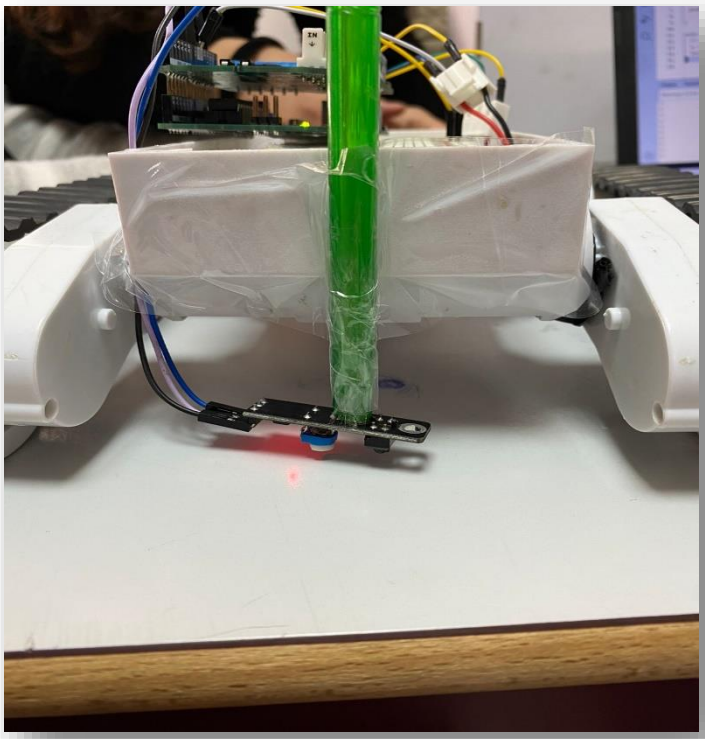
Réalisation de montage :

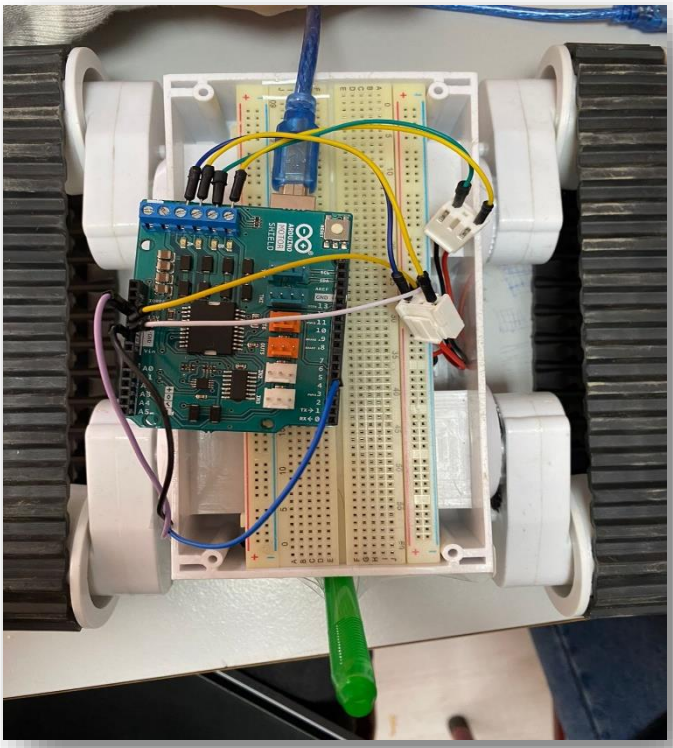
Ce Montage est réalisé par :

- Connecter le port OUT au pin 4 d'Arduino UNO.
- Connecter le port VCC au port 3.3V.
- Connecter le port GND du capteur au port GND d'Arduino UNO.



Code :





Avant d'implémenter le code de manipulation,
il faut tester le capteur s'il détecte la ligne noire suivi par l'envoi des valeurs 0 et 1. Si le code renvoi la valeur 0, le capteur détecte la ligne noire, Sinon le capteur détecte l'arrière-plan.

```
int pincap1=5;
int a;
void setup()
{
  Serial.begin(9600);
  Serial.println(F("Initialize System"));
  pinMode(pincap1, INPUT);
}

void loop()
{
  a = digitalRead(pincap1);
  Serial.println(a);
}
```

```
0
0
0
0
0
0
0
0
1
1
1
1
1
1
1
```

Ensuite, on implémente le code pour que le robot suivre une ligne noire droite c'est-à-dire le capteur détecte la valeur 0. Si ce n'est pas le cas (détection de valeur 1), il doit retourner jusqu'à une nouvelle détection de la valeur 0 et continue tout droite.

Cela est exprimée par le code ci-dessous :

- Nous définissons les constantes pour les broches de direction, vitesse et frein pour les deux moteurs, deux variables (a et pincap1) et la constante (gp2y0a21Pin) qui est associée à la broche analogique A4 de la carte Arduino.

```
int pincap1=5;
int a;
const int gp2y0a21Pin   = A4;
const int directionA = 12;
const int brakeA = 9;
const int speedA = 3;
const int directionB = 13;
const int brakeB = 8;
const int speedB = 11;
```

- **Setup()** : nous initialisons la communication série et en configurant les broches, tout d'about pour les broches (pincap1 et gp2y0a21Pin) en tant qu'entrées, cela signifie qu'elle seront utilisées pour lire les données provenant dans le capteur, et concernant les autres broches sont configurées en tant que sorties, et cela signifie qu'elles seront utilisées pour contrôler les moteurs.

```
void setup()
{
  Serial.begin(9600);
  Serial.println(F("Initialize System"));
  pinMode(pincap1, INPUT);

  pinMode(gp2y0a21Pin, INPUT);
  pinMode(directionA, OUTPUT);
  pinMode(brakeA, OUTPUT);
  pinMode(directionB, OUTPUT);
  pinMode(brakeB, OUTPUT);
}
```

- **moveForward()** : cette fonction permet de configurer les broches et les signaux pour faire avancer les moteurs A et B en définissant la direction sur LOW, désactivant le frein et fixant la vitesse à 255 pour chaque moteur

```
void moveForward() {
  digitalWrite(directionA, LOW);
  digitalWrite(brakeA, LOW);
  analogWrite(speedA, 255);

  digitalWrite(directionB, LOW);
  digitalWrite(brakeB, LOW);
  analogWrite(speedB, 255);
}
```

- **turnRight()** : cette fonction permet de configurer les broches pour faire tourner les moteurs A et B dans des directions opposées, cela entrainera une rotation vers la droite, cela en activant la direction du moteur gauche et en désactivant la direction du moteur droit, tout en relâchant le frein.

```
void turnRight() {
  digitalWrite(directionA, LOW);
  digitalWrite(brakeA, LOW);
  analogWrite(speedA, 255);

  digitalWrite(directionB, HIGH);
  digitalWrite(brakeB, LOW);
  analogWrite(speedB, 255);
}
```

- **stopMotors()** :cette fonction permet d'activer les freins des moteurs A et B, ce qui arrête leur rotation et les immobilise.

```
void stopMotors() {
  digitalWrite(brakeA, HIGH);
  digitalWrite(brakeB, HIGH);
}
```

- **loop()** : dans cette boucle, la valeur de la broche (pincap1) est lue, affichée sur le port série et en fonction de cette valeur, le programme décide d'appeler les fonctions moveForward() et turnRight() pour controler le mouvement du robot.

```
void loop()
{
  a = digitalRead(pincap1);
  Serial.println(a);
  if(a==0)
  {
    moveForward();
  }
  else
  {
    turnRight();
  }
}
```

Tout cela est bien expliqué dans la vidéo prise dans la séance de TP aujourd'hui.