```
#include <Servo.h>


// Variables
int sens;
```

```
// pins Social distance
const int echopin = 19;
const int trigerpin = 18;
int distance;
int distanceDevant;
int distanceDroite;
int distanceGauche;
```

```
// Pins moteurs
int IN1 = 13;
int IN2 = 12;
int ENA = 11;
int IN3 = 8;
int IN4 = 7;
int ENB = 6;
int vitesse = 100;
```

```
Servo myservo;  // create Servo object to control a servo
```

```
// Bluetooth sur RX/TX matériel
char commande;
```

```
int pos = 0;  // variable to store the servo position
```

```
void setup() {
  myservo.attach(3);  // attaches the servo on pin 3 to the Servo object
  myservo.write(90);
```

```
  Serial.begin(19200);
  pinMode(echopin, INPUT);
  pinMode(trigerpin, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
```

```
  // stopper
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}

// Fonction Ajoutées

int sensorsocial() {
  digitalWrite(trigerpin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigerpin, HIGH);
  delay(5);
  digitalWrite(trigerpin, LOW);

  int duree = pulseIn(echopin, HIGH);
  distance = (duree * 0.034) / 2;

  return distance;
}

void ballayage() {

  distanceDevant = sensorsocial();
  Serial.println("_____________________");
  Serial.print(" distance devant : ");
  Serial.print(distanceDevant);
  Serial.println(" cm");
  delay(1000);
  for (pos = 90; pos <= 150; pos += 1) {  // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);  // tell servo to go to position in variable 'pos'
    delay(15);           // waits 15 ms for the servo to reach the position
  }
  distanceGauche = sensorsocial();
  Serial.print(" distance à gauche : ");
  Serial.print(distanceGauche);
  Serial.println(" cm");
  delay(1000);

  for (pos = 150; pos >= 30; pos -= 1) {  // goes from 180 degrees to 0 degrees
    myservo.write(pos);                   // tell servo to go to position in variable 'pos'
```

```
    delay(15);                           // waits 15 ms for the servo to reach the position
  }
  distanceDroite = sensorsocial();
  Serial.print(" distance à droite : ");
  Serial.print(distanceDroite);
  Serial.println(" cm");
  delay(1000);
```

```
  for (pos = 30; pos <= 90; pos += 1) {  // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);  // tell servo to go to position in variable 'pos'
    delay(10);           // waits 15 ms for the servo to reach the position
  }
  delay(1000);
}
```

```
int choixsens() {
```

```
  if (distanceDevant >= distanceDroite && distanceDevant >= distanceGauche) {
    sens = 0;
  } else if (distanceDevant >= distanceDroite && distanceDevant <= distanceGauche) {
    sens = -1;
  } else if (distanceDevant <= distanceDroite && distanceDevant >= distanceGauche) {
    sens = 1;
  }
```

```
  return sens;
}
```

```
// Fonctions moteurs
void avancer(int vitesse) {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  analogWrite(ENA, vitesse);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(ENB, vitesse);
}
```

```
void reculer(int vitesse) {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  analogWrite(ENA, vitesse);
  digitalWrite(IN3, HIGH);
```

```
  digitalWrite(IN4, LOW);
  analogWrite(ENB, vitesse);
}
```

```
void stop() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}
```

```
void gauche(int vitesse) {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  analogWrite(ENA, vitesse);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(ENB, vitesse);
}
```

```
void droite(int vitesse) {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  analogWrite(ENA, vitesse);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(ENB, vitesse);
}
void choixdirection() {
  sens = choixsens();
  Serial.print("Sens choisi : ");
  Serial.println(sens);
  Serial.println("_____");
```

```
  // Si tout est bloqué (distances faibles)
  if (distanceDevant < 20 && distanceDroite < 20 && distanceGauche < 20) {
    Serial.println("  Zone bloquée - manœuvre d'évitement !");
    reculer(vitesse);
    delay(700);
    stop();
```

```
    // Choisir un côté aléatoire pour se dégager
    if (random(0, 2) == 0) {
      gauche(vitesse);
```

```
    Serial.println("  Dégagement gauche");
  } else {
    droite(vitesse);
    Serial.println("  Dégagement droite");
  }
  delay(600);
  stop();
  return;  // Sortir de la fonction pour relancer un balayage
}
```

```
// Si la voie devant est libre
if (distanceDevant >= 25 && sens == 0) {
  avancer(vitesse);
  delay(800);
  stop();
}
```

```
// Si obstacle devant
else if (distanceDevant < 25 && sens == 0) {
  Serial.println("  Obst. devant - recul puis choix latéral");
  reculer(vitesse);
  delay(500);
  stop();
```

```
  // Choisir le meilleur côté libre
  if (distanceDroite > distanceGauche) {
    droite(vitesse);
    delay(500);
  } else {
    gauche(vitesse);
    delay(500);
  }
  stop();
}
```

```
// Si le meilleur sens est à droite
else if (sens == 1) {
  if (distanceDroite > 20) {
    droite(vitesse);
    delay(600);
    stop();
  } else {
    Serial.println("  Droite bloquée, demi-tour");
    reculer(vitesse);
```

```
      delay(500);
      gauche(vitesse);
      delay(700);
      stop();
    }
  }
```

```
  // Si le meilleur sens est à gauche
  else if (sens == -1) {
    if (distanceGauche > 20) {
      gauche(vitesse);
      delay(600);
      stop();
    } else {
      Serial.println("  Gauche bloquée, demi-tour");
      reculer(vitesse);
      delay(500);
      droite(vitesse);
      delay(700);
      stop();
    }
  }
}
```

```
void modeManuelLoop() {
  switch (commande) {
    case 'F':
      avancer(vitesse);
      break;
    case 'B':
      reculer(vitesse);
      break;
    case 'L':
      gauche(vitesse);
      break;
    case 'R':
      droite(vitesse);
      break;
    case 'S':
      stop();
      break;
    default:
      stop();
      break;
```

```
  }
}
```

```
void loop() {
```

```
  ballayage();
  choixdirection();
```

```
  // Lecture Bluetooth
  //if (Serial.available()) {
  //commande = Serial.read();
  //   Serial.println("Je suis Xion2");
  //}
  // modeManuelLoop();
}
```