

```

#include <TFT.h> // Arduino LCD library
#include <SPI.h>
#include "HX711.h"
#include <Adafruit_GFX.h> // Core graphics library
#include <Adafruit_ST7735.h> // Hardware-specific library for ST7735
#include <EEPROM.h>
// pin definition for the Uno

#define cs 10
#define dc 8
#define rst 9

// HX711 circuit wiring

const int LOADCELL_DOUT_PIN = A1;
const int LOADCELL_SCK_PIN = A0;
HX711 scale;

// create an instance of the library
Adafruit_ST7735 tft = Adafruit_ST7735(cs, dc, rst);

// char array to print to the screen
char sensorPrintout[6];

// Define Variables
int shortMean;
unsigned long StoredOffset;

// color definitions
const uint16_t Display_Color_Black = 0x0000;
const uint16_t Display_Color_Blue = 0x001F;
const uint16_t Display_Color_Red = 0xF800;
const uint16_t Display_Color_Green = 0x07E0;
const uint16_t Display_Color_Cyan = 0x07FF;
const uint16_t Display_Color_Magenta = 0xF81F;
const uint16_t Display_Color_Yellow = 0xFFE0;
const uint16_t Display_Color_White = 0xFFFF;
const uint16_t Display_Color_Orange = 0xFB20;

// The colors we actually want to use
uint16_t Display_Text_Color = Display_Color_Orange;
uint16_t Display_Background_Color = Display_Color_Black;
uint16_t Display_Constant_Text_Color = Display_Color_White;
uint16_t Display_Line_Color = Display_Color_Orange;
uint16_t Display_Tare_Text_Color = Display_Color_Orange;
// ***** S E T U P
// *****

void setup() {

// HX711 SETUP

Serial.begin(38400);

scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);

// To calibrate the scale, change the value below.
// Add a known weight to the scale.
// Multiply the current value by the displayed weight (in the serial monitor) then divide by the
known weight.
// Enter the new value below.
scale.set_scale(891.f);

// Buttons Pressed SETUP
pinMode(3, INPUT);
attachInterrupt(1, tarePressed, RISING);

// Get value of of the stored offset (tared spool weight)
EEPROM.get(0, StoredOffset);

```

```

// Display the permanent text
tft.initR(INITR_BLACKTAB);
tft.setRotation(3); // 0 = Portrait (connections bottom) 1 = Landscape (connections RH), 2 =
Portrait (connections top), 3 = Landscape (connections LH)
tft.setFont();
tft.fillScreen(Display_Background_Color);
// tft.drawRect(0, 0, 160, 108, Display_Line_Color);
tft.setTextColor(Display_Constant_Text_Color);
tft.setTextSize(2);
// tft.setCursor(10, 2);
// tft.print("Approx.");
tft.setCursor(70, 62);
tft.print("grams");
tft.drawRect(53, 107, 52, 21, Display_Line_Color);

tft.setTextColor(Display_Tare_Text_Color);
tft.setCursor(56,110);
tft.print("Tare");

// Set the text size for the displayed weight
tft.setTextSize(5);
}

// ***** I S R s *****
*****

// Tare Button Pressed interrupt service routine
void tarePressed()
{
  scale.tare();
  Serial.println(scale.get_offset());
  StoredOffset = (scale.get_offset());
  EEPROM.put(0, StoredOffset);
}

// ***** L O O P *****
*****

void loop() {
int i, n;
double val, sum, mean;
double DStoredOffset = StoredOffset;
n = 20;
i = sum = 0;
while (i<n) {
  val = (((scale.read() - DStoredOffset) / scale.get_scale()));
  sum += val;
  i++;
}
mean = sum / n;
shortMean = mean;

// Send readings to the serial monitor
Serial.print(i); Serial.print(" readings:\t");
Serial.println(sum / n, 3);

// Clear the weight text
tft.setTextColor(Display_Background_Color);
tft.setCursor(40, 18);
tft.print(sensorPrintout);

// convert the reading to a char array
String sensorVal = String(shortMean);
sensorVal.toCharArray(sensorPrintout, 6);

// Display the weight
tft.setTextColor(Display_Text_Color);

```

```
tft.setCursor(40, 18);  
tft.print(sensorPrintout);
```

```
// wait for a moment  
delay(250);
```

```
}
```