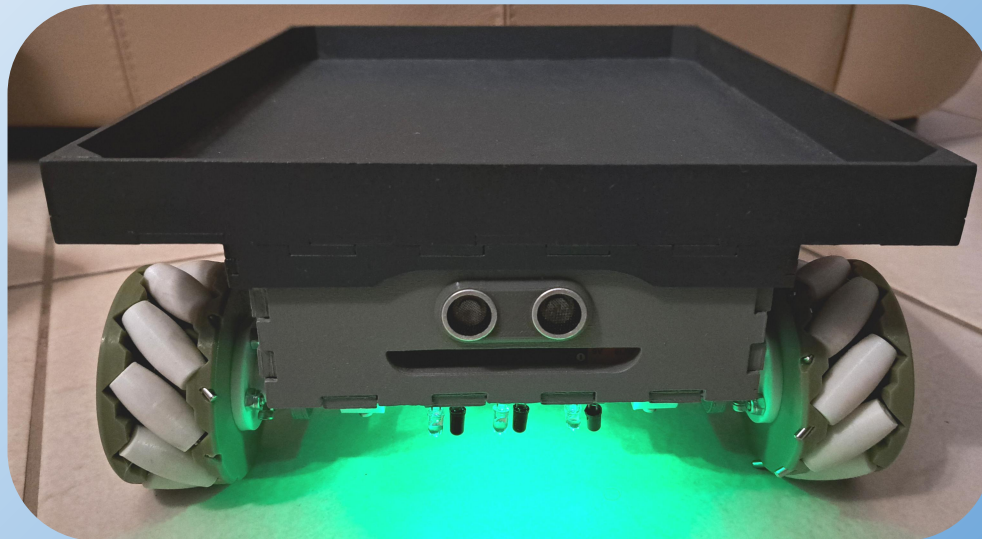


# PROJET ARDUINO

## PEIP 2 :

## POLYMECANUM



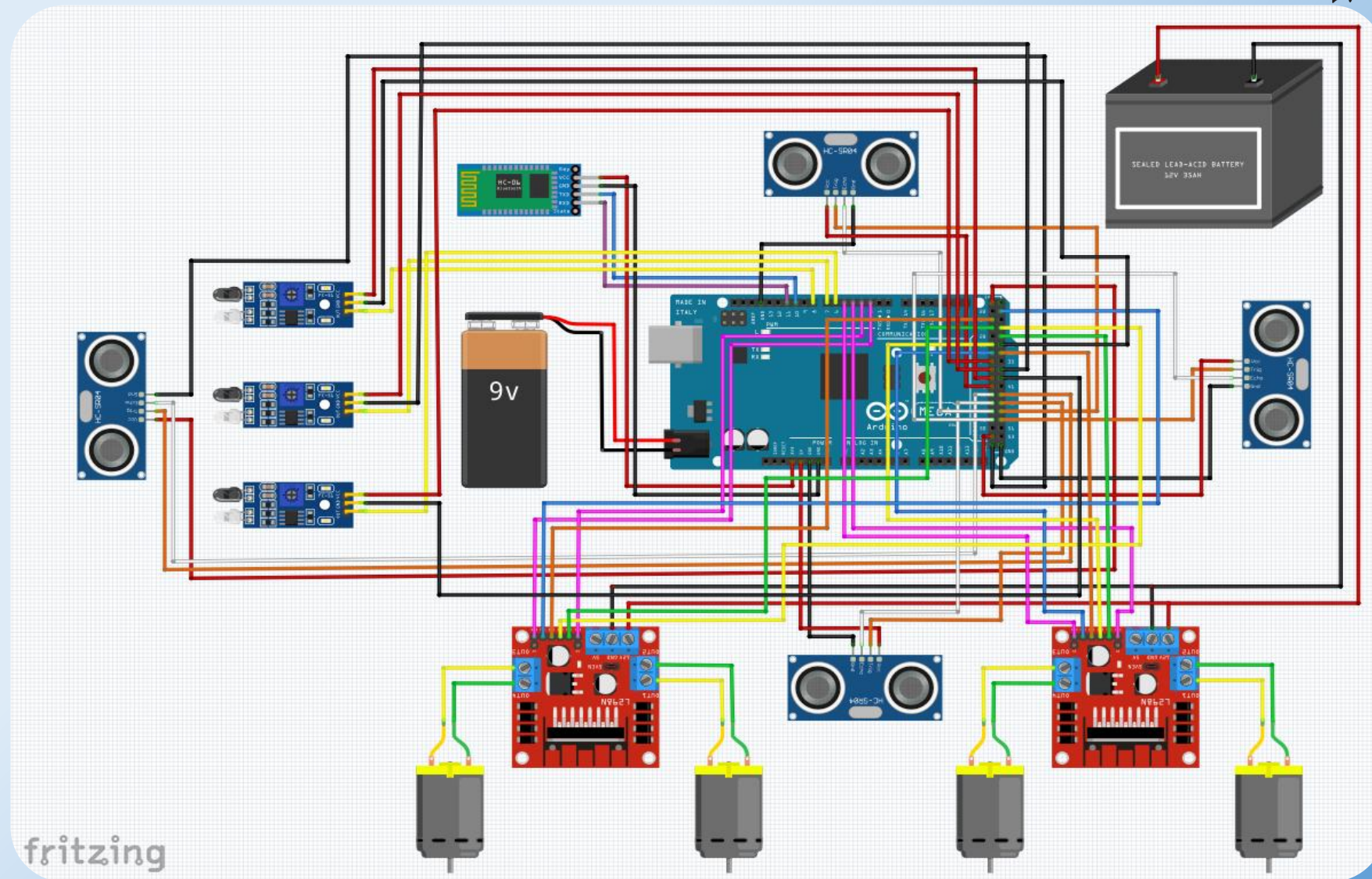
Valentin GIROD-ROUX & Gabriel LIAUTARD



POLYTECH®

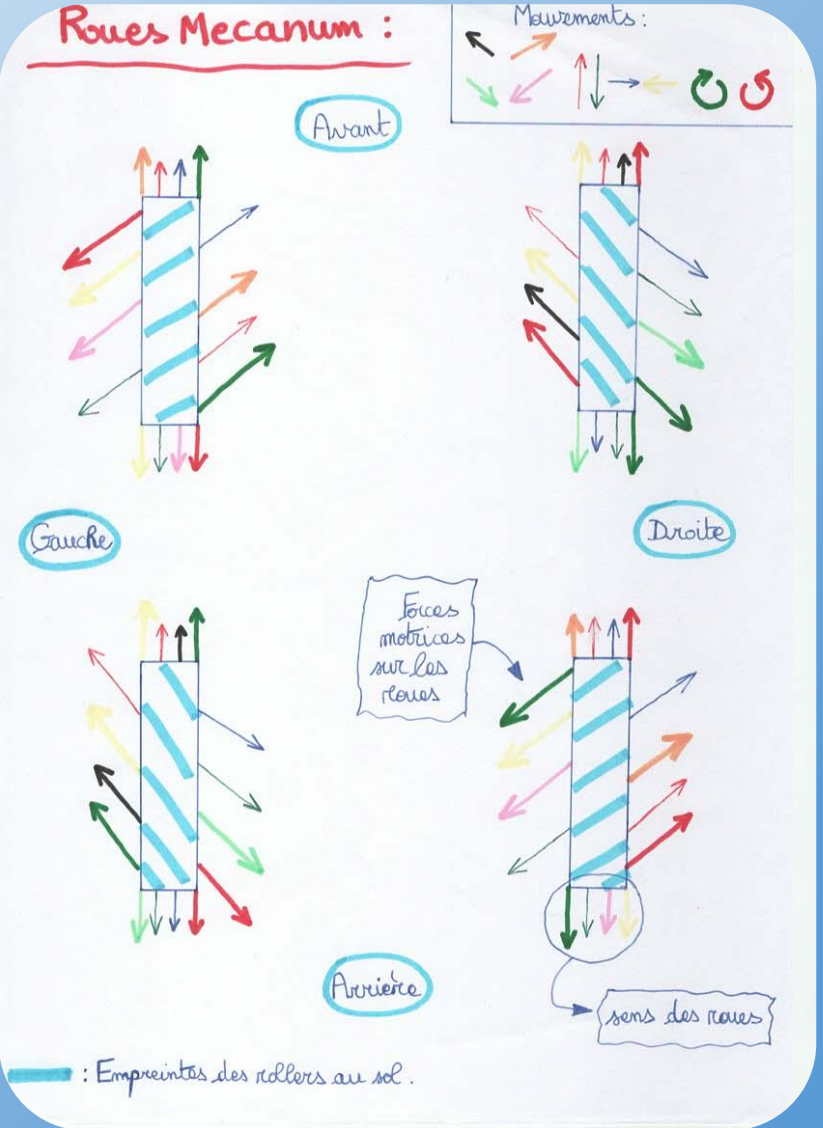
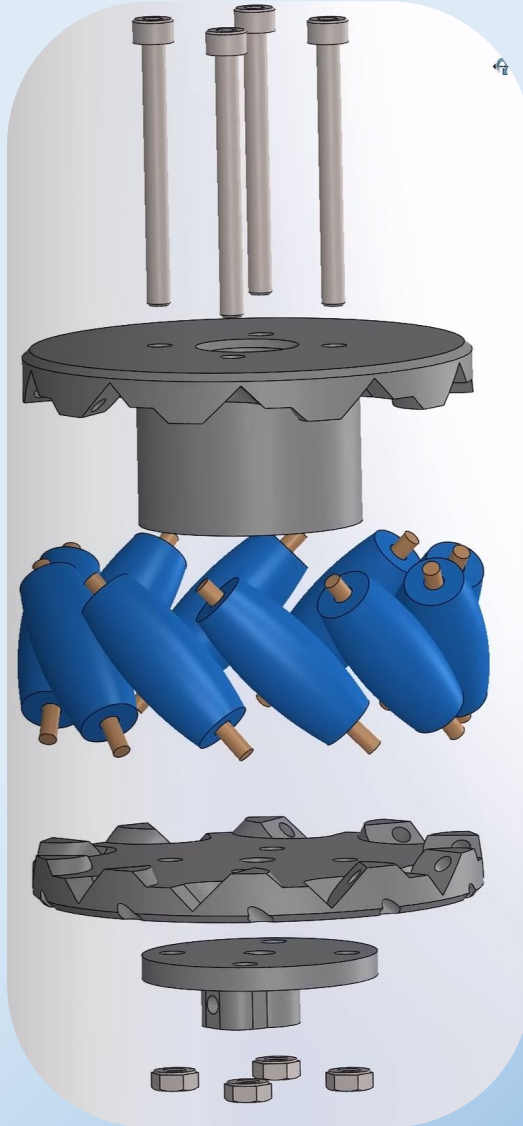


## A cartoon-style illustration of a green and yellow T-Rex dinosaur. The dinosaur is standing on its hind legs, facing left, with its mouth slightly open showing small teeth. It has a yellow underbelly and yellow stripes along its back and tail. The background is a solid light blue.

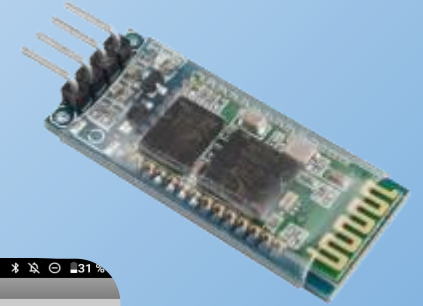
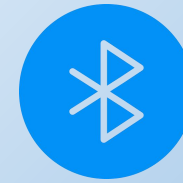




# DÉPLACEMENT OMNIDIRECTIONNEL



# CONTRÔLE À DISTANCE

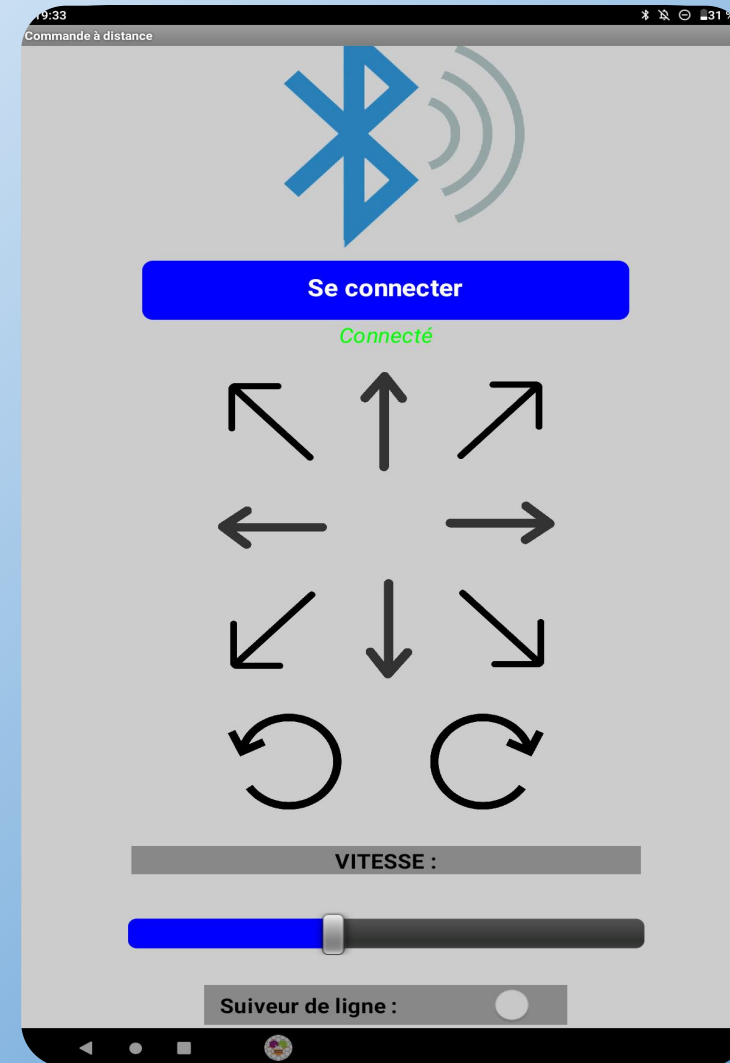


Lorsque le bouton *avancer* est enfoncé, l'application envoie le caractère 'a' au module Bluetooth.



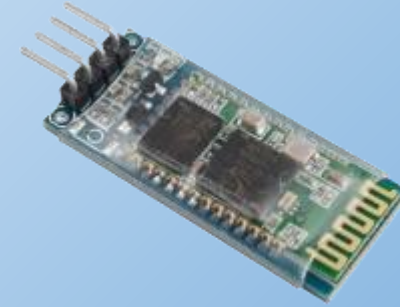
```
else if (message == "a"){
  Serial.println("Avancer");

  // Sens des moteurs
  digitalWrite(INAVG1, LOW);
  digitalWrite(INAVG2, HIGH);
  digitalWrite(INAVD1, HIGH);
  digitalWrite(INAVD2, LOW);
  digitalWrite(INARG1, HIGH);
  digitalWrite(INARG2, LOW);
  digitalWrite(INARD1, LOW);
  digitalWrite(INARD2, HIGH);
}
```





# CONTRÔLE À DISTANCE



Curseur pour ajuster la vitesse de la plate-forme.



initialise global Vitesse à 0

quand Ascenseur1 .Position changée

Position pouce

faire mettre global Vitesse à arrondi obtenir Position pouce

mettre Label2 . Texte à obtenir global Vitesse

appeler BluetoothClient1 .Envoyer texte  
texte " A "

appeler BluetoothClient1 .Envoyer texte  
texte obtenir global Vitesse

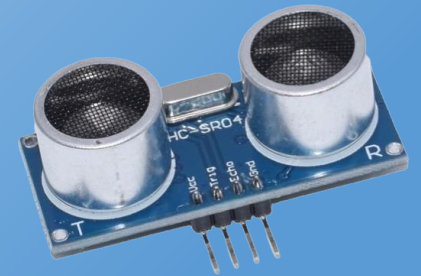
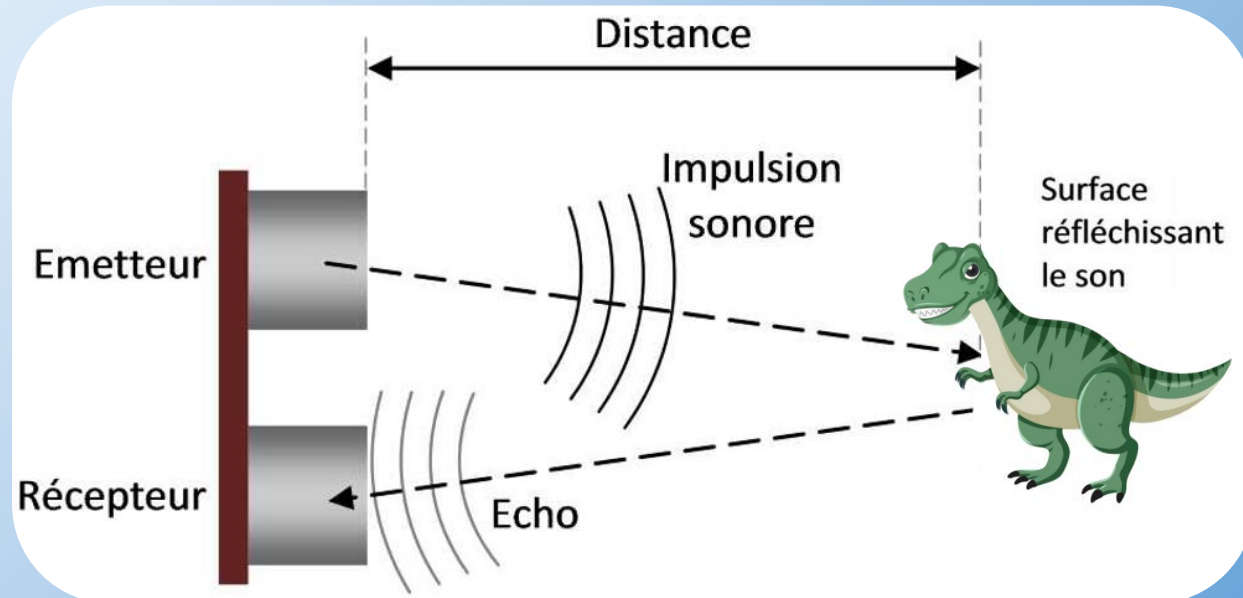
appeler BluetoothClient1 .Envoyer texte  
texte " F "



# DÉTECTION DES OBSTACLES

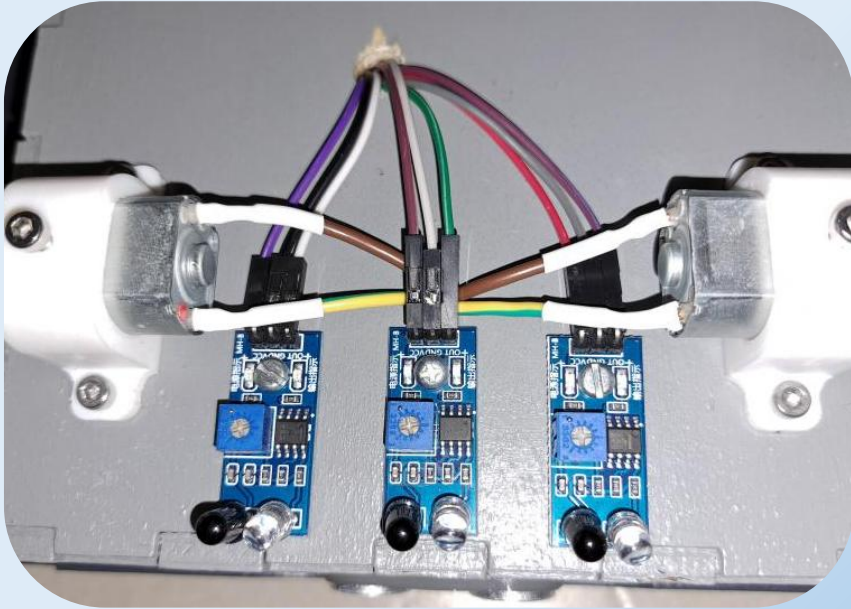
```
// Recupere la distance obstacles
distAV = sonarAV.ping_cm();
distAR = sonarAR.ping_cm();
distG = sonarG.ping_cm();
distD = sonarD.ping_cm();
Serial.println("#####");
Serial.println(distAV);
Serial.println(distAR);
Serial.println(distG);
Serial.println(distD);
```

```
if (distAV <= distMIN and distAR <= distMIN and distG <= distMIN and distD <= distMIN){
  analogWrite(ENAVG,0);
  analogWrite(ENAVD,0);
  analogWrite(ENARG,0);
  analogWrite(ENARD,0);
}
```



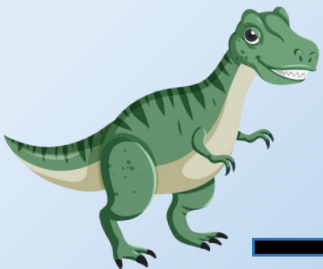


# PARTIE INFRAROUGE (*SUIVEUR DE LIGNE*)



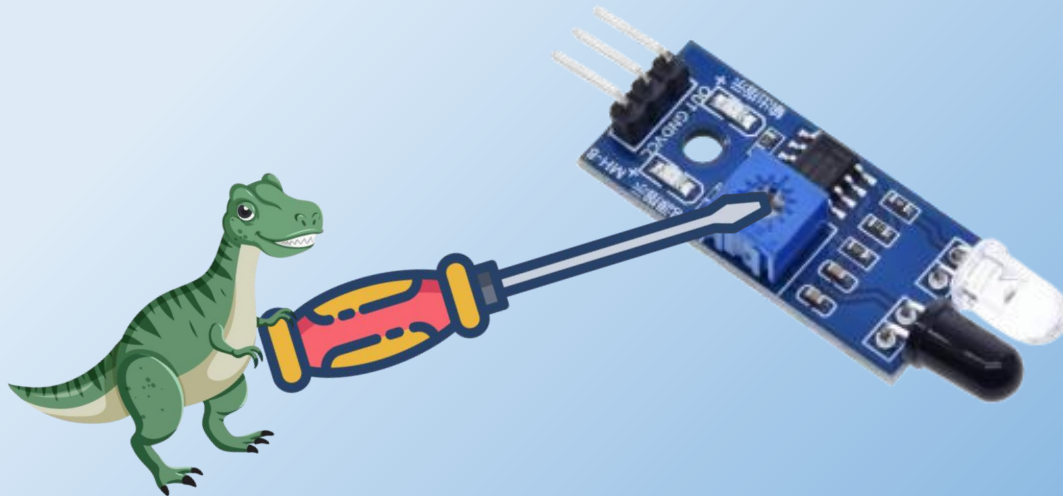
```
// Si le capteur central détecte la bande noire alors avancer droit
else if (valeurIRG == LOW && valeurIRC == HIGH && valeurIRD == LOW){
    Serial.println("NOIR Centre -> Avancer tout droit");
    // Vitesse des moteurs
    analogWrite(ENAVG, vitesse);
    analogWrite(ENAVD, vitesse);
    analogWrite(ENARG, vitesse);
    analogWrite(ENARD, vitesse);
}

// Si le capteur de gauche détecte la bande noire alors tourner légèrement à gauche
else if (valeurIRG == HIGH && valeurIRC == LOW && valeurIRD == LOW){
    memoire = "Gauche";
    Serial.println("NOIR Gauche -> Tourner légèrement à gauche");
    // Vitesse des moteurs
    analogWrite(ENAVG, vitesse * coefVitesse); // Réduire la vitesse du moteur
    analogWrite(ENAVD, vitesse);
    analogWrite(ENARG, vitesse * coefVitesse);
    analogWrite(ENARD, vitesse);
}
```



# PARTIE INFRAROUGE (SUIVEUR DE LIGNE)

```
// Si le capteur de droite détecte la bande noire alors tourner légèrement à droite
else if (valeurIRG == LOW && valeurIRC == LOW && valeurIRD == HIGH){
    memoire = "Droite";
    Serial.println("NOIR Droite -> Tourner légèrement à droite");
    // Vitesse des moteurs
    analogWrite(ENAVG, vitesse);
    analogWrite(ENAVD, vitesse * coefVitesse); // Réduire la vitesse
    analogWrite(ENARG, vitesse);
    analogWrite(ENARD, vitesse * coefVitesse);
}
```



```
else if (memoire == "Gauche"){ // Perdu à droite
    // Vitesse des moteurs
    analogWrite(ENAVG, 0);
    analogWrite(ENAVD, vitesse);
    analogWrite(ENARG, 0);
    analogWrite(ENARD, vitesse);
}

else if (memoire == "Droite"){ // Perdu à gauche
    // Vitesse des moteurs
    analogWrite(ENAVG, vitesse);
    analogWrite(ENAVD, 0);
    analogWrite(ENARG, vitesse);
    analogWrite(ENARD, 0);
}

Serial.println("Suiveur de ligne desactive");
// Vitesse des moteurs STOP
analogWrite(ENAVG, 0);
analogWrite(ENAVD, 0);
analogWrite(ENARG, 0);
analogWrite(ENARD, 0);
```



# CONCLUSION



- Mise en pratique des compétences acquises en prépa dans les différentes matières.
- Système polyvalent : *roues, contrôle à distance, détection d'obstacles* et *suiveur de ligne*.
- Améliorations : *caméra* -> prévoir les virages plus précisément / *bras* avec des *servomoteurs* et une *pince* afin de récupérer les équipements / implémentation de l'*IA* pour éviter les obstacles de manière *autonome*.

MERCI !

