JMC FILAMENT SCALE

December 9, 2021

GENERAL

Like many other 3D printer users, I often find myself in a situation where I'm not sure if the spool I want to use has enough filament left to complete a print. It is a guessing game in which I too often lose. As a result, I created a scale that gives an accurate estimate of the amount of filament left on a given spool. A quick internet search found several projects which were designed to do just that. However, none of them were exactly what I was looking for. Most of the projects were very well done. Here are some of the scales that I especially liked:

- **Filament Health Monitor** at https://replicantfx.com/filament-health-monitor/ . This scale gives humidity, temperature, and automatic detection of the color of the filament that is loaded..
- **Build a 3D Printer Filament Scale** in Nuts and Volts Magazine 2019/Issue-4. For Nuts and Volts subscribers it can be found at https://nutsvolts.texterity.com/nutsvolts/201904/MobilePagedReplica.action?pm=2&folio=44#pg44 This scale is very well done and very well documented. Some notable features are a database for many common filament types, latest settings are preserved through power cycles, and a nice display that uses a rotary encoder for selection.
- **Filament Holder and Scales** at https://www.youmagine.com/designs/filament-holder-and-scales . This scale has fairly simple functionality, but I liked the 3D printed spool holder it presented, and I used it in my project.
- **3D Printer Filament Measuring with Arduino** at https://www.electroschematics.com/3d-printer-filament-measuring-arduino/ . This scale is nice, but simple.

FEATURES

The scale presented here has the following features that I deemed useful:

- Accurate measurement of weight up to 5 kg with units selection of grams (g), kilograms (kg), ounces (oz), or pounds (lb).
- Accurate measure of filament length based on spool weight and filament density. Length units are selectable as millimeters (mm), centimeters (cm), meters (m), inches (in), feet (ft), or yards (yd).
- Extensive user interface using a color 1.8" TFT display and rotary encoder.
- Menu driven option settings.
- Displays environmental data of temperature and humidity with selectable Fahrenheit/Celsius selection.
- Database of up to 15 spools with associated ID string, empty spool weight, filament color, filament type, filament density, and filament diameter.
- Database of default filament densities with 12 built-in default densities, and 3 user definable densities.

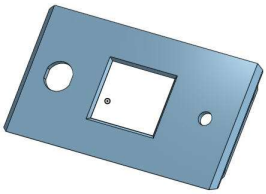
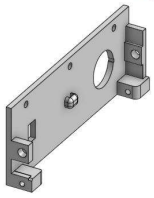
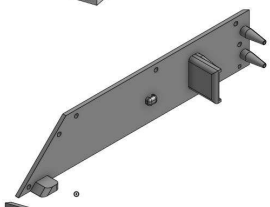
- WiFi compatible, with built-in web page.
- Non-volatile storage of all user settings which are preserved through power cycles.

PARTS LIST

All plastic parts can be 3d printed with no support. See Thingiverse #5164448 . https://www.thingiverse.com/thing:5164448. All files for the project can also be found on github at: https://github.com/regnaDkciN/Filament-Scale. Links are provided for all other parts. These are only suggestions. In most cases, other parts can be substituted.

Plastic (3D Printed) Parts

Table 1: Plastic Parts

Part	Quantity	Picture / Link
Spool Holder	1	Link
Front	1	
Back	1	
Left	1	
Right	1	
Top	1	
Load Cell Mount	1	

Electronic / Miscellaneous Parts

Table 2: Hardware Parts

Part	Quantity	Link	Comment
Adafruit HUZZAH32 – ESP32 Feather	1	Link	
Adafruit 1.8” TFT Display	1	Link	The linked display is preferred.
HX711 ADC Converter with 5KG Load Cell	1	Link	
DHT22 Temperature and Humidity Sensor	1	Link	Any brand should work.
Rotary Encoder with Switch Module	1	Link	Any encoder module.
SPST N.O. Pushbutton Switch (16mm)	1	Link	Any switch of the same size.
DPDT Rocker Switch (3/4”)	1	Link	Any similar switch.
Resistor, 10K 1/8 Watt	1	-	
5V Power Supply with Micro USB Connector	1	Link	Any similar supply.
2-Pin Connector, 0.1”	1	Link	Optional.
3-Pin Connector, 0.1”	1	Link	Optional.
4-Pin Connector, 0.1”	3	Link	Optional.
10-Pin Connector, 0.1”	1	Link	Optional.
Hookup Wire	-		Any 26AWG or larger.
608ZZ Bearing	4	Link	For Spool Holder.
Assorted M2 and M3 Screws	-		For mounting electronics.
M3 Threaded Brass Inserts	15	Link	
M3 x 6mm Screw	15		For scale base.
M3 x 12mm Screw	6		For spool holder.
½” Self-Adhesive Foam Pads	24	Link	Only need 4
½” Rubber Grommet	1	Link	Only need 1
4” Cable Ties (100 pack)	1	Link	Only need about 4.
3/32” Heat Shrink Tubing	1	Link	
Knob for Rotary Encoder	1		

REQUIRED TOOLS

- Soldering Iron
- Solder
- Side Cutters
- Needle Nose Pliers
- Small Wrench or Pliers
- Knife and/or Wire Strippers
- Small Phillips Screw Driver
- Small Files or Sand Paper
- Multimeter (optional)
- Computer with Arduino V1.8.13 or Later Software Installed

- Micro USB Cable (for HUZZAH32 programming)
- 3d Printer

PLASTIC PARTS

Spool Holder

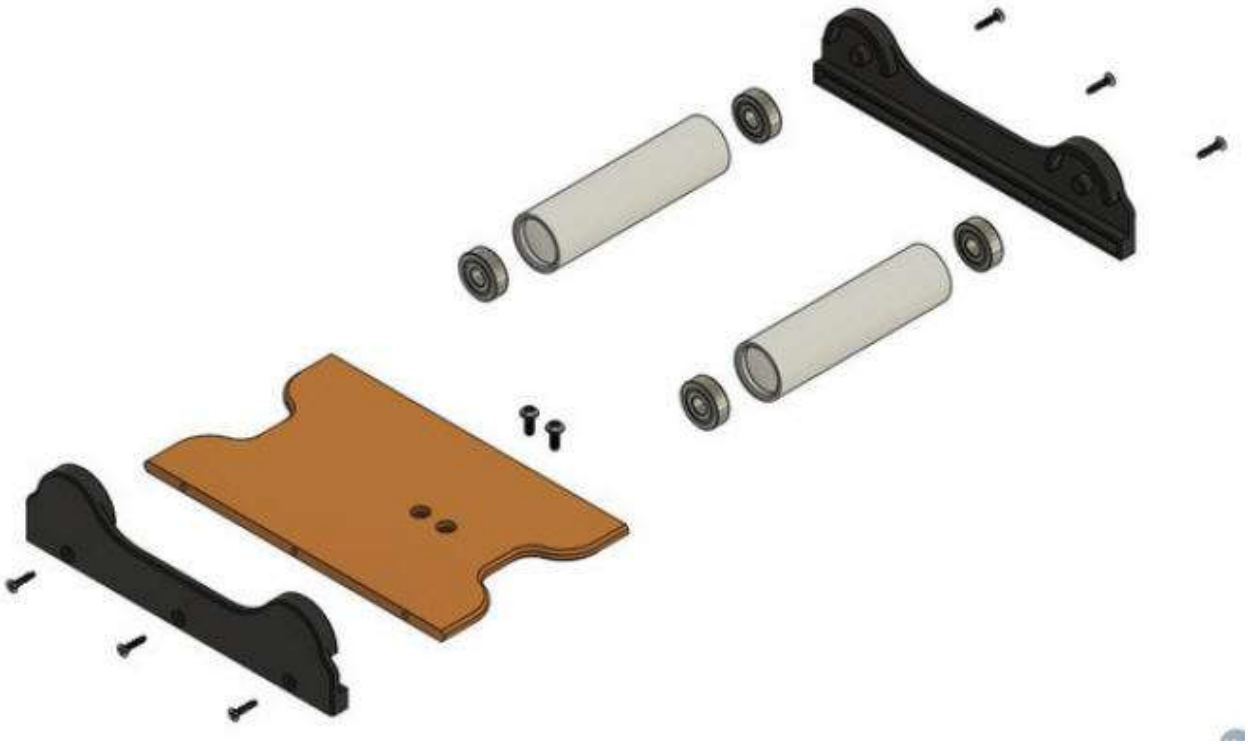
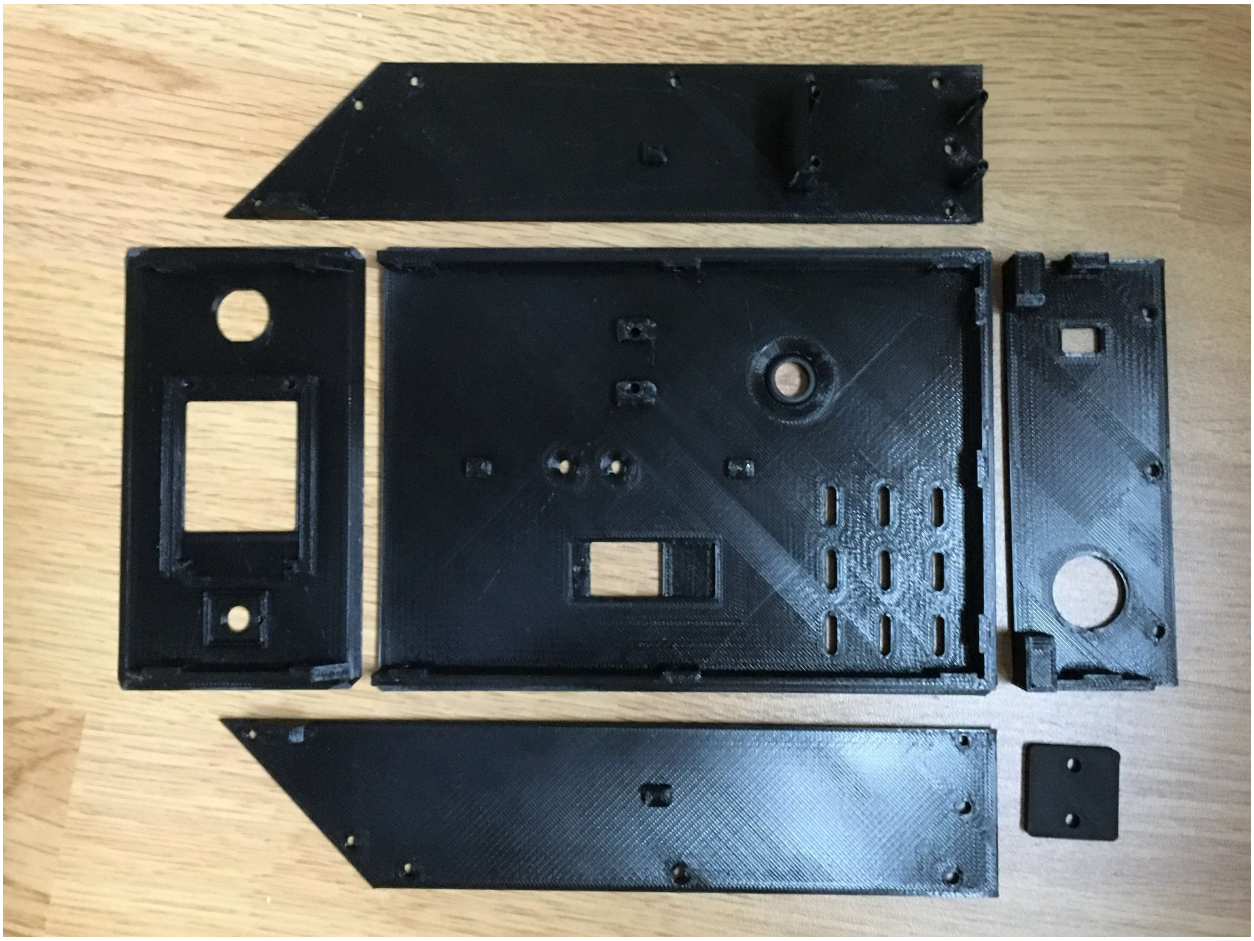
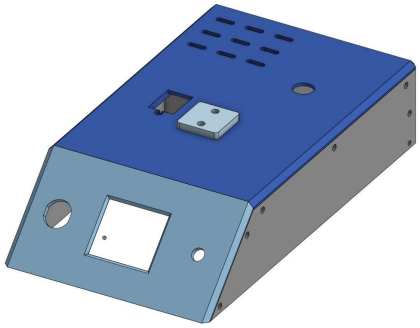


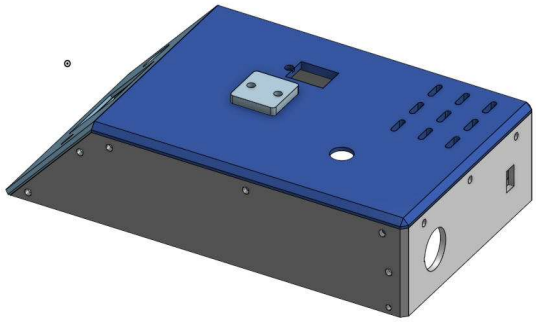
Figure 1: Spool Holder

The 3D files for the spool holder can be found at https://www.youmagine.com/designs/filament-holder-and-scales as part of a different filament scale project by moXDesign. The only parts needed from moXDesign's design are the Bearing Side Housing, the Bed, and the Roller. The four 608ZZ bearings are used in the spool holder to provide smooth operation. I used 20% infill for all the spool holder parts. The Bearing Side Housing and Bed parts do not need supports or any other special printer settings. However, the Rollers will need supports since they are printed on end.

Scale Base







The 3D printed parts for the scale base can be found at https://www.thingiverse.com/thing:5164448. The base was developed using OnShape, and can be found at the following address:

https://cad.onshape.com/documents/4d66991c1357f64191cafd03/w/26c83d555c56fd81e85680c5/e/a300c1e193726131c11e6a15?renderMode=0&uiState=6185397788c4d04ef803d062

To print the base parts, I recommend using an infill of 25% and no supports. Note that the STL files for the base may not be oriented for best printing. You will have to reorient them as needed. The slope of the front panel is 45 degree. In order to print it, it will need to be rotated -135 degrees about the X axis in order to lay it flat for printing. Rotation needed for the rest of the parts should be obvious.

Threaded Inserts

The front, back, and top pieces of the scale base need to have the M3 Threaded Brass Inserts installed. The following diagram shows the locations to add the inserts. Note that inserts need to be added to both sides of the front, back and top panels. A total of 15 inserts are needed.

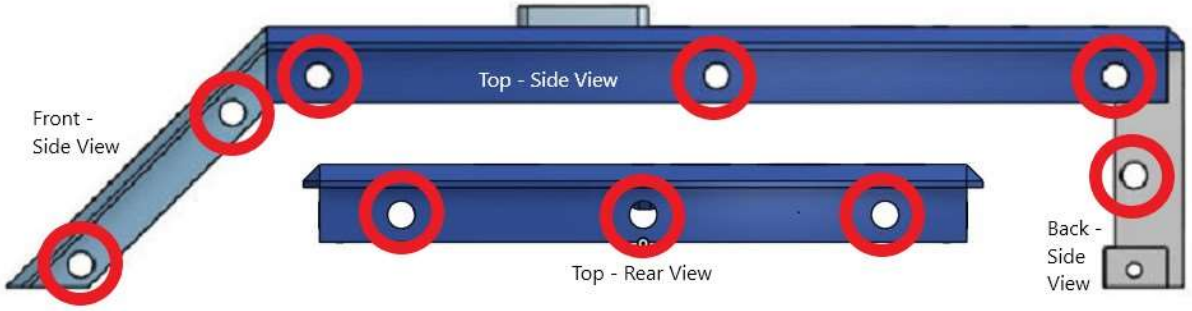
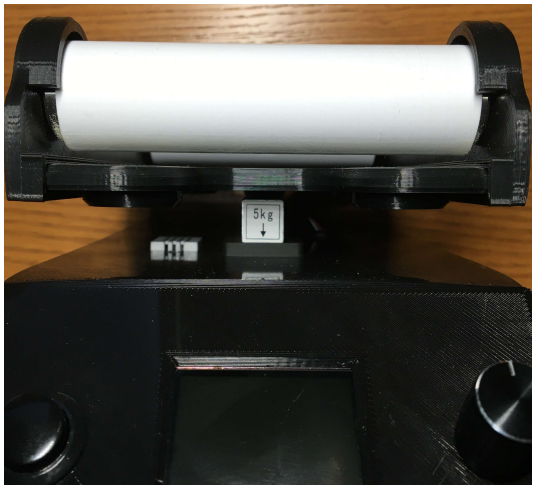


Figure 2: Threaded Insert Locations

Load Cell Mount

The Load Cell Mount (3D printed part) must be inserted between the base and the load cell itself. The load cell must be mounted in the correct orientation. It should contain a label with an arrow. The arrow must point down as shown:



WIRING

The wiring diagram and wiring table are given below. The wiring diagram was created using Fritzing, and the associated .fzz file is included. Please note the following:

- Rear Rocker Switch should use the normally closed contacts. The reason for this is that power for the system comes via the HUZZAH32 micro USB connector. There is no way to switch this power to completely disable the board. However, the HUZZAH32 contains an on-board 3.3V

regulator which supplies power to the entire HUZZAH32 board as well as the rest of the peripherals for the scale (HX711, encoder, …). Adafruit provided the EN connection, which when pulled to ground disables the regulator and thus the entire system. The EN pin is normally pulled high, except when the rear switch is turned on, which grounds the EN pin and turns off (most) the power to the system.

- The load cell is not shown in the wiring diagram, and the colors of the wires from the load cell may be different from different vendors, so some experimentation may be needed.
- Be sure to connect all the wires labeled (+) together.
- Be sure to connect all the wires labeled (Gnd) together.

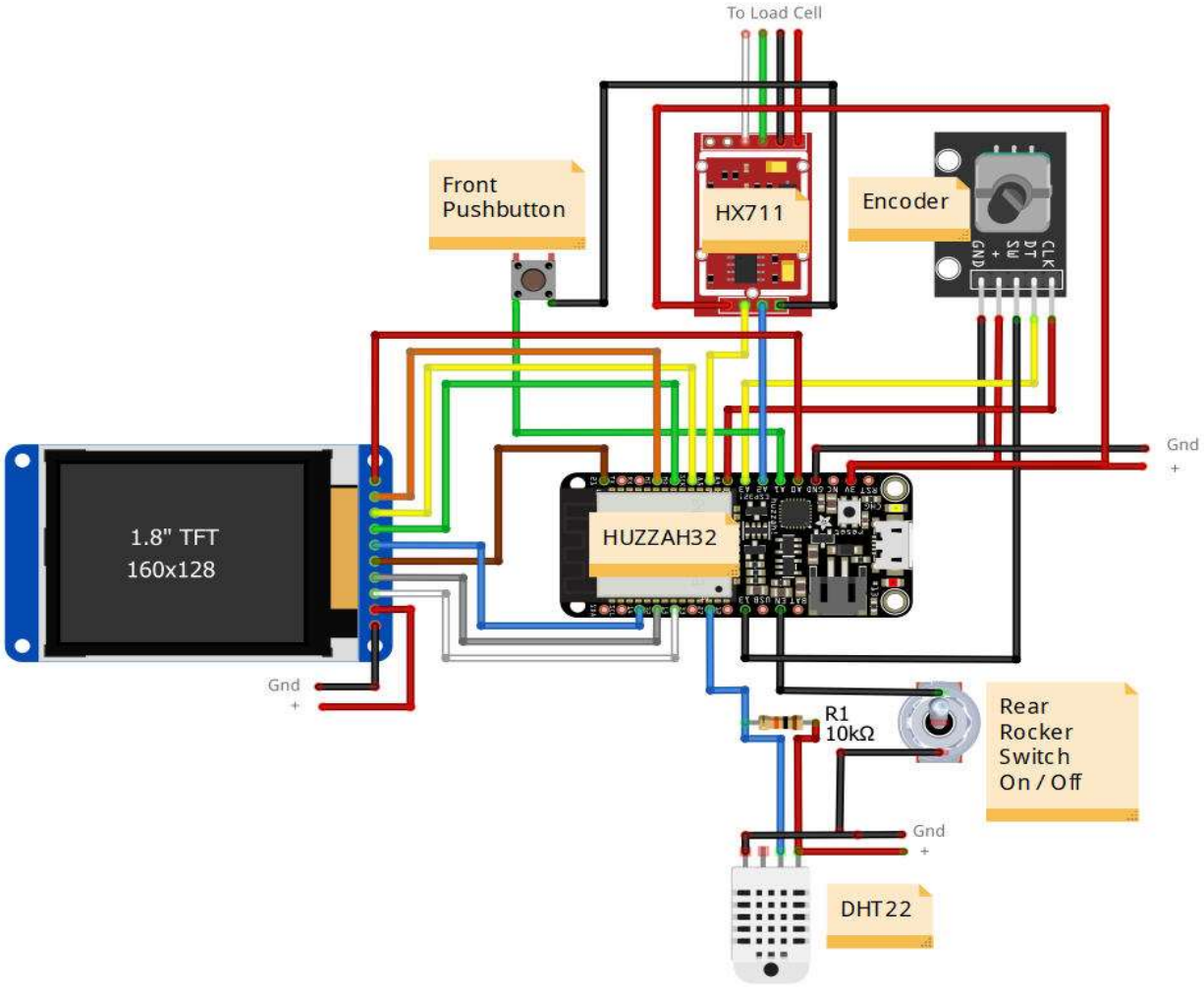


Figure 3: Wiring Diagram

Table 3: Wiring Table

PIN	CONNECTED TO	WIRE COLOR
HUZZAH A0 – GPIO 26	TFT LITE	Red
HUZZAH A1 – GPIO 25	FRONT PB	Green
HUZZAH A2 – GPIO 34	HX711 DOUT	Blue
HUZZAH A3 – GPIO 39	ENCODER DT	Yellow
HUZZAH A4 – GPIO 36	ENCODER CLK	Red
HUZZAH A5 – GPIO 4	HX711 SCK	Yellow
HUZZAH GPIO 13	ENCODER SW	Black
HUZZAH GPIO 27	DHT22 SIGNAL	Blue
HUZZAH GPIO 15	TFT RST	White
HUZZAH GPIO 32	TFT DC	Gray
HUZZAH GPIO 14	TFT TFT CS	Blue
HUZZAH GPIO 21	TFT CARD CS	Brown
HUZZAH MOSI	TFT MOSI	Green
HUZZAH SCK	TFT SCLK	Yellow
HUZZAH MISO	TFT MISO	Orange
HUZZAH EN	REAR ROCKER SW (NC)	Black
HUZZAH GND	SYSTEM GROUND	Black
HUZZAH 3V	SYSTEM 3.3V	Red
HX711 E+	LOAD CELL	Red
HX711 E-	LOAD CELL	Black
HX711 A+	LOAD CELL	Green
HX711 A-	LOAD CELL	White

Feel free to use whatever wiring techniques you are familiar with. For this project, I soldered wires directly to the HUZZAH32, and used pre-wired connectors to connect to each of the peripheral devices (except the two switches). The 3D printed base parts all contain built-in anchors for cable ties which are meant to help with tidying up the final wiring. The following picture shows my wiring.

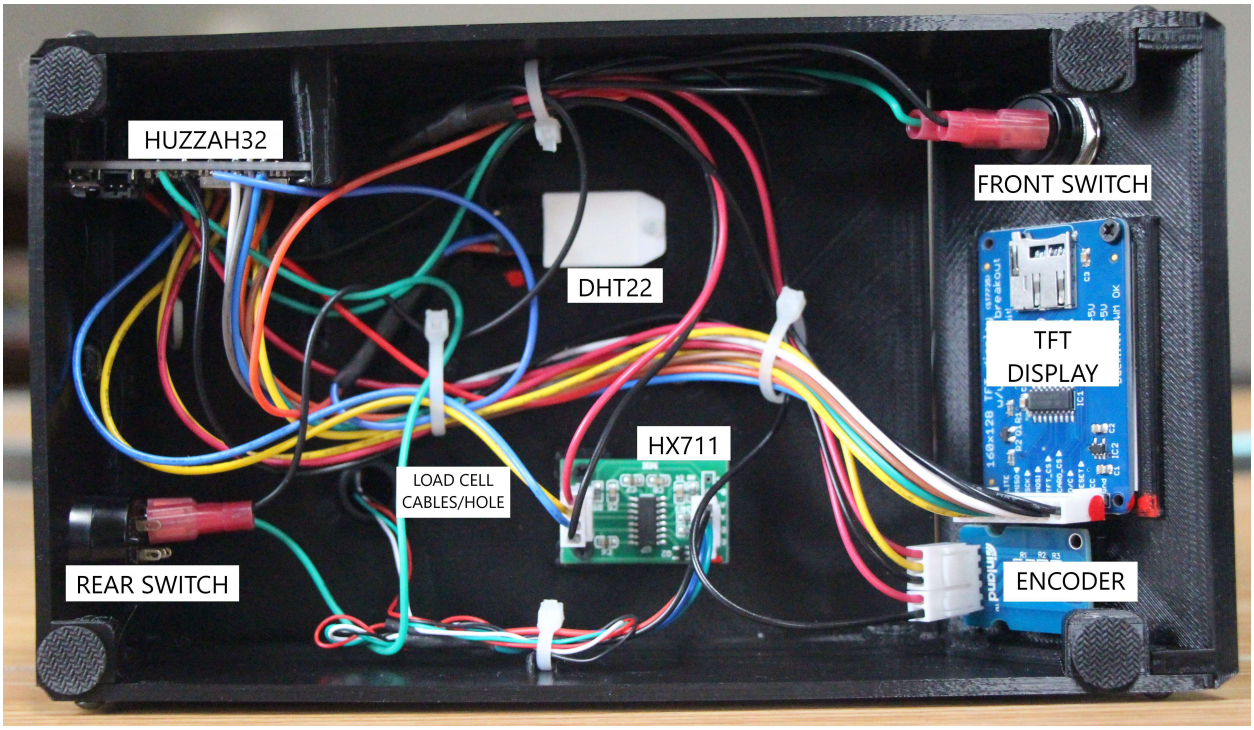


Figure 4: Scale Wiring

Wiring Notes

- The specified rotary encoder module comes with right angled header pins which are meant to enable it to be plugged into a breadboard and sit vertically. In order to have the encoder module fit into the base unit, it is necessary to replace the right angled pins with straight pins on the opposite side of the module circuit board. The removal of the original encoder module pins is fairly easy if you snip the pin base plastic so that the pins can be extracted individually.

- The rotary encoder module has 5 pins. A 5-pin connector could not be found, so two connectors were used to interface to the encoder. A 3-pin connector was used for the data pins, and a 2-pin connector was used for the power pins.

- The pins on the DHT22 need to be bent back in order to accept the connector.

- The leads from the load cell to the HX711 carry low level analog voltages. It is good to twist these leads to reduce noise. It also helps to route these wires away from other wires where possible.

FIRMWARE

Preliminary Work

A few things are needed before work on installing the scale firmware can begin. This section covers the preliminary work.

Install the Arduino Libraries

The following libraries are needed in order to run the main Guitar Chord Chart application.

Table 4: Arduino Libraries

Library	Link
Arduino-esp32	https://github.com/espressif/arduino-esp32#using-through-arduino-ide
ArduinoMenu	https://github.com/neu-rah/ArduinoMenu/wiki
Adafruit GFX	https://github.com/adafruit/Adafruit-GFX-Library
Adafruit ST7735 and XT7789	https://github.com/adafruit/Adafruit-ST7735-Library
ArduinoJson	https://arduinojson.org/?utm_source=meta&utm_medium=library.properties
ESP32Encoder	https://github.com/madhephaestus/ESP32Encoder/
Bounce2	https://github.com/thomasfredericks/Bounce2
DHT sensor library for ESPx	https://desire.giesecke.tk/index.php/2018/01/30/esp32-dht11/
HX711 *	https://github.com/RobTillaart/HX711
ESP_WiFiManager	https://github.com/madhephaestus/Esp32WifiManager
ESP32WebServer	https://platformio.org/lib/show/2007/ESP32WebServer

In general, to install these libraries, use the Arduino library manager. Consult google if you are unsure how to do this.

Special ArduinoMenu Files

Several files in the ArduinoMenu library needed minor editing in order to fix bugs and to suit the needs of the scale. These files are items.cpp, items.h, menuBase.cpp, and menuBase.h. These changed files must be copied into the ArduinoMenu library on your computer. The location of these files should be something like the following:

C:\Users\YOUR_USER_NAME\Documents\Arduino\libraries\ArduinoMenu_library\src

I have submitted issues regarding these changed files to the author of the ArduinoMenu library, but until and unless the author makes the recommended changes, you will need to copy the files as stated above.

Special HX711 Library Files

The HX711 library source files have been modified to add the ‘const’ qualifier to several of the class methods. As a result, it will be necessary to copy the modified files (HX711,cpp and HX711.h) from the “! Special HX711 Library Files” directory to the appropriate Arduino library. The location of these files should be something like the following:

C:\Users\YOUR_USER_NAME\Documents\Arduino\libraries\HX711_Arduino_Library\src

Firmware Files

As with all Arduino projects, the .ino file is the main firmware file which contains the setup() and loop() functions. In this case, the file is called **JmcFilamentScale.ino**. A number of supporting files and libraries are also needed. As mentioned above, several libraries must be installed, as well as modifying two files from the ArduinoMenu library, Beside these, the files listed in the table below must reside in the Arduino compiler's path. It is probably best to have all of these files in the same directory.

Table 5: Supporting Files

File	Class(es)	Behaviour
AuxPb.h	AuxPb	Initializes and monitors a pushbutton.
Display.cpp, Display.h	DisplayState Display	Handles miscellaneous scale related display tasks.
EnvSensor.cpp, EnvSensor.h	EnvSensor	Interfaces to the DHT22 temperature and humidity sensor.
ESP32EncoderStream.h	ESP32EncoderStream	Provides the encoder and pushbutton stream for the ArduinoMenu library.
Filament.cpp Filament.h	Filament	Maintains data pertaining to filament densities.
HslColor.cpp HslColor.h	HslColor	Supports HSL<==>RGB565 conversion.
JmcFilamentScale.h. JmcFilamentScale.ino	SpoolData	Contains constant and variable declarations for shared scale data. Contains the main Arduino loop() and some supporting functions.
LengthManager.cpp LengthManager.h	LengthManager	Contains methods to handle calculation of filament length given weight, density, and filament diameter.
LoadCell.cpp LoadCell.h	LoadCell	Interfaces with the HX711 load cell.
MainScreen.cpp MainScreen.h	MainScreen	Supports the display of data on the main display screen.
MovingAverage.h	MovingAverage	Implements the moving average calculation template class.
Network.cpp Network.h	Network	Handles miscellaneous scale elated network connection tasks.
ScaleIcon.cpp ScaleIcon.h		Contains the bit pattern for the welcome screen icon.
ScaleMenu.cpp, ScaleMenu.h		Contains data related to console menu handling.
SCB.cpp SCB.h	SCB	Contains a SCB (Screen Control Block) for each field on the main console screen.
SimpleLock.h	SimpleLock	Implements a simple interlocking mechanism (mutex) used for WiFi/Console locking.
Spool.cpp	Spool	Maintains data pertaining to the filament contained

File	Class(es)	Behaviour
Spool.h		on a spool.
SpoolManager.h	SpoolManager	Maintains the collection of spools, and manages the selection of a particular spool.
WebData.cpp WebData.h		Supports handling of web requests from the web client.
WebPages.h		Contains the (very long) string that represents the root web page (HTML & Java Script) for the filament scale.

Modifying the Firmware

In general, the scale firmware should not need much, if any, modification. This section describes possible firmware modifications that a user may want to make to customize the scale.

JmcFilamentScale.h

JmcFilamentScale.h contains many global constants which affect the behavior of the scale. Only a few of these constants are worth changing. The following list describes these constants.

- **NUMBER_SPOOLS** - This is the number of spools that is maintained by the SpoolManager database. Due to limitations in the menu system library (ArduinoMenu), the maximum value for this constant is 15.
- **MAIN_PAGE_BG_COLOR, MAIN_PAGE_FG_COLOR** - These two constants specify the background and foreground colors displayed on the console display. The colors are specified in RGB565 format.
- **gNetworkApName –** This specifies the string that is used as the name for the scale while in Access Point mode. Its default value is “ScaleWifiSetup”.
- **gNetworkApIpAddr** – This specifies the IP address used by the scale while in Access Point mode. Its default value is 192.168.4.1 .

JmcFilamentScale.ino

This file contains the initialization and loop code for the scale. Within it, all of the peripheral devices are declared. Each peripheral has a section near the top of the file to specify its configuration. If the default hardware configuration needs to change, this is the place to do it.

1.8” TFT Display

```
////////////////////////////////////////////////////////////////////////////
// ST7735 1.8TFT 128x160 Pin assignments and related variables.
////////////////////////////////////////////////////////////////////////////
static const int TFT_CS   = 14;     // TFT display CS pin.
static const int TFT_DC   = 32;     // TFT display DC pin.
static const int TFT_RST  = 15;     // TFT display RST pin.
static const int TFT_LITE = A0;     // TFT display backlight pin.
```

Rotary Encoder

```
////////////////////////////////////////////////////////////////////////////
// Rotary encoder pins.
////////////////////////////////////////////////////////////////////////////
static const int ENC_PIN_A      = A3;   // Encoder A pin.
static const int ENC_PIN_B      = A4;   // Encoder B pin.
static const int ENC_BUTTON_PIN = 13;   // Encoder pushbutton pin.
static const int ENC_SENSITIVITY = 4;   // Encoder sensitivity.
```

Auxiliary Pushbutton

```
////////////////////////////////////////////////////////////////////////////
// Auxiliary pushbutton pins.
////////////////////////////////////////////////////////////////////////////
static const int AUX_BUTTON_PIN = A1;   // Auxiliary pushbutton pin.
```

Load Cell (HX711)

```
////////////////////////////////////////////////////////////////////////////
// Define the LoadCell sensor related I/O pins and global data and constants.
////////////////////////////////////////////////////////////////////////////
static const int LOADCELL_DOUT_PIN  = A2;   // LoadCell DOUT signal pin.
static const int LOADCELL_SCK_PIN   = A5;   // LoadCell CLOCK pin.
```

Environmental Sensor (DHT22)

```
////////////////////////////////////////////////////////////////////////////
// Define the environmental sensor (temperature and humidity) pins and global
// data and constants.
////////////////////////////////////////////////////////////////////////////
static const uint8_t ENV_DAT_PIN    = 27;   // Sensor data pin.
static const uint8_t ENV_TYPE       = DHT22;// Sensor type.
```

Scale WiFi Name

```
// Access the scale via this name (i.e. http://JmcScale.local).
const char *gNetworkServerName = "JmcScale";
```

EnvSensor.cpp

The 1.8” TFT display uses a peculiar value to display the degree symbol (°). This value is stored near the top of EnvSensor.cpp as follows:

```
#define DEGREE_SYMBOL "\xF7"
```

If another display is used, this value will probably need to be changed.

Filament.cpp

The default filament densities are stored in an array near the top of Filament.cpp. Changing these values will result in changes to the default filament densities.

```
////////////////////////////////////////////////////////////////////////////
// The density table.  Each entry corresponds to a filament type.  Values
// for the table were taken from an article in Nuts/Volts magazine,
// 2019/issue-4, entitled "Build a 3D Printer Filament Scale".
////////////////////////////////////////////////////////////////////////////
float Filament::m_Densities[NUMBER_FILAMENTS] =
{
//  ABS   ASA   Copr  HIPS  Nyln  PETG  PLA   PMMA
    1.04, 1.07,   3.90, 1.07,  1.08, 1.27,   1.24,   1.18,
//  PlyC  PVA   TPE   TPU   USR1  USR2  USR3
    1.20, 1.19,  1.20,   1.20,  1.24,   1.24,    1.24
}; // End m_Densities[].
```

Spool.cpp

The default values for the default spool filament type, spool weight, filament diameter, and spool name are stored near the top of Spool.cpp.

```
////////////////////////////////////////////////////////////////////////////
// Spool class constants.  Should be self explanatory.
////////////////////////////////////////////////////////////////////////////
const FilamentType Spool::DEFAULT_FILAMENT_TYPE = eFtPla;
const float Spool::DEFAULT_SPOOL_WEIGHT        = 250.0;
const float Spool::DEFAULT_FILAMENT_DIAMETER    = 1.75;
const char *Spool::DEFAULT_SPOOL_NAME          = "Spool";
```

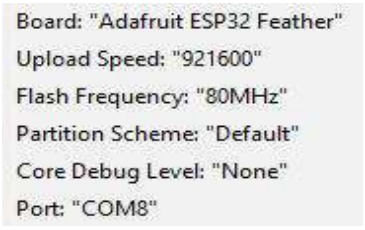
ScaleIcon.cpp

This file contains the bit pattern of the power-up (welcome) screen background. It is 120 x 120 pixels.

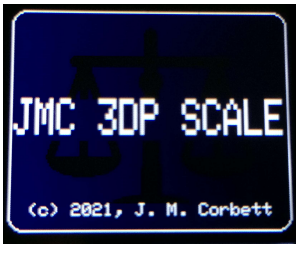
Compiling and Loading the Firmware

Once the preliminary firmware work is done (see above), the code may be compiled and downloaded to the Huzzah. Make sure that all of the included files are within the search path of the Arduino compiler. I just place them in the same directory as the JmcFilamentScale.ino file.

Make sure you use a recent version of the Arduino software. I used version 1.8.13. Use the Tools menu to select the following setup:



Set the Port to the USB port that is connected to your Huzzah. The image above shows “COM8”. Yours may be different. Once this is done, simply compile and download the code to the Huzzah. If all goes well, the following screen should appear:



ADD THE FINAL TOUCHES

Rubber feet may be added to the bottom of the case to keep it from sliding when using the front panel controls. The rubber feet shown in the parts list above work great.

Clean up the wiring by using zip ties.

DESIGN CONSIDERATIONS

As with any design, several options were considered. This section contains the rationale for the selected design.

Filament Color Selection

The initial design employed a TCS34725 color sensor similar to the design used by **Filament Health Monitor** at https://replicantfx.com/filament-health-monitor/. This sensor was meant to detect the color of the filament that placed on the scale, thus eliminating the need for the user to specify the color. It was found that the color could be made to work, but it's reliability and repeatability were not sufficient for the intended use. Ambient light played a major role in affecting the quality of the color reading. I never found a setup that could produce acceptable results in all ambient light conditions. Further, I realized that if the color was saved as part of the spool database data, it would only need to be setup once per spool, and could be used to help select the spool from the database. As a result, the color sensor was replaced by a spool setting that specifies the filament color, and displays the color as part of the spool selection display.

Menu System

There seem to be only two main menu libraries that could be used with the ESP32 processor with the Arduino development system. They are **ArduinoMenu** and **TcMenu**. I tried both menu libraries. TcMenu looked promising, but after several days of trying I could not get it to compile with my setup. This left the ArduinoMenu library. I have mixed thoughts regarding this library:

- I was able to get it to work with my setup, and it has a decent amount of flexibility.
- Its code is built with layer upon layer of macros, which make it extremely hard to understand.
- Comments are practically non-existent in its code.
- It was originally written to support the Arduino family of processors. These processors have very limited memory. As a result, the library traded off features and flexibility for minimal memory utilization. However, the Huzzah's ESP32 processor is not as limited as the Arduino processors, and could easily support more features and capability.
- It is possible to extend the library's limited capabilities via deriving C++ classes. However, due to the lack of comments in the code and the layered use of macros, it requires a C++ expert to figure out how/where to do it. In defense of the library's author, he is quite good at answering questions and supplying information when requested. I just wish that the code was written to be

more understandable so that the author's help would be less of a necessity.

Power Connection

The current design never completely powers the system down, even when the power switch is in the OFF position. This is a consequence of using the Huzzah development board. When the power switch is in the OFF position, a small amount of current is still supplied to the USB-Serial converter on the Huzzah board. Everything else is turned off. It was decided that this behavior was acceptable.

Code Design

Although libraries exist for the various peripherals used by the scale, I chose to further abstract each of their interfaces to supply an interface that more closely meets the system needs. An attempt was made to use good coding practices. For example, the **MISRA** coding standard was followed where practical, and the code is heavily commented. However, even though they are considered evil, the system uses quite a few global variables. This eased the development. The code could use some re-factoring and possibly make better use of name spaces, but I'll leave that for later.

USER MANUAL

The scale contains an extensive user interface that consists of:

- 1.8" color TFT display, referred to as the "Display" or "Screen";
- Rotary encoder with pushbutton, referred to as the "***Encoder***" and "***Encoder PB***";
- Pushbutton, referred to as the "***Units PB***" or "***Tare PB***" or "***Units/Tare PB***" or "***Home PB***" depending on the current state of the display.

Home Page Controls

The ***Encoder***, ***Encoder PB***, and ***Units/Tare PB*** generate different behaviors depending on the currently displayed Screen. The table below describes the behavior of the front panel input devices when the Home Page is displayed.

Table 6: Home Page Controls

CONTROL ACTION	BEHAVIOR
Rotate ***Encoder*** CW	Scroll Home Page data up. The net weight field is always displayed at the top of the Home Page. If a spool is selected, the filament length field is always displayed below the net weight field. The non-fixed fields can be scrolled via the ***Encoder***.
Rotate ***Encoder*** CCW	Scroll Home Page data down. (See above).
Press ***Encoder PB*** (*CLICK)*	Enter the SETUP/OPTIONS page.
Press ***Units/Tare PB*** for less than 1 second	Select the next weight unit. Weight units cycle as follows: grams(g) → kilograms(kg) → ounces(oz) → pounds(lb)
Press ***Units/Tare PB*** for greater than 1 second	Perform a TARE operation. Be sure the scale is unloaded before doing this.

When the Home Page is active, various information is available for display. The ***Encoder*** can be used to scroll through this information, or an option can be set to enable auto scrolling of this information at a specified rate. The net weight field is always displayed at the top of the Home Page. If a spool is selected, the filament length field is always displayed below the net weight field. The remainder of the scale's data fields can be scrolled via rotating the ***Encoder*** or automatically. The following data may be displayed:

- **Net Weight**: Always displayed at the top of the Home Page. Units are selectable. If a spool is selected, then this it the gross weight seen by the scale minus the empty weight of the selected spool. If no spool is selected, then this is the gross weight seen by the scale.

- **Length**: Always displayed below the **Net Weight** field only if a spool has been selected. If no spool is selected, then this field is not displayed. The units displayed for length are selectable by the user.
- **Gross Weight:** This is the total weight seen by the scale. This field is only displayed if a spool is selected. Gross Weight is displayed using the same units as Net Weight.
- **Spool ID:** This is the name of the selected spool as programmed by the user.
- **Spool Weight:** If a spool is selected, this field displays the spool's empty weight as programmed by the user. Spool Weight is displayed using the same units as Net Weight. If no spool is selected, this field is not displayed.
- **Color:** This is the color of the selected spool as programmed by the user. If no spool is selected, this field is not displayed.
- **Fil Type:** This is the filament type related to the selected spool. There are 12 built-in filament types and 3 user definable filament types. The built-in filament types are: ABS; ASA; Copper; HIPS; Nylon; PETG; PLA; PMMA; PolyC; PVA; TPE, and TPU. If no spool is selected, this field is not displayed.
- **Dens:** This is the density of the filament if a spool is selected. This value is selectable by the user. If no spool is selected, this field is not displayed.
- **Dia:** This is the diameter of the filament if a spool is selected. This value is selectable by the user. If no spool is selected, this field is not displayed. The units of this field is always millimeters (mm).
- **Net Name:** This is name of the WiFi network to which the scale is connected. If no network connection exists, "OFFLINE" is displayed in red.
- **IP Addr:** This is the IP address of the scale if connected to the WiFi network. This field is not displayed if the scale is offline.
- **WiFi Signal:** This field displays the signal strength of the WiFi network if online. The field displays 0 to 4 bars that indicate the relative strength. It also displays the strength as a dBm value. This is normally a negative value with numbers closer to zero being better. The field is also color coded with bright green being best strength to red being worst. This field is only displayed if the scale is online.
- **Access Point Net Name:** This is WiFi name to look for when setting up the WiFi network parameters. This is explained later. This field is only displayed if the scale is not connected to a WiFi network.
- **Access Point IP Addr:** This is the WiFi IP address to look for when setting up the WiFi network parameters. This is explained later. This field is only displayed if the scale is not connected to a WiFi network.
- **Temp:** This is the temperature detected by the scale. It can be programmed to display as either Fahrenheit (F) or Celsius (C).

- **Hum:** This is the percent relative humidity detected by the scale.
- **Up Time:** This is the amount of time that has elapsed since the scale was turned on.

Setup/Options Controls

The Setup/Options mode is entered by pressing the ***Encoder PB*** while the Home Page is displayed. This mode allows the scale options to be customized. The table below describes the behavior of the front panel controls while in Setup/Options mode.

Table 7: Setup/Options Controls

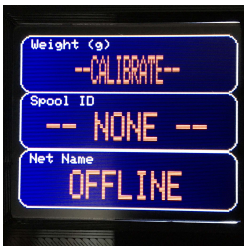
CONTROL ACTION	BEHAVIOR
Rotate ***Encoder*** CW (non-edit field)	Scroll the cursor down.
Rotate ***Encoder*** CCW (non-edit field)	Scroll the cursor up.
Rotate ***Encoder*** CW (edit field)	Increase the numeric value, or cycle to next value.
Rotate ***Encoder*** CCW (edit field)	Decrease the numeric value, or cycle to previous value.
Press ***Encoder PB*** less than 1 second (*CLICK)*	Select an option, cycle to next option, or enter edit mode.
Press ***Encoder PB*** greater than 1 second	Back up one level.
Press ***Home PB***	Exit Setup/Options mode (return to the Home Page)

One of the following methods is used to edit a value in Setup/Options mode:

- **Cyclic Edit Mode:** Some values, like the *Display→Weight* value, will cycle through valid entries with each click of the ***Encoder PB***. For example, repeatedly clicking the ***Encoder PB*** with the Display→Weight field selected will cycle through the valid weight display entries:
  - g→kg→oz→lb→g→ …
- **Numeric Edit Mode:** Most numeric values, like the *Spool→Spool Info→Wt* value, provide a coarse/fine value setting method. With the field highlighted, a click of the ***Encoder PB*** causes entry to numeric edit mode. In this mode, the color of the highlighted value changes to red, and a "!!" symbol appears before the value. The "!!" symbol indicates that the value will change by a large amount with each increment of ***Encoder*** rotation. When a value close to the desired value is reached, a click of the ***Encoder PB*** will cause the "!!" symbol to change to "!". This indicates that the value will change by a small amount with each increment of ***Encoder*** rotation. Once the exact value is reached, a final click of the ***Encoder PB*** will exit edit mode.
- **String Edit Mode:** String edit fields, like the *Spool→Spool Info→Spool ID* string, allow for editing of each individual character of the string. With the string highlighted, a click of the ***Encoder PB*** causes entry to the string edit mode. In this mode, a ":" appears before the existing string and the first character of the string is highlighted. At this point, rotating the ***Encoder*** clockwise will cause the next string character to be selected. Once a character that needs to be changed is

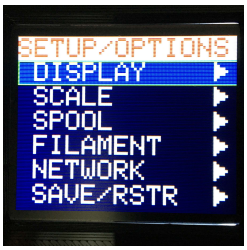
selected, a click of the ***Encoder PB*** will cause the selected character to get a dark background. This indicates that the character may now be changed by rotating the ***Encoder***. Once the desired character appears, clicking the ***Encoder PB*** will cause the new character to be accepted. A new character may now be selected and edited. After the final character has been changed, a final click of the ***Encoder PB*** will exit the string edit mode, and the leading ":" will disappear.

Initial Power-up



On initial power-up, the scale is not calibrated, and it is not connected to WiFi. This is evidenced by the display showing "--CALIBRATE--" in red for the Weight field, and "OFFLINE" in red for the Net Name field. The scale needs to be calibrated before it can display any weight values. If network operation is desired, the local WiFi parameters will also need to be setup. This section describes how to calibrate the scale and connect to the local WiFi.

Scale Calibration

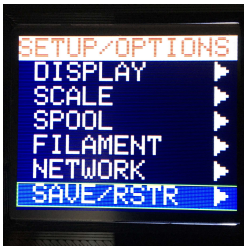


To calibrate the scale, click the ***Encoder PB***, which will bring up this screen. Rotate the ***Encoder*** to select the **SCALE>** option, and click the ***Encoder PB***.

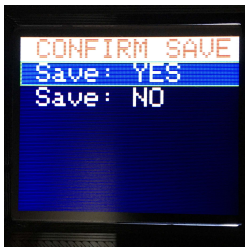
Select the **CALIBRATE>** option.

Remove any load from the scale, then click the ***Encoder PB***. This will perform a tare (zero) operation

Once the tare completes, place a load of known weight on the scale and update the known weight using **Numeric Edit Mode**. Once the known weight has been entered, select **Ready>**.



The scale is now calibrated. In order to save the new calibration values, scroll down to the **SAVE/RSTR>** option and select it.



Select the **SAVE>** option, and confirm the save. The calibration values will now be active upon any future power cycles. Current values may be saved in the future at any time desired.

To return to the Home Page, press the ***Home PB*** on the left of the front panel.

WiFi Setup

In order for the scale to connect to the local WiFi network, it must first know the network's ID and password. This section describes the procedure to set up the scale for a local WiFi connection.

Upon initial power-up, or after a network reset, the scale enters Access Point mode. In this mode, it can be seen as a WiFi access point with a network name of "**ScaleWifiSetup**", and a network address of **192.168.4.1**. A computer or smart phone will need to connect to the scale using the scale's access point name – "**ScaleWifiSetup**". The following are the steps for Microsoft Windows 10:

1. Select the Network icon on the far right side of the taskbar. The scale should show up in the network list as "**ScaleWifiSetup**".
2. Select "**ScaleWifiSetup**",.
3. Click the "Connect" button.

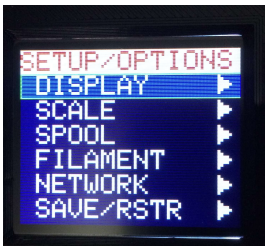
This should result in "**ScaleWifiSetup**" being connected, but with no internet access. This is expected.

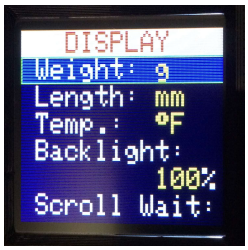
Once the scale is connected, open a browser, and enter "**192.168.4.1**" (without the quotes) into the browser address. This should bring up the page to the left. Click on the "**Configure WiFi**" button.

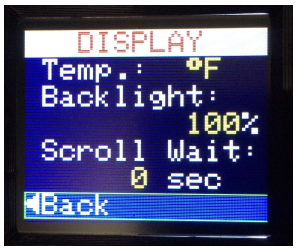
The page to the left will be displayed, containing a list of networks that the scale can see. Select one from the list, or enter a different one in the **SSID** field. Enter the password for the network then click on the "**Save**" button. This will cause the scale to reset. When it returns, it should be connected to the specified WiFi network.

Once the WiFi is connected, the scale may be found at "**JmcScale.local**". If the browser cannot find the scale at "**JmcScale.local**", then scroll the scale's Home Page display until the **IP Addr** field is displayed. Enter the address from the **IP Addr** field into the browser and something like the image below should be displayed. All of the scale's functionality is available from this screen.

Display Settings







This section describes the options available on the *SETUP/OPTIONS→DISPLAY>* page.

Weight Units

The **Weight** option allows for selection of the units that will be displayed for all weight values. This option uses the Cyclic Edit Mode. Successive clicks of the ***Encoder PB*** will cycle the selection through the following four values:

grams(g) → kilograms(kg) → ounces(oz) → pounds(lb) → grams(g)

Length Units

The **Length** option allows for selection of the units that will be displayed for all length values. This option uses the Cyclic Edit Mode. Successive clicks of the ***Encoder PB*** will cycle the selection through the following six values:

millimeters(mm) → centimeters(cm) → meters(m) → inches(in) → feet(ft) → yards(yd)

Temperature Units

The **Temp** options allows for selection of the units that will be displayed for temperature values. This option uses the Cyclic Edit Mode. Successive clicks of the ***Encoder PB*** will cycle the selection through the following 2 values:

Fahrenheit(°F) → Celsius(°C)

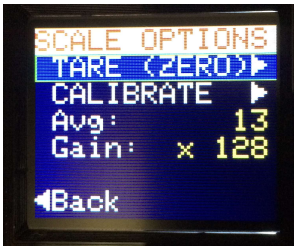
Display Brightness

The **Backlight** option allows for the setting of the display brightness. This option uses the Numeric Edit Mode. Values are indicated in percent, and range from 0%, which is completely dark, to 100% which is brightest.

Scroll Delay

The **Scroll Wait** option allows for the setting of the auto scrolling of data on the Home Page. This option uses the Numeric Edit Mode. Values are specified in seconds, and range from 0, which does not scroll the data to 120. A value of zero is a special case in which the Home Page data will only scroll via rotation of the **Encoder**. Smaller values (greater than zero) cause the values to scroll faster, while larger values cause the data to remain on the screen longer before being scrolled.

Scale Settings



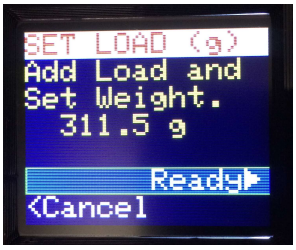
This section describes the options available on the *SETUP/OPTIONS→SCALE>* page. The scale settings handle

Tare (Zero)

The **TARE (ZERO)>** option allows the user to tare (zero) the scale. It is useful to do this after each power-up since the scale wanders a little over time. This operation can also be easily performed directly from the Home Page via a long press of the ***Units/Tare PB***. Simply follow the on-screen instructions to perform the tare.

Calibrate

The **CALIBRATE>** option allows the user to calibrate the scale. This must be done at least once before the scale will display any weight data. A proper calibration requires that a tare (zero) be done as well as loading the scale with a known weight. Thus this is a two stop process. Upon selecting the **CALIBRATE>** option, the user will be directed to do a TARE operation as described above.



Once the TARE operation completes, the user will be directed to add a load to the scale and set its weight. The weight is entered using the Numeric Edit Mode. Now, clicking on **Ready>** will complete the calibration process. Note that the calibration settings are not saved automatically. It is necessary to save the new settings via the *SETUP/OPTIONS→SAVE/RSTR→SAVE function (see below).*

Weight Display Averaging

The **Avg** option allows the user to determine the amount of smoothing that will be done when displaying weight values read from the load cell. **Avg** sets the number of readings that will be used for the running average of the weight values read from the load cell. The load cell is sampled 10 times per second (once per 100 milliseconds), so the value of **Avg** represents the number of tenths of a second that will be used to average the data. For example, the default value of 13 represents 1.3 seconds of averaging, which is a good starting value. This option uses the Numeric Edit Mode. Valid values range from 1 through 25.

Gain

The **Gain** option sets the gain of the load cell integrator. It should not normally be necessary to change this value from its default **x128** value. After changing this value, a **CALIBRATE** operation is required before any valid weight data can be displayed. This option uses the Cyclic Edit Mode. Successive clicks of the ***Encoder PB*** will cycle the selection through the following 2 values:

x128 → x64

Spool Settings

This section describes the options available on the *SETUP/OPTIONS→SPOOL>* page. The **SPOOL** page maintains a database of spools and their associated filament properties. A maximum of 15 spools may be contained within the database.

Spool Select

The **Select** option of the **SPOOL INFO** page determines whether or not the specific spool is selected (or active). When a spool is selected, the Home Page will display extra information pertaining to the spool and the filament it contains. Only one spool may be selected at a time. Upon selecting a spool, any other spool that may have previously been selected will be deselected. This option uses the Cyclic Edit Mode. Successive clicks of the ***Encoder PB*** will cycle the selection through the following 2 values:

No → Yes

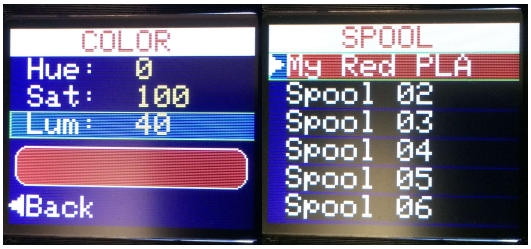
Spool ID

The **Spool ID** field sets the name for the spool. Names can be up to 12 characters in length. Initially, the **Spool ID** field is set to “Spool xx”, but may be changed as desired. Names do not need to be unique. For example, the database could contain 2 different spools with the **Spool ID** of “New Spool” without causing any problems. The **Spool ID** field uses the String Edit Mode.

Spool Weight

The **Wt** field specifies the empty weight of the spool. That is, it specifies the weight of the spool without any filament. This can easily be determined with brand new spools. Simply weigh the new spool and subtract the manufacturer’s filament weight from it. For example if a brand new spool weighs 1311.5 grams, and it is specified as containing 1Kg of filament, then the spool weight is 311.5 grams. Note that the **Wt** field is entered in the currently active weight units (g, kg, oz, lb). This option uses the Numeric Edit Mode. Valid values range from 0.0 grams through 5000.0 grams (or the equivalent weight in the current weight units).

Spool Color



Spool color is selectable via the *SETUP/OPTIONS→SPOOL>→Color>* selection. Color selection is via the HSL color model in which:

- H represents the HUE of the color (**Hue:**). Its value ranges from 0 (red) to 120 (green), 240 (blue) and wraps back to red at 360.
- S represents the SATURATION of the color (**Sat:**). This measures the intensity of the color. Its value ranges from 0 through 100 with 0 being completely unsaturated and 100 being completely saturated. For example, a saturation of 100 will produce a stronger red shade than a saturation of 50.
- L represents the LUMINOSITY or LIGHTNESS of the color (**Lum:**). This is a measure of how bright or dark the color is. Its value ranges from 0 through 100 with 0 being darkest, and 100 being the brightest.

All of these values use the Numeric Edit Mode.

Note that it is best to start with a **Sat:** value to 100 and the **Lum:** value to 50 when setting the color.

Spool Filament Type

The **Type** field specifies the filament type contained on the spool. There are 12 built-in filament types and 3 user definable filament types. The built-in filament types are: ABS; ASA; Copper; HIPS; Nylon; PETG; PLA; PMMA; PolyC; PVA; TPE; and TPU. The user definable types are displayed as USR1, USR2, and USR3. This option uses the Cyclic Edit Mode. Successive clicks of the ***Encoder PB*** will cycle the selection through the following 15 values:

ABS → ASA → Copper → HIPS → Nylon → PETG → PLA → PMMA → PolyC → PVA → TPE → TPU → USR1 → USR2 → USR3

When the filament **Type** is changed, the corresponding **Density** value will change to the default density for that filament type. If the default density for this spool is not acceptable, the **Density** field may be changed as described below.

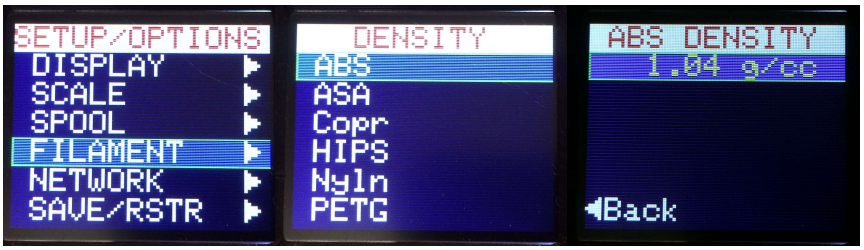
Spool Filament Density

The **Density** field specifies the density of the filament contained on the spool. Its units are always specified as g/cm^3, even if non-gram units are specified for display. This is due to the prevalence of manufacturers specifying these units. The default values for filament density should normally suffice in most circumstanced. However, special cases may exist. Hence the capability to change the density value is provided. This option uses the Numeric Edit Mode. Valid values range from 0.01 g/cm^3 through 5.00 g/cm^3.

Spool Filament Diameter

The **Dia** field specifies the diameter of the filament in millimeters (mm). Common diameters are 1.75mm and 3mm. However, manufacturing tolerances allow for deviation from these values. If extreme accuracy is desired, the filament diameter value may be modified to suit the particular spool. This option uses the Numeric Edit Mode. Valid values range from 1.00mm through 4.00mm.

Filament Settings



This section describes the setting of the default filament density values for the 15 supported filament types. It can be found at *SETUP/OPTIONS→FILAMENT>*.

Default Filament Density

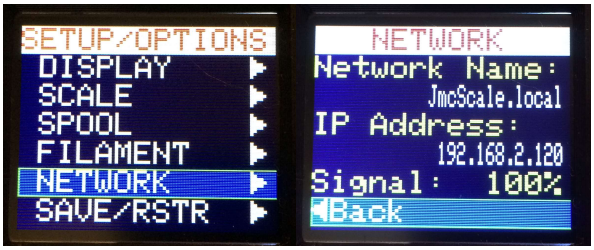
Every type of filament has a particular density. An online search will show that different vendors may have different densities for the same type of filament. However, in general filament densities vary only very slightly between vendors. The scale contains a database of average density values for its supported filament types given in the following table:

Table 8: Default Filament Densities

FILAMENT TYPE	DEFAULT DENSITY (g/cm^3)		FILAMENT TYPE	DEFAULT DENSITY (g/cm^3)
ABS	1.04		**PolyC**	1.20
ASA	1.07		**PVA**	1.19
Copper	3.90		**TPE**	1.20
HIPS	1.07		**TPU**	1.20
Nylon	1.08		**USER1 ***	1.24
PETG	1.27		**USER2 ***	1.24
PLA	1.24		**USER3 ***	1.24
PMMA	1.18			

Any time that the spool filament type is changed, its corresponding density value is updated the appropriate value from this database. This option uses the Numeric Edit Mode. Valid values range from 0.01 through 5.00.

Network Settings



There are no settable values on the *SETUP/OPTIONS→NETWORK*> page. This page only contains status information related to the network (WiFi) connection.

Network Name

The **Network Name** field shows the name that the scale presents to the WiFi network. This name is currently fixed as "JmcScale.local", but only if the WiFi connection has been setup. For many systems/browsers, entering this name into the search field will connect to the scale and display the scale's network page.

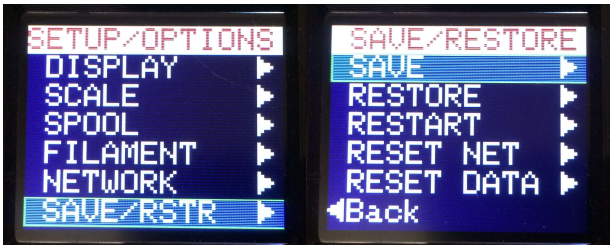
IP Address

The **IP Address** field shows the scale's IP address if connected to a WiFi network. The IP address may be entered into a browser's search field in order to connect to the scale.

Signal Strength

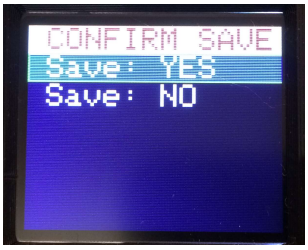
The **Signal** field displays the relative strength of the WiFi signal reaching the scale. Higher values represent better signal strength.

Save / Restore



The *SETUP/OPTIONS→SAVE/RSTR* page provides the ability to save and restore system settings as well as resetting system data to initial its conditions.

Save Current Settings

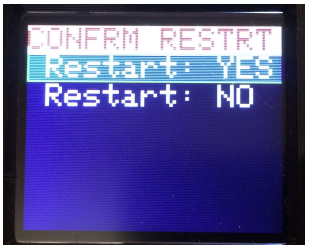


Selecting the *SAVE* option of the *SETUP/OPTIONS→SAVE/RSTR* page brings up a confirmation page. Choosing *YES* will cause all of the user changeable options/settings to be saved to non-volatile memory. Once the save is complete, any future power cycles will keep the current settings upon power-up. Choosing *NO* will abort the operation.

Restore Previously Saved Settings

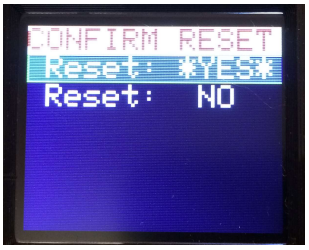
Selecting the *RESTORE* option of the *SETUP/OPTIONS→SAVE/RSTR* page brings up a confirmation page. Choosing *YES* will cause all of the settings that were active when the last *SAVE* was done to be restored. That is, any settings that have been changed since the last *SAVE* was performed will be lost. Choosing *NO* will abort the operation.

Restart Scale



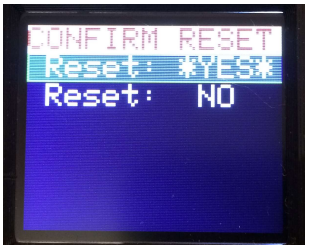
Selecting the *RESTART* option of the *SETUP/OPTIONS→SAVE/RSTR* page brings up a confirmation page. Choosing *YES* will cause the scale to restart, just as if the power had been cycled. Note that any unsaved setting changes will be lost. Choosing *NO* will abort the operation.

Reset Network



Selecting the *RESET NET* option of the *SETUP/OPTIONS→SAVE/RSTR* page brings up a confirmation page. Choosing **YES** will cause the network credentials to be lost and will require execution of a ***NETWORK SETUP*** procedure before the scale will be able to talk to the local WiFi network. This is usually necessary only when using the scale in a new environment that uses a different WiFi credentials. Note that this will also cause any unsaved settings that have changed since the last *SAVE* to be lost. Choosing *NO* will abort the operation, and keep the current WiFi settings.

Factory Reset (Data)



Selecting the *RESET DATA* option of the *SETUP/OPTIONS→SAVE/RSTR* page brings up a confirmation page. Choosing **YES** will cause all scale data that was previously saved to be reset to the initial factory settings. This should not normally be necessary. Be cautious using this. Choosing *NO* will abort the operation.

WiFi Display

The use of the WiFi page that is shown in a computer or cell phone browser when connected to the scale should be obvious. It follows normal internet conventions. Although the WiFi page may be used to monitor the scale and to change its settings, a few points must be kept in mind:

- The WiFi page and the console display are interlocked such that only one device may modify settings at a time. That is, while the WiFi page is editing any setting, any attempt to enter the

SETUP/OPTIONS page from the console will produce a page briefly displaying the text "**DISPLAY IN USE TRY LATER**". This indicates that settings are being modified by the WiFi connection. Similarly, if the *SETUP/OPTIONS* page is being accessed from the console, any attempt to enter any settings from the WiFi will result in a message on the browser stating "**Cannot modify XXX data. Options currently being modified.**".

- It is generally easier to change settings via the WiFi page than via the console. This is especially true when selecting the filament color since the WiFi page uses a visual color pallet.
- All settings apply to both the console display and the WiFi page. For example, the weight units apply to the display of weights on both the console and the WiFi page.
- Since the *RESTART*, *RESET NET*, AND *RESET DATA* commands cause the scale to perform a reset, the WiFi connection will be briefly lost after execution of these commands. If the WiFi connection is not restored within 30 seconds, then it may be necessary to refresh the browser.

POSSIBLE ENHANCEMENTS

As with most projects, lots of bells and whistles could be added to improve this scale. However, I have spent enough time on it and am ready to move on to something else. Some improvements that have come to mind include (in no particular order):

- **Bluetooth** – All modern smart phones contain Bluetooth, and the Huzzah supports it. Adding Bluetooth support would allow for easier connection to smart phones.
- **Modifiable Filament Types** – This version of the scale has hard coded the filament type names, along with supplying 3 user types named USR1, USR2, and USR3. These cannot be modified by the user. A nice addition would be to allow the user to rename any of the filament types to suit their need.
- **Audible Alert** – A fairly simple addition would be to add the capability to detect an empty spool and signal an audible warning. Similarly, a jam could be detected if the weight doesn't change for an extended period and an audible warning could be sounded.
- **Real Time Clock** – A WiFi connection is already supported, so it wouldn't take much effort to add a real time clock using NIST time. This would also require the addition of a timezone setting, and the address of the local NIST server.
- **Improved Web Page** - Many web page improvements are possible, such as allowing popups to be dragged, allowing page colors to be changed, support an audible alert as mentioned above, etc.
- **Temperature Compensation** – The scale drifts quite a bit due to temperature changes. I've seen a roughly 5 gram variation between 59° F and 69° F. This is why it is a good idea to perform a tare operation immediately after each power-up. With some experimentation, it should be possible to determine a compensation curve and implement it in the firmware.
- **Larger Filament Database –** The limit of 15 filament entries in the filament database is mainly due to a limitation of the ArduinoMenu library. It looks like, with some effort, it might be possible to extend the library to support more than 15 items. Fifteen entries was sufficient for my needs so I did not expend the effort to support more. (Thanks to mmally for this suggestion).
- **RFID Reader –** Mmally also suggested the use of an RFID reader to identify a spool and update the database automatically. This would require some work, but sounds like a great feature.