#include <Servo.h>

// Definición de pines para el motor DC

const int pinIN1 = 11; // Control de dirección 1

const int pinIN2 = 12; // Control de dirección 2

const int pinENA = 10; // Control PWM

// Definición de pines para el sensor ultrasónico

const int trigPin = 7;

const int echoPin = 6;

// Definición de pines para el sensor de color

const int s0 = 2;

const int s1 = 3;

const int s2 = 4;

const int s3 = 5;

const int outPin = 8;

const int ledPin = 13;

// Crear un objeto Servo

Servo miServo;

// Definir el pin donde está conectado el servomotor

const int pinServo = 9;

// Variable para alternar esquivar

bool esquivarIzquierda = true;

void setup() {

// Configurar los pines del motor como salida

pinMode(pinIN1, OUTPUT);

pinMode(pinIN2, OUTPUT);

pinMode(pinENA, OUTPUT);

// Configurar los pines del sensor ultrasónico

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

// Configurar los pines del sensor de color

pinMode(s0, OUTPUT);

pinMode(s1, OUTPUT);

pinMode(s2, OUTPUT);

pinMode(s3, OUTPUT);

pinMode(outPin, INPUT);

pinMode(ledPin, OUTPUT);

// Inicializar el sensor de color

digitalWrite(s0, HIGH);

digitalWrite(s1, LOW);

digitalWrite(ledPin, HIGH); // Encender los LEDs del sensor de color

// Asociar el objeto Servo al pin

miServo.attach(pinServo);

// Iniciar la comunicación serial

Serial.begin(9600);

}

// Funciones para controlar el servomotor

void moverIzquierda() {

miServo.write(60);

delay(1000); // Esperar 1 segundo

}

void moverCentro() {

miServo.write(90);

delay(1000); // Esperar 1 segundo

}

void moverDerecha() {

miServo.write(120);

delay(1000); // Esperar 1 segundo

}

// Función para medir la distancia con el sensor ultrasónico

long medirDistancia() {

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

long duration = pulseIn(echoPin, HIGH);

long distance = duration \* 0.034 / 2; // Convertir a distancia en cm

return distance;

}

// Función para medir el color con el sensor de color

String detectarColor() {

digitalWrite(s2, LOW);

digitalWrite(s3, LOW);

int rojo = pulseIn(outPin, LOW);

digitalWrite(s2, HIGH);

digitalWrite(s3, HIGH);

int verde = pulseIn(outPin, LOW);

digitalWrite(s2, LOW);

digitalWrite(s3, HIGH);

int azul = pulseIn(outPin, LOW);

if (rojo < verde && rojo < azul) {

return "ROJO";

} else if (verde < rojo && verde < azul) {

return "VERDE";

} else {

return "AZUL";

}

}

// Función para controlar el motor DC

void avanzar() {

digitalWrite(pinIN1, LOW);

digitalWrite(pinIN2, HIGH);

analogWrite(pinENA, 255); // Velocidad máxima

}

void detenerMotor() {

analogWrite(pinENA, 0); // Detener el motor

}

void loop() {

long distancia = medirDistancia();

Serial.print("Distancia: ");

Serial.print(distancia);

Serial.println(" cm");

String color = detectarColor();

Serial.print("Color detectado: ");

Serial.println(color);

if (distancia > 10) { // Avanzar si no hay obstáculo cercano

avanzar();

} else { // Detener el motor y esquivar el obstáculo si está cerca

detenerMotor();

delay(500); // Esperar 500 milisegundos para estabilizar el motor

if (color == "VERDE") {

moverIzquierda();

} else if (color == "ROJO") {

moverDerecha();

} else {

// Alternar entre esquivar a la izquierda y a la derecha

if (esquivarIzquierda) {

moverIzquierda();

} else {

moverDerecha();

}

esquivarIzquierda = !esquivarIzquierda; // Cambiar la dirección de esquivar para la próxima vez

}

delay(1000); // Esperar un segundo para girar

avanzar(); // Avanzar un poco después de girar

delay(1000); // Esperar un segundo para avanzar

moverCentro(); // Regresar el servo al centro

}

delay(100); // Esperar 100 ms antes de la siguiente medición

}