#include <Servo.h> // Include Servo library for controlling the seed dropper

// Motor driver pin definitions

#define ENA 9 // Enable pin for motor A (left side)

#define IN1 8 // Motor A direction pin 1

#define IN2 7 // Motor A direction pin 2

#define IN3 6 // Motor B direction pin 1

#define IN4 5 // Motor B direction pin 2

#define ENB 10 // Enable pin for motor B (right side)

// Seed dropper and sensor pin definitions

#define SERVO\_PIN 3 // Pin connected to the servo motor

#define IR\_SENSOR\_PIN 4 // Pin connected to the IR sensor for seed detection

Servo seedServo; // Create a Servo object

char command; // Variable to store received command

bool isMoving = false; // Track if the car is currently moving

unsigned long previousDropTime = 0; // Store the last time a seed was dropped

const int dropInterval = 1000; // Time interval (1 sec) between seed drops

const int motorSpeed = 1; // Adjusted for ~2km/h speed

void setup() {

Serial.begin(9600); // Start serial communication for debugging

Serial.println("RC Car Seed Dropper Initialized!");

seedServo.attach(SERVO\_PIN); // Attach the servo to its pin

seedServo.write(0); // Set servo to default position

Serial.println("Seed servo initialized and set to 0°.");

// Set motor control pins as outputs

pinMode(ENA, OUTPUT);

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

pinMode(ENB, OUTPUT);

pinMode(IR\_SENSOR\_PIN, INPUT); // Set IR sensor as input

stopMotors(); // Ensure motors are stopped initially

}

void loop() {

// Check if a command is received from serial

if (Serial.available()) {

command = Serial.read();

Serial.print("Received command: ");

Serial.println(command);

// Execute the corresponding function based on command

if (command == 'F') { moveForward(); isMoving = true; }

else if (command == 'B') { moveBackward(); isMoving = true; }

else if (command == 'L') { turnLeft(); isMoving = true; }

else if (command == 'R') { turnRight(); isMoving = true; }

else if (command == 'S') { stopMotors(); isMoving = false; }

}

// If the car is moving, check the seed drop mechanism

if (isMoving) {

unsigned long currentMillis = millis(); // Get current time

if (currentMillis - previousDropTime >= dropInterval) {

previousDropTime = currentMillis; // Update last drop time

Serial.println("Checking seed sensor before dropping...");

if (digitalRead(IR\_SENSOR\_PIN) == LOW) { // If a seed is detected

Serial.println("Seed detected, dropping seed...");

dropSeed();

} else {

Serial.println("No seed detected, skipping drop.");

}

}

}

}

// 🏎️ \*\*Motor Control Functions\*\*

void moveForward() {

Serial.println("Moving Forward");

analogWrite(ENA, motorSpeed); // Set motor A speed

analogWrite(ENB, motorSpeed); // Set motor B speed

digitalWrite(IN1, HIGH); // Move motor A forward

digitalWrite(IN2, LOW);

digitalWrite(IN3, HIGH); // Move motor B forward

digitalWrite(IN4, LOW);

}

void moveBackward() {

Serial.println("Moving Backward");

analogWrite(ENA, motorSpeed);

analogWrite(ENB, motorSpeed);

digitalWrite(IN1, LOW); // Move motor A backward

digitalWrite(IN2, HIGH);

digitalWrite(IN3, LOW); // Move motor B backward

digitalWrite(IN4, HIGH);

}

void turnLeft() {

Serial.println("Turning Left");

analogWrite(ENA, motorSpeed);

analogWrite(ENB, motorSpeed);

digitalWrite(IN1, LOW); // Motor A moves backward

digitalWrite(IN2, HIGH);

digitalWrite(IN3, HIGH); // Motor B moves forward

digitalWrite(IN4, LOW);

}

void turnRight() {

Serial.println("Turning Right");

analogWrite(ENA, motorSpeed);

analogWrite(ENB, motorSpeed);

digitalWrite(IN1, HIGH); // Motor A moves forward

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW); // Motor B moves backward

digitalWrite(IN4, HIGH);

}

void stopMotors() {

Serial.println("Stopping Motors"); // Message for debugging

digitalWrite(IN1, LOW); // Stop motor A

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW); // Stop motor B

digitalWrite(IN4, LOW);

}

// 🌱 \*\*Seed Drop Mechanism\*\*

void dropSeed() {

Serial.println("Dropping Seed...");

// Move servo from 0° to 90° smoothly

for (int angle = 0; angle <= 90; angle += 10) {

seedServo.write(angle);

delay(20);

}

Serial.println("Seed dropper servo moved to 90°.");

delay(200); // Hold position for 0.2s

// Move servo back from 90° to 0° smoothly

for (int angle = 90; angle >= 0; angle -= 10) {

seedServo.write(angle);

delay(20);

}

Serial.println("Seed dropper servo returned to 0°.");

Serial.println("Seed Dropped!");

}