#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;

// Crear un objeto Servo

Servo miServo;

// Definir el pin donde está conectado el servomotor

const int pinServo = 9;

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);

// 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 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");

if (distancia > 60) { // Avanzar si no hay obstáculo cercano

avanzar();

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

detenerMotor();

delay(3); // Esperar 3 milisegundos

// Esquivar a la izquierda

moverIzquierda();

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

}