#include <LiquidCrystal\_I2C.h>

#include <HX711.h>

#include <Servo.h>

const int irSensorPin = 45; // Pin for IR sensor

const int servoPin = 44;    // Pin for servo motor

Servo myServo;

int sensorValue = 0;

// Define constants

const int lcdAddr = 0x27; //lcd

const int lcdSda = A4; //lcd

const int lcdScl = A5; //lcd

const int stepPin1 = 2; //stepper motor 1

const int dirPin1 = 3; //stepper motor 1

const int stepPin2 = 4; //stepper motor 2

const int dirPin2 = 5; //stepper motor 2

const int stepPin3 = 6; //stepper motor 3

const int dirPin3 = 7; //stepper motor 3

const int LOADCELL\_DOUT = 27; //loadcell

const int LOADCELL\_SCK = 28; //loadcell

const int in1 = 30; //!!

const int in2 = 31; //!!

const int en = 29; //!!

const int ledRedPin = 25; //red LED

const int ledGreenPin = 22; //green LED

const int ledWhitePin = 23; //white LED

const int ledBluePin = 24; //blue LED

const int buzzerPin = 26; //buzzer

const int TRIG\_PIN2 = 13;  // Ultrasonic sensor for starch

const int ECHO\_PIN2 = 10;  // Ultrasonic sensor for starch

const int TRIG\_PIN1 = 12; // Ultrasonic sensor for water

const int ECHO\_PIN1 = 11; // Ultrasonic sensor for water

const int RELAY\_PIN1 = 32; // water pump

const int RELAY\_PIN2 = 33; // dc motor

const float calibrationFactor = 0.000001; //loadcell

const float offset = 0.02; //loadcell

const float boxWeight = 0.63; //loadcell

// Initialize objects

LiquidCrystal\_I2C lcd(0x27, 16, 3);

HX711 scale;

bool machineOpen = false;

bool stepperRunning1 = false;

bool stepperRunning2 = false;

bool stepperRunning3 = false;

bool actuatorExtended = false;

bool steppercounterRunning2 = false;

unsigned long actuatorExtendTime = 0;

long duration\_us;

int distance\_cm;

long duration\_us2;

int distance\_cm2;

void setup() {

  Serial.begin(9600);

  // Servo setup

  myServo.attach(servoPin);

  myServo.write(90); // Start at 0 degrees

  // IR sensor setup

  pinMode(irSensorPin, INPUT);

  pinMode(stepPin1, OUTPUT);

  pinMode(dirPin1, OUTPUT);

  pinMode(stepPin2, OUTPUT);

  pinMode(dirPin2, OUTPUT);

  pinMode(stepPin3, OUTPUT);

  pinMode(dirPin3, OUTPUT);

  pinMode(in1, OUTPUT);

  pinMode(in2, OUTPUT);

  pinMode(en, OUTPUT);

  pinMode(ledRedPin, OUTPUT);

  pinMode(ledGreenPin, OUTPUT);

  pinMode(ledWhitePin, OUTPUT);

  pinMode(ledBluePin, OUTPUT);

  pinMode(buzzerPin, OUTPUT);

  pinMode(TRIG\_PIN1, OUTPUT);

  pinMode(ECHO\_PIN1, INPUT);

  pinMode(TRIG\_PIN2, OUTPUT);

  pinMode(ECHO\_PIN2, INPUT);

  pinMode(RELAY\_PIN1, OUTPUT);

  pinMode(RELAY\_PIN2, OUTPUT);

  digitalWrite(ledGreenPin, HIGH); // Power on indicator

  lcd.init();

  lcd.backlight();

  lcd.setCursor(0, 1);

  lcd.print("Initializing System.");

  delay(2000);

  lcd.setCursor(0, 1);

  lcd.print("WELCOME USER!                ");

  delay(2000);

  lcd.clear();

  Serial.begin(9600);

  scale.begin(LOADCELL\_DOUT, LOADCELL\_SCK);

  scale.set\_gain(128);

  scale.tare();

  stepperRunning1 = true; // Start stepper 1

}

void loop() {

  // IR Sensor Logic

  sensorValue = digitalRead(irSensorPin);

    //water

    digitalWrite(TRIG\_PIN1, LOW);

    delayMicroseconds(2);

    digitalWrite(TRIG\_PIN1, HIGH);

    delayMicroseconds(10);

    digitalWrite(TRIG\_PIN1, LOW);

    duration\_us = pulseIn(ECHO\_PIN1, HIGH);

    distance\_cm = duration\_us \* 0.034 / 2;

    //strach

    digitalWrite(TRIG\_PIN2, LOW);

    delayMicroseconds(2);

    digitalWrite(TRIG\_PIN2, HIGH);

    delayMicroseconds(10);

    digitalWrite(TRIG\_PIN2, LOW);

    duration\_us2 = pulseIn(ECHO\_PIN2, HIGH);

    distance\_cm2 = duration\_us2 \* 0.034 / 2;

  if (sensorValue == LOW) { // Object detected (active-low)

    Serial.println("Object detected! Moving servo to 90 degrees.");

    myServo.write(0); // Rotate servo

    delay(300);

    Serial.println("Returning servo to 0 degrees.");

    myServo.write(90);

    delay(300);

  }

  if (stepperRunning1) {

    rotateClockwise1(360);

    delay(1);

}

  if (steppercounterRunning2) {

    rotatecounterClockwise4();

    delay(1);

  }

  if (stepperRunning2) {

    rotateClockwise2(360);

    delay(1);

  }

  if (stepperRunning3) {

    rotateClockwise3(360);

    delay(1);

  }

  stopStepper2();

  stopStepper3();

  stopStepper1();

  long reading = scale.read();

  float weight = (reading \* calibrationFactor) - offset;

  float netWeight = weight - boxWeight;

  if (netWeight <= 0) {

    netWeight = 0;

    digitalWrite(ledRedPin, HIGH);

    tone(buzzerPin, 1000);

  } else {

    digitalWrite(ledRedPin, LOW);

    noTone(buzzerPin);

    stepperRunning1 = true;

  }

  lcd.setCursor(0, 1);

  lcd.println("Grinding                                                     ");

  lcd.setCursor(0, 2);

  lcd.print("Weight: ");

  lcd.print(netWeight);

  lcd.print("kg   ");

  lcd.setCursor(0, 3);

  lcd.print("of Charcoal");

  noTone(buzzerPin);

  if (netWeight > 1.00) {

    lcd.clear();

    lcd.setCursor(0, 1);

    lcd.println("Pouring                                                      ");

    lcd.setCursor(0, 2);

    lcd.print("Weight: ");

    lcd.print(netWeight);

    lcd.print("kg");

    lcd.setCursor(0, 3);

    lcd.print("of Charcoal");

    }

  if (actuatorExtended && millis() - actuatorExtendTime > 6000) {

    lcd.clear();

    lcd.setCursor(0, 1);

    lcd.println("Retracting                                                   ");

    lcd.setCursor(0, 2);

    lcd.print("let me cook");

    }

  if (netWeight >= 1.00 && stepperRunning1) {

    stopStepper1();

    linearActuatorExtend();

    actuatorExtendTime = millis();

    actuatorExtended = true;

    stepperRunning1 = false;

    noTone(buzzerPin);

  }

  if (actuatorExtended && millis() - actuatorExtendTime >= 6000) {

    linearActuatorRetract();

    stopStepper1();

    actuatorExtended = false;

    clearLine(1);

    lcd.setCursor(0, 1);

    lcd.println("Dispensing Starch                                           ");

    clearLine(2);

    lcd.setCursor(0, 2);

    lcd.println("Dispensing Water      ");

    stepperRunning1 = false;

    noTone(buzzerPin);

    {

    if (distance\_cm > 13) {

    clearLine(2);

    lcd.setCursor(0, 2);

    lcd.println("refill Water      ");

    } else {

    clearLine(2);

    lcd.setCursor(0, 2);

    lcd.println("Dispensing Water      ");

    }

    if (distance\_cm2 > 10) {

    clearLine(1);

    lcd.setCursor(0, 1);

    lcd.println("refill starch                                           ");

    } else {

    clearLine(1);

    lcd.setCursor(0, 1);

    lcd.println("Dispensing Starch                                           ");

    }

  if (distance\_cm > 13 && distance\_cm2 > 10) {

  digitalWrite(RELAY\_PIN1, LOW);

  digitalWrite(RELAY\_PIN2, LOW);

  digitalWrite(ledBluePin, HIGH);

  digitalWrite(ledWhitePin, HIGH);

  tone(buzzerPin, 1000);

  stepperRunning2 = false;

  stepperRunning3 = false;

  steppercounterRunning2 = false;

} else {

  digitalWrite(RELAY\_PIN1, HIGH);

  delay(8000);

  digitalWrite(RELAY\_PIN2, HIGH);

  delay(8000);

  digitalWrite(RELAY\_PIN1, LOW);

  digitalWrite(RELAY\_PIN2, LOW);

  digitalWrite(ledBluePin, LOW);

  digitalWrite(ledWhitePin, LOW);

  noTone(buzzerPin);

  steppercounterRunning2 = true;

  stepperRunning2 = true;

  stepperRunning3 = true;

  stepperRunning1 = false;

  stopStepper1();

}

  }

    stepperRunning1 = false;

    stopStepper1();

  }

  Serial.print("distance: ");

  Serial.print(distance\_cm);

  Serial.println(" cm");

  delay(1);

  Serial.print("distance2: ");

  Serial.print(distance\_cm2);

  Serial.println(" cm");

  delay(1);

}

void rotateClockwise1(int degrees) {

  digitalWrite(dirPin1, HIGH);

  int stepsPerRevolution = 1600;

  int steps = map(degrees, 0, 360, 0, stepsPerRevolution);

  for (int i = 0; i < steps; i++) {

    digitalWrite(stepPin1, HIGH);

    delayMicroseconds(1000);

    digitalWrite(stepPin1, LOW);

    delayMicroseconds(1000);

  }

}

void rotateClockwise2(int degrees) {

lcd.setCursor(0, 1);

lcd.println("Pls wait.......");

lcd.setCursor(0, 2);

lcd.println("processing....");

digitalWrite(dirPin2, HIGH);

int stepsPerRevolution = 3200;

int steps = map(degrees, 0, 360, 0, stepsPerRevolution);

for (int i = 0; i < steps; i++) {

digitalWrite(stepPin2, HIGH);

delayMicroseconds(1000);

digitalWrite(stepPin2, LOW);

delayMicroseconds(1000);

stepperRunning1 = false;

}

}

void rotatecounterClockwise4() {

  steppercounterRunning2 = true;

    digitalWrite(dirPin2, LOW); // Set direction to counterclockwise

    int stepsPerRevolution = 1600; // Total steps for a full revolution

    int rotationsPerMinute = 60;  // Example: 60 RPM (Adjust this based on your motor's speed)

    int totalSteps = stepsPerRevolution \* rotationsPerMinute \* 1; // Total steps in 5 minutes

    // Calculate delay between steps based on RPM

    int delayBetweenSteps = (30 \* 1000000L) / (stepsPerRevolution \* rotationsPerMinute);

    clearLine(1);

    lcd.setCursor(0, 1);

    lcd.println("Pls wait");

    clearLine(2);

    lcd.setCursor(0, 2);

    lcd.println("mixing");

    for (int i = 0; i < totalSteps; i++) {

        digitalWrite(stepPin2, HIGH); // Step signal HIGH

        delayMicroseconds(delayBetweenSteps / 2); // Half-cycle delay

        digitalWrite(stepPin2, LOW); // Step signal LOW

        delayMicroseconds(delayBetweenSteps / 2); // Half-cycle delay

        steppercounterRunning2 = false;

        digitalWrite(RELAY\_PIN1, LOW);

      digitalWrite(RELAY\_PIN2, LOW);

    }

}

void rotateClockwise3(int degrees) {

digitalWrite(dirPin3, HIGH);

int stepsPerRevolution = 3200;

int steps = map(degrees, 0, 360, 0, stepsPerRevolution);

for (int i = 0; i < steps; i++) {

digitalWrite(stepPin3, HIGH);

delayMicroseconds(1000);

digitalWrite(stepPin3, LOW);

delayMicroseconds(1000);

stepperRunning1 = false;

}

}

void linearActuatorExtend() {

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

analogWrite(en, 200);

delay(12000);

lcd.setCursor(0, 1);

lcd.println("Pls wait.......");

lcd.setCursor(0, 2);

lcd.println("pouring charcoal....");

stepperRunning1 = false;

}

void linearActuatorRetract() {

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

analogWrite(en, 200);

delay(14000);

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

analogWrite(en, 0);

stopStepper1();

lcd.setCursor(0, 1);

lcd.println("Pls wait.......");

lcd.setCursor(0, 2);

lcd.println("let me cook...");

stepperRunning1 = false;

noTone(buzzerPin);

}

void stopStepper1() {

digitalWrite(stepPin1, LOW);

digitalWrite(dirPin1, LOW);

}

void stopStepper2() {

digitalWrite(stepPin2, LOW);

digitalWrite(dirPin2, LOW);

}

void stopStepper3() {

digitalWrite(stepPin3, LOW);

digitalWrite(dirPin3, LOW);

}

void clearLine(int line) {

  lcd.setCursor(0, line);

  lcd.print("                    "); // Clear the line (20 characters for a typical LCD)

}