**INTERFACING**

#include "HX711.h"#include<SoftwareSerial.h>SoftwareSerial mySerial(10, 11);#include <LiquidCrystal.h>#define DOUT 3#define CLK 2////////////////////////////////////////////////////#define DOUT1 10#define CLK1 13//////////////////////////////////////////////////////////const int rs = 12, en = 11, d4 = 4, d5 = 5, d6 = 6, d7 = 7;LiquidCrystal lcd(rs, en, d4, d5, d6, d7);HX711 scale;float calibration\_factor = 7050; //-7050 worked for my 440lb max scale setupconst int trigPin = 8;const int echoPin = 9;long duration;int distance1;const int ph= A0;int PhValue = 0;float specific\_gravity;

void setup() { Serial.begin(9600); mySerial.begin(9600); delay(1000); lcd.begin(16,2); pinMode(trigPin, OUTPUT);pinMode(echoPin, INPUT); //Serial.println("HX711 calibration sketch"); //Serial.println("Remove all weight from scale"); //Serial.println("After readings begin, place known weight on scale"); //Serial.println("Press + or a to increase calibration factor"); //Serial.println("Press - or z to decrease calibration factor"); delay(1000); scale.begin(DOUT1, CLK1); scale.set\_scale(1); scale.tare(); //

//////////////////////////////////////////////////Reset the scale to 0

scale.begin(DOUT, CLK); scale.set\_scale(); scale.tare(); //Reset the scale to 0

///////////////////////////////////////////////////////////////////////////

long zero\_factor = scale.read\_average(); //Get a baseline reading Serial.print("Zero factor: "); //This can be used to remove the need to tare the scale. Useful in permanent scale projects. Serial.println(zero\_factor); delay(1000);}void loop() { //scale.set\_scale(calibration\_factor); //Adjust to this calibration factor //Serial.print("Reading: "); //Serial.print( scale.get\_units(), 1); //Serial.print(" lbs"); //Change this to kg and re-adjust the calibration factor if you follow SI units like a sane person //Serial.print(" calibration\_factor: "); //Serial.print(calibration\_factor); //////////////////////////////////////////////////////////////////////////////////////////////////////////////// Serial.println(); Serial.print("weight:"); // Prints string "Distance" on the LCD Serial.print(scale.get\_units(), 1); // Prints the distance value from the sensor Serial.print(" lbs"); delay(1000); /////////////////////////////////////////////////////////////////////////////////////////////////////////////// Serial.println("weight1: "); // Prints string "Distance" on the LCDSerial.print(scale.get\_units(1), 1); // Prints the distance value from the sensorSerial.println(" lbs");/////////////////////////////////////////////////////////////////////////////////////////////specific\_gravity=(scale.get\_units(1))/(scale.get\_units());Serial.println("specific\_gravity=");Serial.println(specific\_gravity);////////////////////////////////////////////////////////PhValue=analogRead(ph);Serial.println("Ph=");Serial.println(PhValue/100);///////////////////////////////////////////////////////////// lcd.setCursor(0,0); // Sets the location at which subsequent text written to the LCD will be displayed lcd.print("SpGravity: "); // Prints string "Distance" on the LCD lcd.print(specific\_gravity); // Prints the distance value from the sensor //lcd.print(" lbs"); delay(1000); ///////////////////////////////////////////////////////////////////////////////////////////// digitalWrite(trigPin, LOW); delayMicroseconds(2); digitalWrite(trigPin, HIGH); delayMicroseconds(10); digitalWrite(trigPin, LOW);duration = pulseIn(echoPin, HIGH);distance1= duration/58;const int level=(1-(distance1 \*0.0333))\*100;Serial.println("Distance: "); // Prints string "Distance" on the LCDSerial.println(level); // Prints the distance value from the sensorSerial.println(" %");delay(1000);///////////////////////////////////////////////////////////////////////////////////////////lcd.setCursor(0,1); // Sets the location at which subsequent text written to the LCD will be displayedlcd.clear();lcd.print("Level: "); // Prints string "Distance" on the LCDlcd.print(level); // Prints the distance value from the sensorlcd.print("%");delay(1000);lcd.setCursor(0,1);lcd.print("Ph=");lcd.print(PhValue/100);// if(Serial.available())// {// char temp = Serial.read();// if(temp == '+' || temp == 'a')// calibration\_factor += 10;// else if(temp == '-' || temp == 'z')// calibration\_factor -= 10;// } delay(1000);Serial.println("AT");delay(1000);Serial.println("AT+CMGF=1");delay(1000);Serial.println("AT+CMGS=\"+918830833230\""); //CHANGE TO DESTINATION NUMBERdelay(1000);Serial.print("SpGravity: "); // Prints string "Distance" on the LCDSerial.print(specific\_gravity);Serial.print("Level: "); // Prints string "Distance" on the LCDSerial.print(level); // Prints the distance value from the sensorSerial.println("%");Serial.print("Ph=");Serial.print(PhValue/100);Serial.write(26);delay(3000);}#include "HX711.h"#include<SoftwareSerial.h>SoftwareSerial mySerial(10, 11);#include <LiquidCrystal.h>#define DOUT 3#define CLK 2////////////////////////////////////////////////////#define DOUT1 10#define CLK1 13//////////////////////////////////////////////////////////const int rs = 12, en = 11, d4 = 4, d5 = 5, d6 = 6, d7 = 7;LiquidCrystal lcd(rs, en, d4, d5, d6, d7);HX711 scale;float calibration\_factor = 7050; //-7050 worked for my 440lb max scale setupconst int trigPin = 8;const int echoPin = 9;long duration;int distance1;const int ph= A0;int PhValue = 0;float specific\_gravity;void setup() { Serial.begin(9600); mySerial.begin(9600); delay(1000); lcd.begin(16,2); pinMode(trigPin, OUTPUT);pinMode(echoPin, INPUT); //Serial.println("HX711 calibration sketch"); //Serial.println("Remove all weight from scale"); //Serial.println("After readings begin, place known weight on scale"); //Serial.println("Press + or a to increase calibration factor"); //Serial.println("Press - or z to decrease calibration factor"); delay(1000); scale.begin(DOUT1, CLK1); scale.set\_scale(1); scale.tare(); // //////////////////////////////////////////////////Reset the scale to 0 scale.begin(DOUT, CLK); scale.set\_scale(); scale.tare(); //Reset the scale to 0 /////////////////////////////////////////////////////////////////////////// long zero\_factor = scale.read\_average(); //Get a baseline reading Serial.print("Zero factor: "); //This can be used to remove the need to tare the scale. Useful in permanent scale projects. Serial.println(zero\_factor); delay(1000);}void loop() { //scale.set\_scale(calibration\_factor); //Adjust to this calibration factor //Serial.print("Reading: "); //Serial.print( scale.get\_units(), 1); //Serial.print(" lbs"); //Change this to kg and re-adjust the calibration factor if you follow SI units like a sane person //Serial.print(" calibration\_factor: "); //Serial.print(calibration\_factor); //////////////////////////////////////////////////////////////////////////////////////////////////////////////// Serial.println(); Serial.print("weight:"); // Prints string "Distance" on the LCD Serial.print(scale.get\_units(), 1); // Prints the distance value from the sensor Serial.print(" lbs"); delay(1000); /////////////////////////////////////////////////////////////////////////////////////////////////////////////// Serial.println("weight1: "); // Prints string "Distance" on the LCDSerial.print(scale.get\_units(1), 1); // Prints the distance value from the sensorSerial.println(" lbs");/////////////////////////////////////////////////////////////////////////////////////////////specific\_gravity=(scale.get\_units(1))/(scale.get\_units());Serial.println("specific\_gravity=");Serial.println(specific\_gravity);////////////////////////////////////////////////////////PhValue=analogRead(ph);Serial.println("Ph=");Serial.println(PhValue/100);///////////////////////////////////////////////////////////// lcd.setCursor(0,0); // Sets the location at which subsequent text written to the LCD will be displayed lcd.print("SpGravity: "); // Prints string "Distance" on the LCD lcd.print(specific\_gravity); // Prints the distance value from the sensor //lcd.print(" lbs"); delay(1000); ///////////////////////////////////////////////////////////////////////////////////////////// digitalWrite(trigPin, LOW); delayMicroseconds(2); digitalWrite(trigPin, HIGH); delayMicroseconds(10); digitalWrite(trigPin, LOW);duration = pulseIn(echoPin, HIGH);distance1= duration/58;const int level=(1-(distance1 \*0.0333))\*100;Serial.println("Distance: "); // Prints string "Distance" on the LCDSerial.println(level); // Prints the distance value from the sensorSerial.println(" %");delay(1000);///////////////////////////////////////////////////////////////////////////////////////////lcd.setCursor(0,1); // Sets the location at which subsequent text written to the LCD will be displayedlcd.clear();lcd.print("Level: "); // Prints string "Distance" on the LCDlcd.print(level); // Prints the distance value from the sensorlcd.print("%");delay(1000);lcd.setCursor(0,1);lcd.print("Ph=");lcd.print(PhValue/100);// if(Serial.available())// {// char temp = Serial.read();// if(temp == '+' || temp == 'a')// calibration\_factor += 10;// else if(temp == '-' || temp == 'z')// calibration\_factor -= 10;// } delay(1000);Serial.println("AT");delay(1000);Serial.println("AT+CMGF=1");delay(1000);Serial.println("AT+CMGS=\"+918830833230\""); //CHANGE TO DESTINATION NUMBERdelay(1000);Serial.print("SpGravity: "); // Prints string "Distance" on the LCDSerial.print(specific\_gravity);Serial.print("Level: "); // Prints string "Distance" on the LCDSerial.print(level); // Prints the distance value from the sensorSerial.println("%");Serial.print("Ph=");Serial.print(PhValue/100);Serial.write(26);delay(3000);}