//Entrega 2

#include "HX711.h"

#include <Servo.h>

// Definición de variables

int trigPin = 5;

int echoPin = 3;

int ledVerde = 2;     // LED Verde

int ledAmarillo = 11;  // LED Amarillo

int ledRojo = 4;      // LED Rojo

int ledAzul = 12;      // LED Azul

int touch = 10;

Servo servoPuerta;

bool activado = false; // Variable para controlar si el touch ha sido activado

#define DOUT 8 //cambiar 12

#define CLK 7  //cambiar 13

HX711 scale;

// Configuración de los pines

void setup() {

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  pinMode(ledVerde, OUTPUT);

  pinMode(ledAmarillo, OUTPUT);

  pinMode(ledRojo, OUTPUT);

  pinMode(ledAzul, OUTPUT);

  servoPuerta.attach(9);

  digitalWrite(trigPin, LOW);

  pinMode(touch, INPUT);

  scale.begin(DOUT, CLK);

  scale.set\_scale(100.f);

  scale.tare();

  Serial.println("Iniciando prueba de celda de carga de 20kg");

  // Inicialización del puerto serial

  Serial.begin(9600);

}

void activarServo() {

  // Gira el servo a 500° (proporcional)

  servoPuerta.write(0); // Máxima rotación hacia adelante

  delay(200); // Espera 500 ms mientras se mueve

  // Detener el servo (parar)

  servoPuerta.write(90); // Posición neutral para detener el servo

  delay(5001); // Espera 2 segundos

  // Gira el servo hacia atrás (200° proporcional)

  servoPuerta.write(180); // Máxima rotación hacia atrás

  delay(151); // Espera 200 ms mientras se mueve

  // Detener el servo nuevamente

  servoPuerta.write(90); // Detener el servo

}

void loop() {

  digitalWrite(trigPin, HIGH); // Activamos el pin trig

  delay(10);                   // Esperamos un poco

  digitalWrite(trigPin, LOW);  // Lo apagamos

  long t = pulseIn(echoPin, HIGH); // Long para precisión, activamos la lectura del echo

  int distancia = t / 59;          // Distancia según el manual

  // Mostrar la distancia en el monitor serial

  Serial.print("Distancia: ");

  Serial.print(distancia);

  Serial.println(" cm");

  float peso = scale.get\_units(10);

  Serial.print("Peso: ");

  Serial.print(peso);

  Serial.println(" gramos");

  delay(100);

  // Control de los LEDs en función de la distancia

  if (distancia > 5) {

    // Cerca: Encender LED verde

    digitalWrite(ledVerde, LOW);

    digitalWrite(ledAmarillo, LOW);

    digitalWrite(ledRojo, HIGH);

  } else if (distancia <= 5 && distancia > 2.5) {

    // Medio: Encender LED amarillo

    digitalWrite(ledVerde, LOW);

    digitalWrite(ledAmarillo, HIGH);

    digitalWrite(ledRojo, LOW);

  } else {

    // Lejos: Encender LED rojo

    digitalWrite(ledVerde, HIGH);

    digitalWrite(ledAmarillo, LOW);

    digitalWrite(ledRojo, LOW);

  }

  if (digitalRead(touch) == HIGH && !activado) { // Si el sensor táctil está activado y aún no se ha activado

    digitalWrite(ledAzul, HIGH);

    if (peso < 40) {

      activarServo(); // Llama a la función para mover el servo

      activado = true; // Cambia el estado para evitar múltiples activaciones

      delay(300);

    }

  }

  else if (digitalRead(touch) == LOW) { // Si el sensor táctil no está activado

    digitalWrite(ledAzul, LOW);

    activado = false; // Resetea la variable para permitir la próxima activación

  }

  // Pequeña pausa antes de la siguiente medición

}