#include "HX711.h"

#include <Servo.h>

// Configuration

#define calibration\_factor -7050.0 // This value is obtained using the HX711\_Calibration sketch

int trialRuns = 3; // This defines the number of times to measure the DC motor's force on the

// load cell. These measurments will be averaged to come up with a single reading.

float armLength = 5.75; // This is the distance (in cm) between the DC motor's center of rotation and the load cell.

// This is based on the mechanical design of the test fixture.

// Define connections between the HX711 and the Arduino

#define DT 3 // The HX711 DT pin connects to D3 on the Arduino

#define SCK 2 // The HX711 SCK pin connects to D2 on the Arduino

// Initialize the HX711

HX711 loadCell(DT, SCK);

// Create a servo object

Servo testServo;

void setup() {

// Begin Serial communication

Serial.begin(9600);

Serial.println(" - Torque Measurement Tool - "); // Print a heading

Serial.println();

//pin to control DC motor

pinMode(11,OUTPUT);

pinMode(12,OUTPUT);

// Set the pin used to control the servo

testServo.attach(9);

loadCell.set\_scale(calibration\_factor); // This value is obtained by using the HX711\_Calibration sketch

loadCell.tare(); // Reset the scale to zero to compensate for any existing load

Serial.println("Initialization complete.");

Serial.println("Send 's' to begin testing. Send 'h' for help.");

Serial.println();

}

void loop() {

// If the user sends 's' over Serial, begin testing the torque

if(Serial.read() == 's' || Serial.read() == 'S') {

measureTorque();

sortMotor();

}

// If the user sends 'i' over Serial, show some instructions

if(Serial.read() == 'h' || Serial.read() == 'h') {

Serial.println("Right now, the Arduino has moved the DC motor into its starting position.");

Serial.println("In the DC motor's current position, install the horn so that it is just touching the load cell.");

Serial.println("When you are ready, send 's' over Serial and the Arduino will begin testing the DC motor's torque.");

Serial.println("You will see the Arduino move the DC motor into the load cell five different times.");

Serial.println("Each time the DC motor hits the load cell, the Arduino will take a reading.");

Serial.println("Those readings will be averaged to calculate the torque delivered by the DC motor.");

Serial.println("Keep an eye on the Serial monitor to see the results.");

Serial.println();

Serial.println("Send 's' to begin testing.");

Serial.println();

}

}

void measureTorque() {

/\*

To test the DC motor's torque, the Arduino will move the motor arm so that it presses on the load

cell. The resulting force will produce a reading from the load cell. The Arduino will take

three readings to compute an average force value. Because the distance between the DC motor's

center of rotation and the load cell is known from the frame design, the Arduino

can calculate the torque produced by the DC motor.

\*/

Serial.println("Individual Readings: ");

float individualReadings[trialRuns]; // This array will store the load cell readings for the five tests

for(int i = 0; i < 3; i++) {

loadCell.tare(); // Reset the scale to zero to compensate for any existing load

digitalWrite(11,HIGH);

digitalWrite(12,LOW);

delay(1000); // Move the DC motor into the load cell so the DC motor will be exerting force on

// the load cell.

digitalWrite(11,LOW);

digitalWrite(12,HIGH);//Move the motor wing away from the load cell

delay(1000);

digitalWrite(11,LOW);

digitalWrite(12,LOW);

individualReadings[i] = loadCell.get\_units(); // Take a measurment from the load cell

delay(1000);

digitalWrite(11,LOW);

digitalWrite(12,LOW);

individualReadings[i] = loadCell.get\_units(); // Take a measurment from the load cell

Serial.print(individualReadings[i]); // Print the measurment over Serial

Serial.print(" ");

}

// Now that we have three individual readings, average them to get one average load reading

float readingsSum = 0; // Create a variable to store the sum of all readings

// Loop through the array and add together all the readings

for(int y = 0; y < trialRuns; y++) {

readingsSum = readingsSum + individualReadings[y];

}

float averageReading = readingsSum / trialRuns; // Divide by the numer of readings to get the average

Serial.println();

Serial.println();

Serial.println("Average Reading:"); // Print the average reading over Serial

Serial.println(averageReading);

// From the average reading, calculate the torque delivered by the DC motor

// using the formula T = F \* r where T is the torque, F is the load cell

// reading (a force), and r is the radius of rotation (the distance between

// the DC motor and the load cell).

// The units for the torque will be kg\*cm

float servoTorque = averageReading \* armLength; // Calculate the torque

Serial.println();

Serial.println("Torque:"); // Print the torque

Serial.print(servoTorque);

Serial.println(" kgcm");

}

void sortMotor(){

float servoTorque;

if(servoTorque>3){

testServo.write(90);

delay(5000);

testServo.write(0);

}

else{

testServo.write(-90);

delay(5000);

testServo.write(0);

}

}