

```

#include <Servo.h>           // Biblioteca para controlar o servo
#include <NewPing.h>         // Biblioteca para controlar o sensor ultrassônico
#include <AFMotor.h>         // Biblioteca para controlar os motores

NewPing Dist(A2, A1, 200); // Inicializa o sensor ultrassônico nos pinos A2 (trigger) e A1
                             // (echo), com alcance máximo de 200 cm

Servo myServo;              // Criação de um objeto do tipo Servo
int distancia = 0;          // Variável para armazenar a distância frontal
int distanciaD = 0;         // Variável para armazenar a distância à direita
int distanciaE = 0;         // Variável para armazenar a distância à esquerda
boolean Booleano = false;   // Variável para verificar o estado do movimento (true = em
                             // movimento)

AF_DCMotor Motor1(1, MOTOR12_1KHZ); // Motor 1 ligado ao canal 1 com frequência de 1
kHz
AF_DCMotor Motor2(2, MOTOR12_1KHZ); // Motor 2 ligado ao canal 2
AF_DCMotor Motor3(3, MOTOR34_1KHZ); // Motor 3 ligado ao canal 3
AF_DCMotor Motor4(4, MOTOR34_1KHZ); // Motor 4 ligado ao canal 4

int velocidadeInicial = 100; // Velocidade inicial dos motores
int velocidadeMaxima = 100; // Velocidade máxima dos motores em movimento à frente
int velocidadeCurva = 255; // Velocidade para curvas (aumentada para fazer curvas mais
rápidas)

void setup() {
  myServo.attach(10);      // Define o pino 10 para o controle do servo
  myServo.write(115);      // Posiciona o servo no centro (115 graus)
  delay(1000);             // Aguarda 1 segundo para estabilização
}

void loop() {
  moverServoOscilante();    // Função para fazer o servo oscilar constantemente

  distancia = medirDistancia(); // Mede a distância frontal

  if (distancia >= 25 && distancia <= 250) { // Se a distância frontal está entre 25 cm e 250 cm
    Adiante();                // Move para frente
  } else if (distancia < 25) { // Se a distância é menor que 25 cm
    Frear();                  // Para o carrinho
    delay(200);               // Pequena pausa
    darRe();                  // Dá ré ao detectar um obstáculo
    Frear();                  // Para após dar ré
    delay(200);               // Pausa após dar ré
  }
}

```

```

distanciaD = olharDireita();          // Mede a distância à direita
distanciaE = olharEsquerda();        // Mede a distância à esquerda

// Decide para onde virar com base nas distâncias
if (distanciaD >= distanciaE && distanciaD >= 25) {
    VirarEsquerda();                  // Vira para a esquerda se a direita tiver maior distância
} else if (distanciaE >= distanciaD && distanciaE >= 25) {
    VirarDireita();                   // Vira para a direita se a esquerda tiver maior distância
} else {
    Frear();                          // Para o carrinho
    delay(100);                       // Pausa antes da curva
    CurvaDireitaRapida();              // Realiza uma curva larga à direita
}
}
}

void moverServoOscilante() {
    static int anguloAtual = 115;      // Ângulo inicial do servo no centro
    static bool movendoDireita = true; // Variável para indicar a direção do movimento do servo

    if (movendoDireita) {              // Se movendo para a direita
        anguloAtual += 10;             // Aumenta o ângulo
        if (anguloAtual >= 165) movendoDireita = false; // Inverte a direção ao chegar a 165 graus
    } else {
        anguloAtual -= 10;             // Reduz o ângulo se movendo para a esquerda
        if (anguloAtual <= 60) movendoDireita = true;  // Inverte a direção ao chegar a 60 graus
    }

    myServo.write(anguloAtual);        // Posiciona o servo no ângulo atual
    delay(50);                         // Pausa de 50 ms entre movimentos do servo
}

int olharDireita() {
    myServo.write(60);                 // Move o servo para olhar à direita
    delay(600);                       // Pausa para estabilização
    int distancia = medirDistancia();  // Mede a distância à direita
    return distancia;                  // Retorna a distância medida
}

int olharEsquerda() {
    myServo.write(165);               // Move o servo para olhar à esquerda
    delay(600);                       // Pausa para estabilização
}

```

```

int distancia = medirDistancia();          // Mede a distância à esquerda
delay(50);                                // Pausa extra para estabilização
myServo.write(115);                        // Retorna o servo ao centro
return distancia;                          // Retorna a distância medida
}

int medirDistancia() {
    delay(10);                             // Pequena pausa antes de medir
    int distanciaCM = Dist.ping_cm();       // Mede a distância usando o sensor ultrassônico
    if (distanciaCM <= 0 || distanciaCM >= 250) {
        distanciaCM = 250;                 // Limita a distância máxima a 250 cm
    }
    return distanciaCM;                    // Retorna a distância em cm
}

void Frear() {
    Motor1.run(RELEASE);                   // Para o motor 1
    Motor2.run(RELEASE);                   // Para o motor 2
    Motor3.run(RELEASE);                   // Para o motor 3
    Motor4.run(RELEASE);                   // Para o motor 4
}

void Adiante() {
    if (!Booleano) {                       // Verifica se o carrinho não está em movimento
        Booleano = true;                   // Define como em movimento
        Motor1.run(FORWARD);               // Configura o motor 1 para frente
        Motor2.run(FORWARD);               // Configura o motor 2 para frente
        Motor3.run(FORWARD);               // Configura o motor 3 para frente
        Motor4.run(FORWARD);               // Configura o motor 4 para frente
        controleVelocidade();               // Aumenta a velocidade gradualmente
    }
}

void darRe() {
    Booleano = false;                       // Define o estado como parado
    Motor1.run(BACKWARD);                   // Configura o motor 1 para trás
    Motor2.run(BACKWARD);                   // Configura o motor 2 para trás
    Motor3.run(BACKWARD);                   // Configura o motor 3 para trás
    Motor4.run(BACKWARD);                   // Configura o motor 4 para trás
    for (int velocidade = velocidadeInicial; velocidade <= 120; velocidade += 10) {
        Motor1.setSpeed(velocidade);       // Aumenta a velocidade do motor 1
        Motor2.setSpeed(velocidade);       // Aumenta a velocidade do motor 2
        Motor3.setSpeed(velocidade);       // Aumenta a velocidade do motor 3
        Motor4.setSpeed(velocidade);       // Aumenta a velocidade do motor 4
    }
}

```

```

    delay(3);
}
delay(0);           // Tempo de ré ajustável
}

void VirarDireita() {
    Motor1.run(FORWARD);           // Motor 1 para frente
    Motor2.run(FORWARD);           // Motor 2 para frente
    Motor3.run(BACKWARD);          // Motor 3 para trás
    Motor4.run(BACKWARD);          // Motor 4 para trás
    Motor1.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 1
    Motor2.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 2
    Motor3.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 3
    Motor4.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 4
    delay(600);                     // Tempo da curva
}

void VirarEsquerda() {
    Motor1.run(BACKWARD);           // Motor 1 para trás
    Motor2.run(BACKWARD);           // Motor 2 para trás
    Motor3.run(FORWARD);            // Motor 3 para frente
    Motor4.run(FORWARD);            // Motor 4 para frente
    Motor1.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 1
    Motor2.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 2
    Motor3.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 3
    Motor4.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 4
    delay(600);                     // Tempo da curva
}

void CurvaDireitaRapida() {
    Motor1.run(FORWARD);           // Motor 1 para frente
    Motor2.run(FORWARD);           // Motor 2 para frente
    Motor3.run(BACKWARD);          // Motor 3 para trás
    Motor4.run(BACKWARD);          // Motor 4 para trás
    Motor1.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 1
    Motor2.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 2
    Motor3.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 3
    Motor4.setSpeed(velocidadeCurva); // Ajusta a velocidade do motor 4
    delay(700);                     // Tempo da curva
    Frear();                         // Para o carrinho após a curva
}

void controleVelocidade() {
    for (int velocidade = velocidadeInicial; velocidade <= velocidadeMaxima; velocidade += 5) {

```

```
Motor1.setSpeed(velocidade);      // Aumenta gradualmente a velocidade do motor 1
Motor2.setSpeed(velocidade);      // Aumenta gradualmente a velocidade do motor 2
Motor3.setSpeed(velocidade);      // Aumenta gradualmente a velocidade do motor 3
Motor4.setSpeed(velocidade);      // Aumenta gradualmente a velocidade do motor 4
delay(2);                          // Pausa entre os incrementos de velocidade
}
```