

iot

1. Détection de la ligne :

Les capteurs infrarouges détectent une ligne noire sur un fond blanc en utilisant des différences de réflexion lumineuse.

Code associé :

```
#define LEFT_SENSOR 2
#define MIDDLE_SENSOR 3
#define RIGHT_SENSOR 4

if (middleValue == 1) {
    moveForward();
} else if (leftValue == 1 && rightValue == 0) {
    turnLeft();
} else if (rightValue == 1 && leftValue == 0) {
    turnRight();
}
```

2. Détection des obstacles :

Le capteur à ultrasons détecte les obstacles à proximité et les évite.

Code associé :

```
#define echo 6
#define trig 5

float distance = getUltrasonicDistance();
if (distance <= 20) { // Obstacle detected
    avoidObstacle();
}
```

```
// Fonction pour mesurer la distance avec le capteur à ultras
ons
float getUltrasonicDistance() {
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    float duration = pulseIn(echo, HIGH);
    return duration * 0.034 / 2; // Convertir en distance en
cm
}
```

3. Traitement des données :

La carte Arduino traite les données des capteurs (infrarouges et ultrasons) pour localiser la ligne ou détecter un obstacle.

Code associé :

```
Lecture des capteurs et traitement des données
float distance = getUltrasonicDistance();
Serial.print("Distance: ");
Serial.println(distance);
if (distance <= 20) {
    avoidObstacle();
} else {
    if (middleValue == 1) {
        moveForward();
    } else if (leftValue == 1 && rightValue == 0) {
        turnLeft();
    } else if (rightValue == 1 && leftValue == 0) {
        turnRight();
    }
}
```

4. Contrôle des moteurs :

L'Arduino ajuste la direction et la vitesse des moteurs en fonction de la position de la ligne détectée.

Code associé :

```
#define ENA 10 // Speed control for Motor A#define ENB 11 //  
Speed control for Motor B#define IN1 8 // Direction control  
for Motor A#define IN2 9  
#define IN3 12 // Direction control for Motor B#define IN4 13  
  
pinMode(ENA, OUTPUT);  
pinMode(ENB, OUTPUT);  
pinMode(IN1, OUTPUT);  
pinMode(IN2, OUTPUT);  
pinMode(IN3, OUTPUT);  
pinMode(IN4, OUTPUT);
```

5. Navigation autonome :

Le robot suit la ligne et contourne les obstacles en temps réel grâce aux fonctions de mouvement.

Code associé :

```
if (middleValue == 1) {  
    moveForward();  
} else if (leftValue == 1 && rightValue == 0) {  
    turnLeft();  
} else if (rightValue == 1 && leftValue == 0) {  
    turnRight();  
}  
  
delay(50); // Pause pour la stabilité
```

Résumé :

- **Détection de la ligne** : Capteurs infrarouges (`LEFT_SENSOR` , `MIDDLE_SENSOR` , `RIGHT_SENSOR`).
- **Détection des obstacles** : Capteur ultrasons (`echo` , `trig` , `getUltrasonicDistance()`).
- **Traitement des données** : Analyse de la ligne ou des obstacles (`if-else` conditions).
- **Contrôle des moteurs** : Pilotes moteur (`ENA` , `ENB` , `IN1` , `IN2` , `IN3` , `IN4`).
- **Navigation autonome** : Fonctionnement continu dans la boucle `loop()` .