UROFLOW SYSTEM USING MRK1010 AND LOW-COST FOOD SCALE

Abstract

This document describes the construction, validation, use, and design of a uroflow measurement system based on a low-cost Arduino MKR1010 electronics platform and an inexpensive food weigh scale. This system measures peak urinary flow rate that might be helpful in tracking the benefit of drug therapies for treating urinary obstruction associated with benign prostatic hyperplasia. The system can transfer the time-stamped data that is stored on a uSD data card to a PC through a wifi connection, and an associated Java program plots time-stamped flow-vs-time records and a plot of weekly histograms of peak urinary flow. Accuracy is maintained by checking the final electronic volume measurement against visual volume measurements, and by providing an independent check of the time base.

This file is available at https://github.com/jerrygeo/uflow, Double-click uroflow.pdf

Figure 1. Uroflow measurement hardware

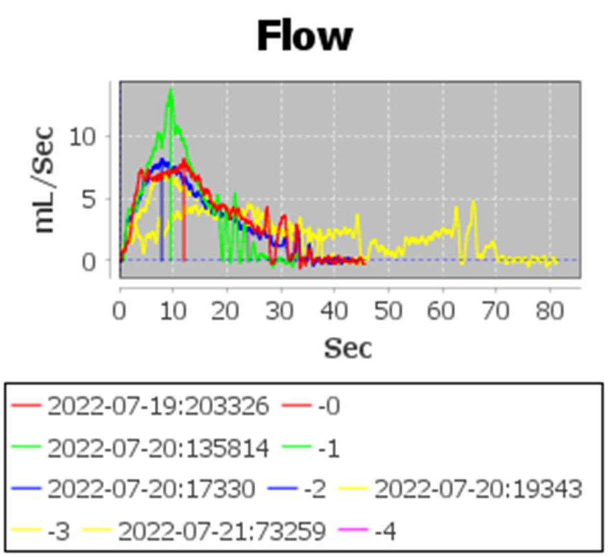


Figure 2. Plot showing flow vs time, with vertical lines for automatically detected peak

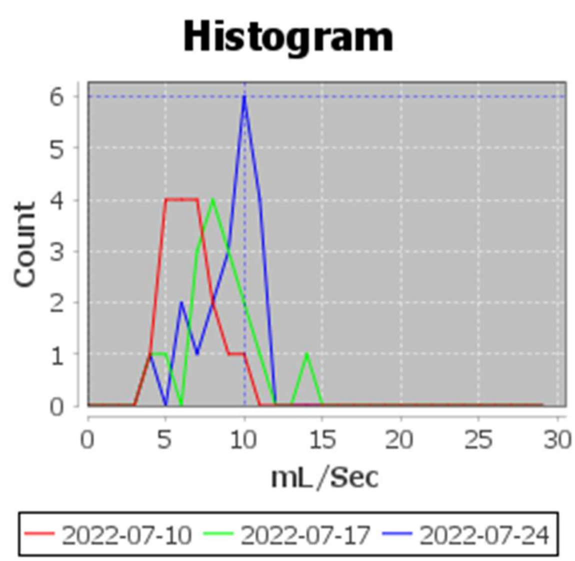


Figure 3. Plot showing weekly histograms flow of Qmax (peak flow rate)

1. Introduction

Benign prostatic hyperplasia (BPH), that is, enlargement of the prostate gland, occurs in a majority of older men and can obstruct the urethra, causing difficulty in urinating. One way of quantitatively evaluating the progression of this condition and assessing the benefit of the various available treatments is by measuring the maximum flow rate (Qmax) during urination. Although this measurement can be performed in a urologist's office, the results can vary widely, and multiple measurements at home can provide a more reliable result [*Home and office uroflowmetry for evaluation of LUTS from benign prostatic enlargement, D. Porru* et al*, Prostate Cancer and Prostatic Diseases (2005) 8, 45-49*]. Commercial devices that perform this measurement are available, but purchasing these systems or renting them for extended periods can be costly. The system described below, based on a low-cost food scale and an Arduino Mkr1010 processor board, can be constructed for about $200. A symptom survey (such as the IPSS score) is probably a better indication of treatment efficacy, but the histogram graphs of the Qmax measurement might be helpful in providing confidence that gradual changes in symptoms are moving in the right direction. If the histogram graph shifts to the right with time, the condition is improving.

Other battery-operated MKR1010 projects may be able to use the housing (4.53"L x 3.54"W x 2.17"H), PC board, color LCD (or other LCD with an SPI interface), and 18650 lithium battery. The LCD could be mounted on the inside of the (currently unused) top of the clear plastic enclosure. The PCB schematic and layout are included. These were designed using the free version of Diptrace, so modifications to the PCB using alternate peripheral devices could be implemented relatively easily.

Figure 4. Mkr1010 system in enclosure

Table of Contents

2. Materials and Equipment

Equipment Required:

Soldering iron, solder
Drill w/ 3/16”, 3/8”, 5/16”, 20mm bits (or use a smaller bit and jig saw for 20mm)
Jig saw
Small Phillips screwdriver
Hot melt glue gun
Crimping tool, such as Sparkfun TOL-13193 ($32.50)
Heat gun
Dremel/Moto tool with small grinding cylinder
Wire cutters
Wire stripper
Pliers

Helpful for debugging

Multimeter
Oscilloscope
EZ hook test leads

BILL OF MATERIALS

Qty	**Description**	**Vendor**	**Approx. Cost Each**
1	Coil cell battery – 12mm CR1225 (PRT-00337)	Sparkfun	$1.95
1	Plastic Box 4.53”L x 3.54”W x 2.17”H (377-1542-ND, HB2723)	Digikey	$14.60
1	HitLego (10pcs $7.07) 18650 Battery Holder	Amazon	$0.71
1	18650 3.7V rechargeable battery (4 pack for $25)	Amazon	$7.00
1	CapitolBrand CBV-V440941-E1 Polypropylene 250mL Molded Graduated Beaker (alternate needed – no longer available)	Amazon	$10.00
1	Winco PF-32 plastic funnel, 6.25-inch diameter	Amazon	$6.84
1	Ozeri ZK-14S food scale	Amazon	$9.84
4 ea	Screw, pan head, M4 x 3/8” (10mm)		
2 ea	Screw, pan head, M3 x 5mm		
2 ea	Screw, pan head, nylon 6-32 x ¼”		
4 ea	Nut, nylon, 6-32		

PC Assembly

1	U1	Arduino MKR WiFi 1010 (ABX00023)	Digikey	$34.90
1	U1	Arduino MKR MEM Shield (ASX00008)	Digikey	$19.50
1	--	3247-USDCOEM-32GB-ND 32GByte micro SD card	Digikey	$9.17
1	U2	DeadOn RTC Breakout DS3234 (BOB-10160)	Sparkfun	$22.50
1	J2	Sullins SBH11-PBPC-D05-ST-BK 10 position vertical header S9169-ND	Digikey	.42
1	J2 (alternate)	FCI 69176-010LF 609-2382-ND CONN RECPT HSG 10POS 2.54mm	Digikey	$2.53
2	J3,J4	Amphenol FCI 87800-111 Friction Latch header POL 4 POS	Mouser 649-87800-111	$.73
1	R1	Resistor, 10K ohm, 1/6W, 1% Yageo MFR-12-52-10K	Mouser	.1
1	R2	Resistor, 15K ohm, 1/4W, 1% Yageo MFR-25FTE52-15K	Mouser	.1
1	R3	Resistor, 40K ohm, 1/6W, 1% Yageo MFR-12FTF52-40K	Mouser	.1
1	R4	Resistor, 6.65K ohm, 1/4W, 1% Yageo MFR-25FBF52-6K65	Mouser	.1
1	R5	Resistor, 2.1K ohm, 1/4W, 1% Yageo MFR-25FRF52-2K1	Mouser	.1
4	U1,U3,J1	Sparkfun PRT-16279 1568-PRT-16279-ND Aduino Nano Stackable Headers	Digikey	$1.50
2	U3	Breakaway vertical header,Harwin M20-9994046 (used as spacer and for pins of HX711 module	Digikey	1.64
1	U3	TE Connectivity 2-825437-5 1x25 MODII Hdr, Unshr, .1 Right angle header (mount on HX711 module)	Mouser	3.78.
1	Printed circuit board fabricated per ScaleMkr1010_gerberx2.zip	See section 3.3		$25

HARNESSES			
26	FCI 649-47213-000LF 47213-000LF 28-32AWG crimp pin, Tin	Mouser	$.205
A/R	Hookup wire AWG 28 Solid		
	LCD HARNESS		
1 ea	Amphenol ICC (FCI) 78211-008LF, Conn Rcpt HSG 8POS 2.54mm , Digikey 609-2400-ND		1.60
1 ea	Amphenol FCI 69176-010LF 2x5 Pos housing		2.59
	PUSH-BUTTON SWITCH HARNESSES		
3 ea	FCI 74211-004LF, Conn, Rcpt HSG 4POS 2.54mm		1.23
2 ea	Cylewet 1A 250V AC SPST Momentary Mini Push Button Switch (pack of 12 $6.21)	Amazon	$0.51
	BATTERY/POWER SWITCH HARNESS		
1 ea	JST Jumper 2 wire assembly (JST PH connector for battery) Sparkfun PRT-09914		$1.05
1	Rocker switch, COM-11138	Sparkfun	$0.55
1	HitLego (10pcs $7.07) 18650 Battery Holder	Amazon	$0.71

3. Construction

3.1. Download project files from Github

At https://www.github.com/jerrygeo/uFlow click Code | Download Zip.
Unzip the files to a folder, such as Documents/uflow.

3.2. Download Arduino IDE

See https://www.arduino.cc/en/software.

3.3. Order PCB

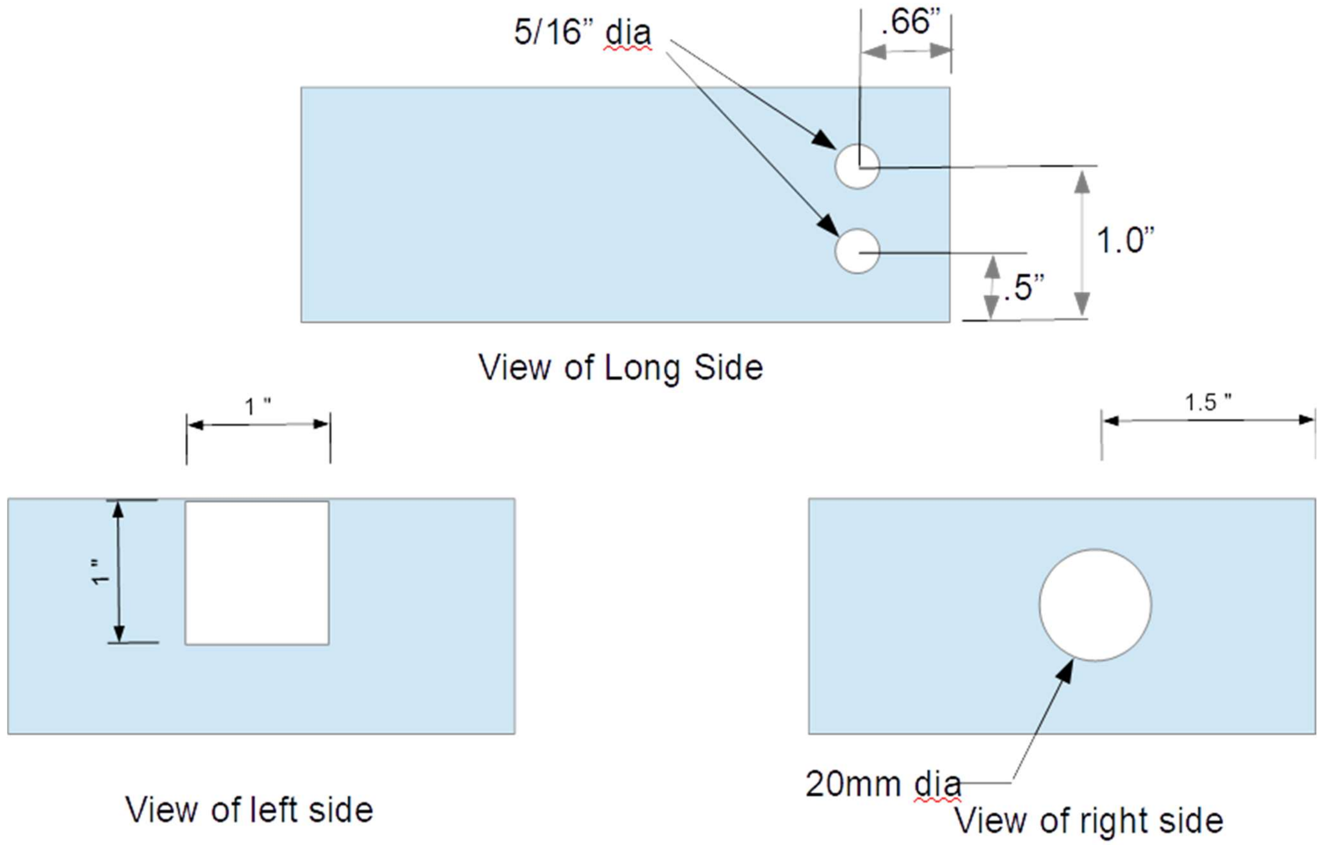
The zip file containing relevant files for the PCB is ScaleMkr1010_gerberx2.zip. This package can be sent to a PCB fabrication vendor. I used a Chinese company and got good results at a low cost. Another site recommends JLC PCB, Seeedstudio, AllPCB, or OSH Park.

The PC board was designed used DipTrace. The design files are included in the Github directory.

3.4. Modify plastic enclosure

Only the bottom section of the enclosure is used.
Drill holes for 2 push button switches using 5/16" bit . Currently, only one of these switches is used, for the Bluetooth connection to download the data to a computer. The other switch is a spare. Also, drill (or cut with a jig saw) the hole for the 20mm power switch, and cut the slot for accessing the USB cable and uSD memory card.

The hole and slot locations are indicated in Figure 2:



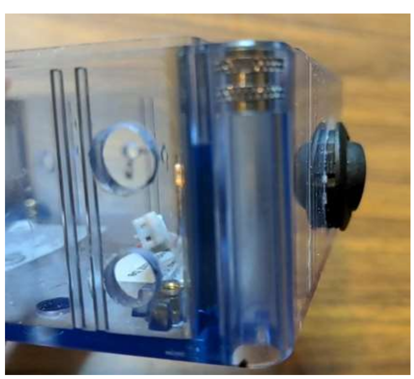


Figure 5. Chassis Machining on HB2723 Enclosure Body

3.5. Disassemble weigh scale

1. Remove the circular top scale platform by rotating it counter-clockwise, then pulling it off.

2. Remove the rubber feet on the back of the scale to reveal the 4 small Philips head screws, and remove these screws, placing them in a secure location to avoid losing them. Remove a 5th screw accessed from the top as shown in Figure 7 below. There is a 6th screw that is accessed from the top, by partially lifting the clear plastic cover over the top overlay, then lifting a small section of the black overlay using an Xacto knife. Alternatively, the boss for this screw can be drilled out from the front side as can be seen in the photo. The scale enclosure can be adequately reassembled without using this 6th screw.

Figure 6. Access to top screw on scale cover

3. Remove the 2 screws securing the mounting bracket for the circular top platform.

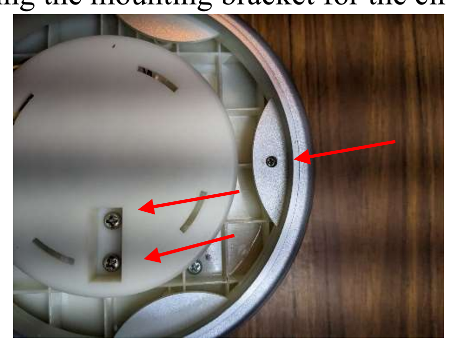


Figure 7. Screws securing mounting bracket for circular platform and 5th screw securing enclosure top to bottom

4. Drill the holes for mounting the electronics enclosure to the base of the scale using 3/16” bit (see below). I used the captive screws in the unused top of the HB2723 enclosure to mark hole positions on the rear of the scale base. The photo below shows 2 of these screws – the other 2 are under the front section of the scale’s top cover.

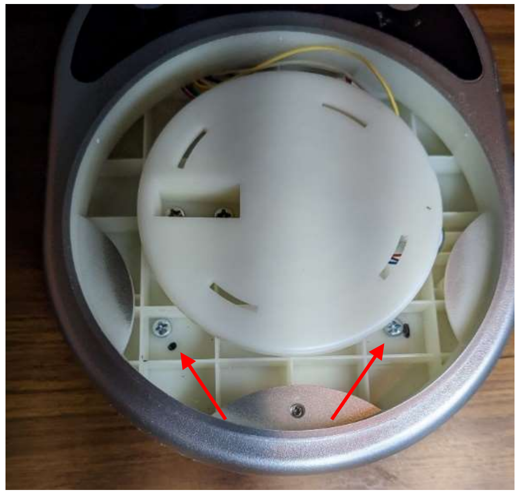
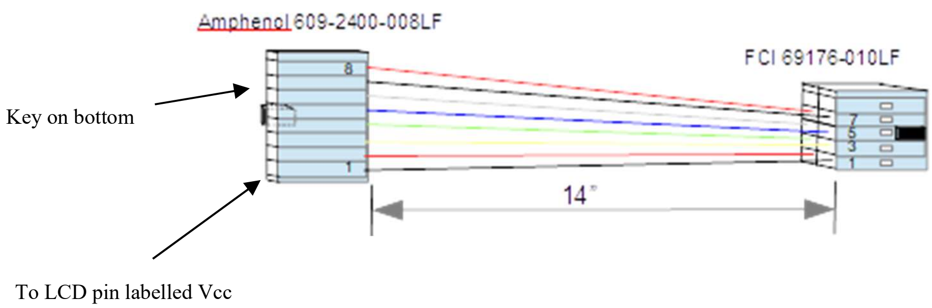


Figure 8. Holes for screws attaching electronics housing to scale

3.6. Assemble wiring harnesses

Crimp pins (FCI 47213-000LF) must be attached to the wires to the FCI Berg housings (LCD harness and push-button switch harnesses). Strip insulation from the end of the wire to expose about 1/8" bare wire, then crimp the pins to the wires. Performing this crimping and getting the wires into the proper locations requires care and patience.

1. LCD harness



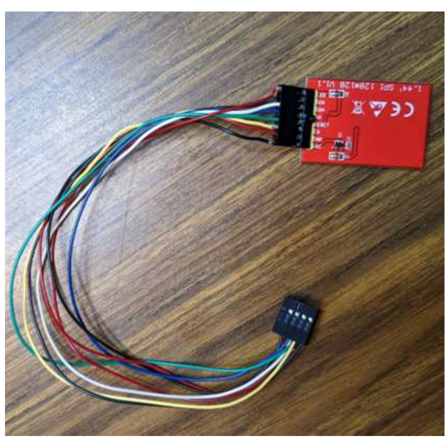


Figure 9. LCD Harness

2. Battery Harness

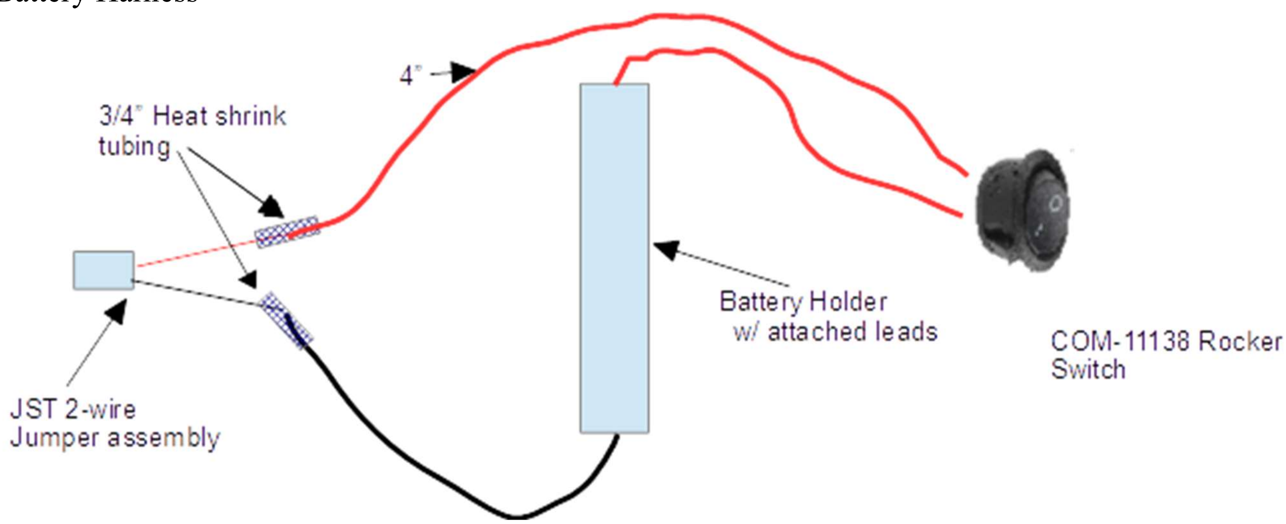
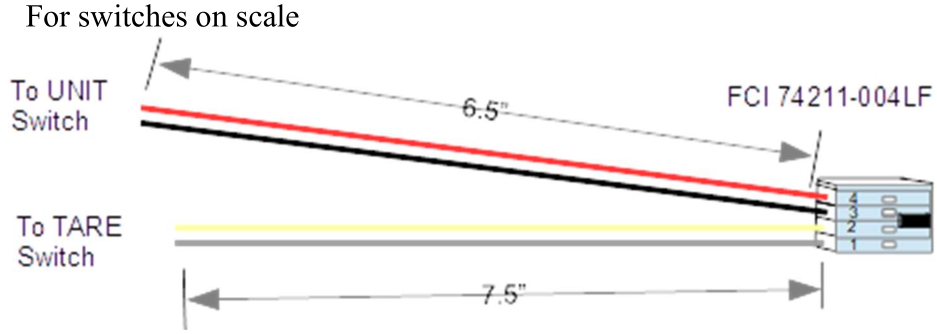


Figure 10. Battery harness

Do not solder the red wires to the rocker switch until the switch has been mounted on the enclosure.

3. Push-button switch harnesses

For switches on scale



The UNIT and TARE switches are on the top panel of the scale, as will be discussed below.

For enclosure-mounted push-button switches

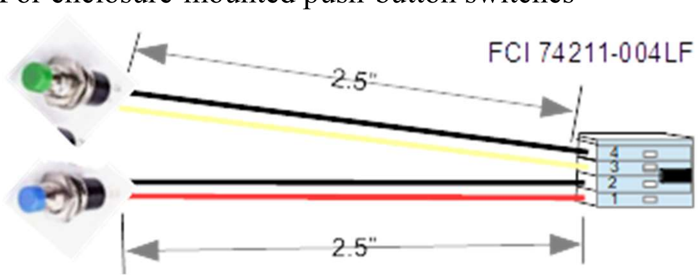


Figure 11. Push-button switch harnesses

Only the blue switch is currently used, and this is only used to initiate wifi data transfer. This wifi connection turns out to be fairly slow (~20kbps) and the direct USB connection is much faster, so this switch harness is optional; however, the wifi connection is convenient, and the uxfer.jar program simplifies transferring data over wifi.

3.7. Install components on the PCB

Mount the components according to the reference designators indicated in the bill of materials above. To make the connectors for J1 and U3, cut sections of stackable headers (PRT-16279) to the appropriate number of pins using a diagonal plier wire cutter.

J1 has 2 rows of pins to allow using the circuit board for other MKR1010 applications with jumper wires to the unused pins. Attach the connector in the outermost row (with the square pad). Install the connector at an angle to ensure that the RTC module will clear the HX711 module, as shown below:

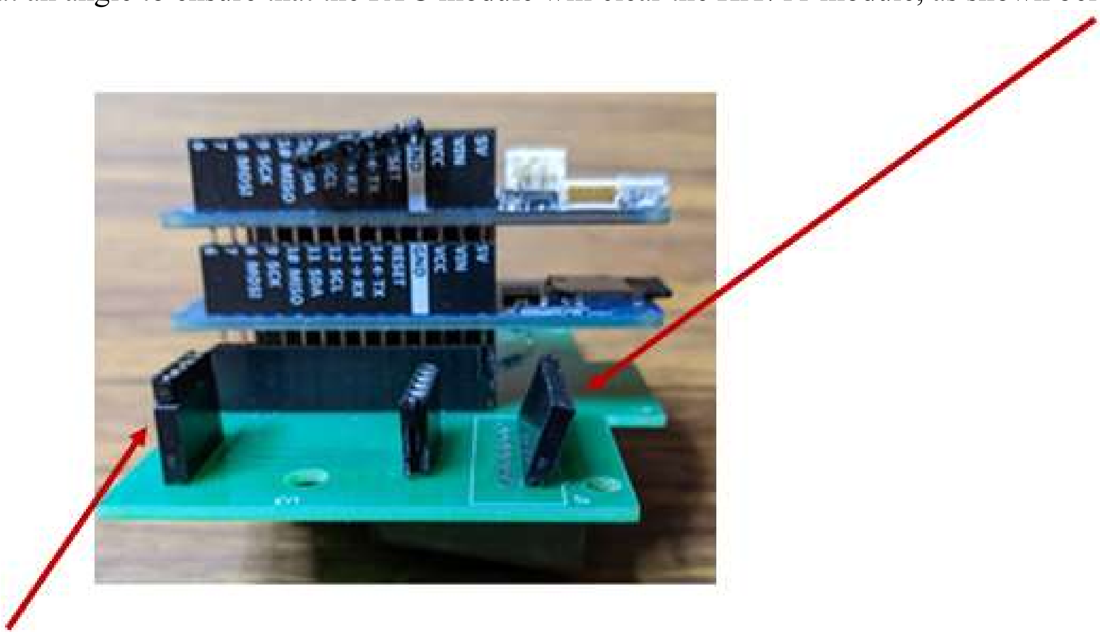


Figure 12. Connector mounting on the PCB

The 6 pin header on U3 does not connect to any pins—it is only used to stabilize the HX711 module. An extra row of straight breakaway header pins is added for additional spacing to match the header mounted on the bottom side of the HX711 module.

Mount straight header pins on HX711 module as shown below, connecting only to the pads E+,E-, A+, A-. Leave the pads labeled B+ and B- open.

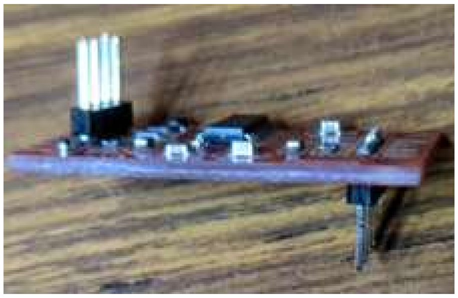


Figure 13. Connectors for HX711 module

Mount right angle header pins (2-825437-5) onto the DS3234 RTC module:

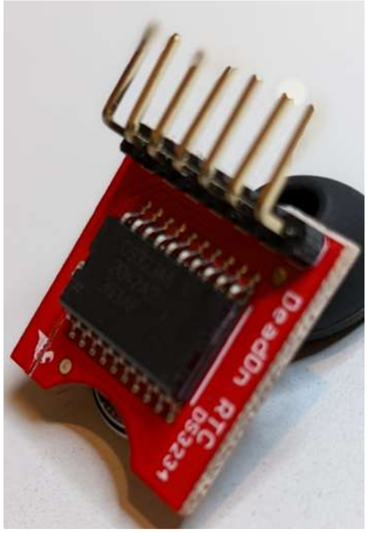


Figure 14. Connector for DS3234 module

Install the CR1225 coin cell battery onto the DS3234 RTC module

3.8. Mount modules into the PC assembly

Stack the MKR Mem Shield and the MKR 1010 Wifi modules, then install them into the sockets at U1.

Install the uSD card into the MKR Mem Shield board. NOTE: The SD card receptable on this board is not rugged and can be ruined by inserting the SD card with too much force, as I found out. This board does not seem suitable for removing and inserting the SD card very often.

Install the DS3234 module into the socket at J1 with the battery side of the board facing the center of the enclosure, and install the HX711 module at J3.

The completed PC subassembly is shown below:

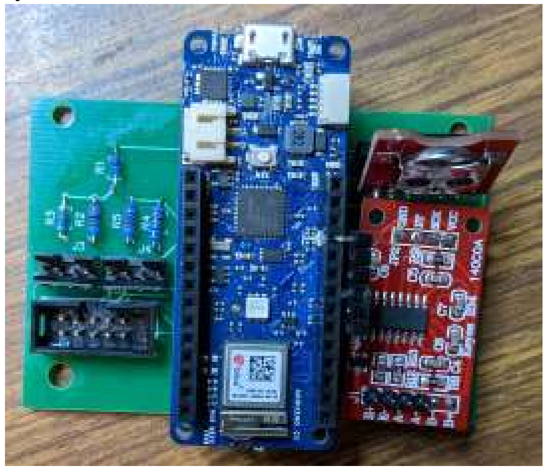


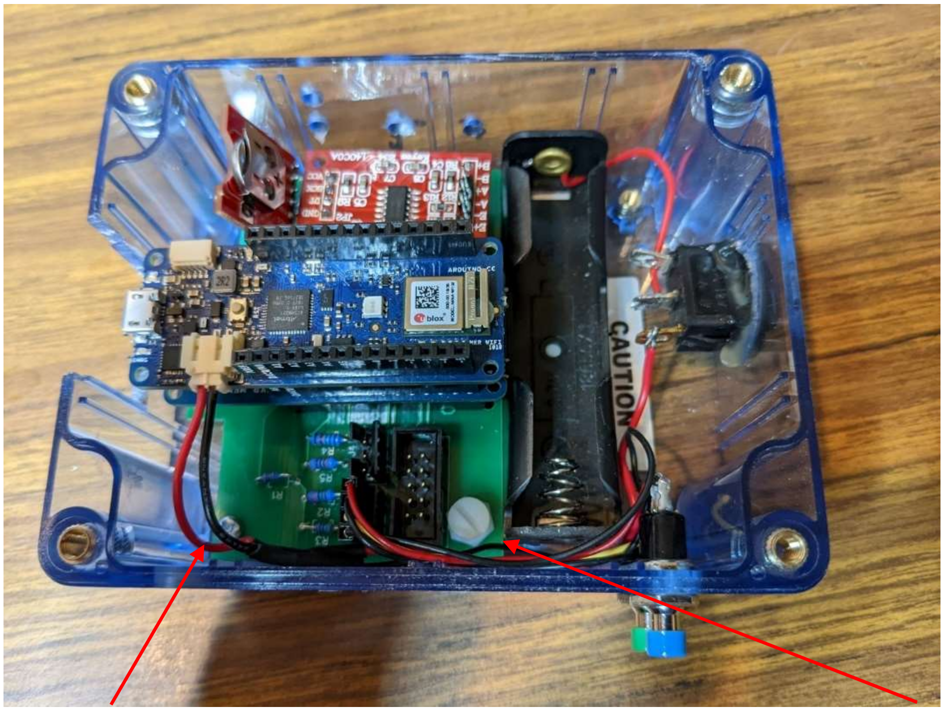
Figure 15. Completed PCB subassembly

3.9. Mount subassemblies onto the enclosure

1. Press the rocker switch into the enclosure and secure it with hot melt glue around the inside.
2. Solder the wires from the battery harness to the rocker switch, then secure the battery holder to the enclosure with hot melt glue.

Figure 16. Enclosure with battery and power switch mounted

3. Install the push button switches into the enclosure using pliers or a 10mm socket wrench.
4. Attach the JST battery connector to the MKR 1010, and plug the push button connector into J3.
5. Attach 6-32 x ¼" nylon screws to the bottom end of the PC board using the nylon nuts. These screws just act as spacers for the PC board to match the height of the bosses at the corners of the enclosure.
6. Attach the PC assembly to the enclosure using the 2 M3 screws.



M3 screw Figure 17. Completed electronics assembly Nylon screw

3.10. Rework scale cover assembly

The LCD that comes with the scale is secured with a PC board that is screwed into the top cover. Remove the screws holding this PC board and remove the LCD. Unsolder the wires that connect to the scale's strain gauge. There are 2 small PC boards that hold the switches for the TARE and UNIT buttons. Unsolder the wires connected to these boards.

There is a lip around the edge of the LCD hole that secures the LCD. Using a Dremel tool, diagonal pliers, or other means, remove a section from the center of the bottom of this lip as shown below to allow the HiLetgo LCD to fit in this space. Cut two 1" x 5/8" rectangles of black construction paper to mask off the sections of the display window that will not be occupied by the HiLetgo LCD and mount them in the LCD cavity (I used sections of a black dense foam sheet).

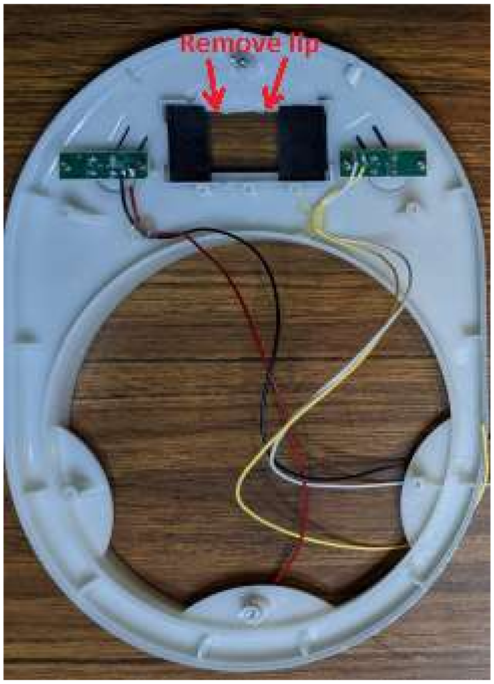


Figure 18. Modification to scale top cover and scale switch connections

Install the LCD in the orientation shown below, and secure it with hot melt glue around the edges. Cover the back of the PC board that was originally in the scale with an insulating material (such as electrical tape or Kapton tape) and replace the PC board to help hold the LCD in place.

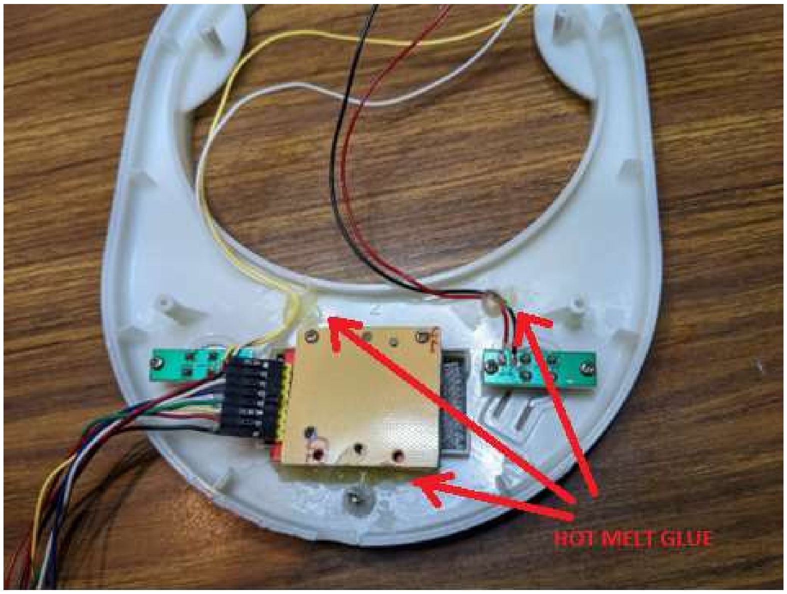


Figure 19. LCD installation and hot melt glue application

Solder the wires from the top enclosure push-button switch harness onto the switch PC boards as shown above. Apply small amounts of hot melt glue to provide some strain relief.

Crimp Berg connector pins (FCI 649-47213-000LF) onto the 4 wires from the scale's strain gauge and insert them into a 4 pin housing (FCI 74211-004LF) as shown below:

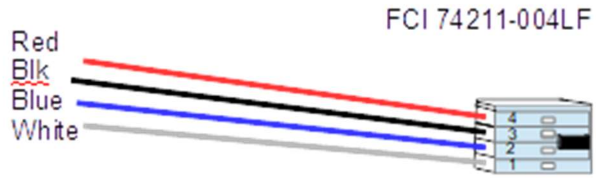


Figure 20. Scale strain gauge connector

3.11. Final assembly

Drill two 3/8" holes in the bottom section of the scale housing and file these together to form a slot to allow the wires to pass through, as shown below:

Figure 21. Slot on scale bottom for connections to electronics

After checking that it is charged, install the 18650 lithium battery into the battery holder in the electronics enclosure with the + side near the red wire. The battery can be recharged through the USB port if the power switch is turned on.

With the top supported on its side, attach the connector from the top cover switches to J4, and attach the connector from the strain gauge to the HX711 with the red on E+ and the white on A-

.

Figure 22. Attaching scale wires to electronics.

Position the scale on top of the electronics enclosure, line up the enclosure holes with the threaded inserts, and install the 4 M4 x 10mm screws to secure the scale to the electronics enclosure.

Finally, replace the 4 screws holding the scale enclosure's top to its bottom, and re-attach the circular top scale platform, rotating it clockwise after seating it into the slots in the plastic bracket.

3.12. Install code and test the electronic system

Connect a USB cable between the PC that has the Arduino IDE and the MKR 1010. The USB connector is somewhat difficult to install since the orientation is not obvious. On my cable, there is a small circular indent with the USB logo – this faces the bottom.

Launch the Arduino IDE and load the Scalemkr1010 project. Make sure that the processor in *Tools* | *Board*: is selected to be Arduino MKR Wifi 1010.

Install the following libraries using Tools | Manage Libraries:

- RTC by Makuna version 2.3.5 (GNU Lesser General Public License v3.0)
- Adafruit GFX Library by Adafruit version 1.10.10 (BSD License)
- Cooperative Multitasking by Andreas Motzik Version 1.0.11 (License – ?)
- WifiNina by Arduino version 1.8.12 (GNU Lesser General Public License v2.1 or later)
- HX711 version 0.1 (from https://github.com/bogde/HX711. Download the zip file and follow these instructions: https://docs.arduino.cc/software/ide-v1/tutorials/installing-libraries) (MIT License)

The following library, installed as part of the IDE, is also used:

- SD (GPL-3.0 license)

I needed to modify one of the library functions to allow the DS3234 code to compile since the MKR1010 includes an RTC. The DS3234 has its own coin cell long-life battery, avoiding the need for figuring out a low power mode for the MKR1010—the MKR1010 is completely turned off with the power switch. The DS3234 also provides an independent check on the clock rate. The file samd21g18a.h was in my directory as follows:

C:\Users\jerry\AppData\Local\Arduino15\packages\arduino\tools\CMSIS-Atmel\1.2.0\CMSIS\Device\ATMEL\samd21\include\samd21g18a.h

I needed to comment out line 265:
#include "component/rtc.h"

Select *Sketch* | *Verify/Compile* to confirm that the code builds. Since the Arduino platform keeps evolving, effort may be needed for this step.

Modify the file Arduino_secrets.h to enter the SSID and password for your wifi router. The SSID is the name of the wifi as it shows on your cell phone. Then re-build with *Sketch* | *Verify/Compile.*

With the USB cable connected between the PC and MKR1010, select *Sketch* | *Upload* to load the code onto the processor. After a short delay, the LCD should display the version number and the message Press TARE. If this does not display, it's likely that there is a problem with the LCD cable harness.

You can confirm that the code is running (even if the display is not showing) by selecting *Tools* | *Serial Monitor* then typing "h' to get a help message showing the available terminal commands. Ensure that the Serial Monitor window indicates Newline for the character sent when Enter is typed:

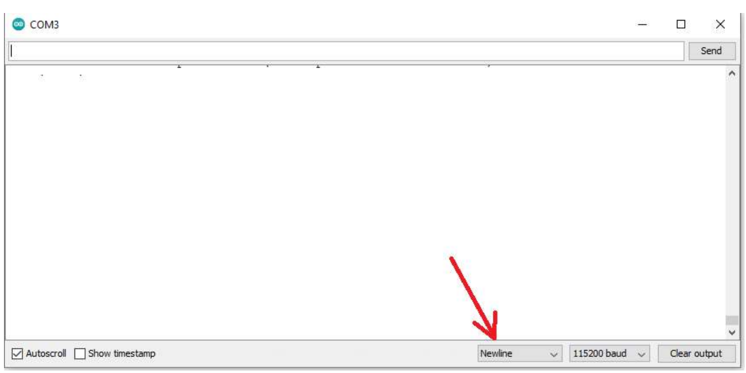


Figure 23. Configuration for Serial Monitor in Arduino IDE

The buttons all connect to a single analog port, with different analog values corresponding to different buttons. The buttons can be tested using the BUTTONS command. This command displays the table used to identify the various analog port button values, then for 15 seconds it displays the analog value read and the corresponding button every ½ second. This command can be helpful in identifying problems with the button wiring harnesses.

4. Calibration and Validation

4.1. Calibration

The system generates a default calibration file on the uSD memory when it first starts, but this calibration constant needs to be tuned for each unit.

1. Start with the system off with the USB cable disconnected. It may be easier to disconnect the cable from the PC than from the MKR1010. Fill the graduated beaker with 100mL of water, then transfer the water into another container. Place the empty graduated beaker on the scale.
2. Attach the USB cable and allow the system to power up.
3. From the Serial Monitor of the Arduino IDE, enter the command to set the time, using the format TIME Month dd yyyy hh mm ss

   Where: Month names are JanFebMarAprMayJunJulAugSepOctNovDec
   dd - Day of month (range 1-31)
   yyyy – Year
   hh – Hour (range 0 – 23)
   mm – Minute (range 0-59)
   ss – Second (range 0-59)

   Example :

   TIME Jan 04 2022 14 08 56
4. Enter the calibration mode by issuing the command RECAL. The message “Recalibrate” should appear in the serial monitor window.
5. Press and release the TARE button.
6. Press and release the UNIT button. The message ADD 100mL THEN PRESS TARE should appear in the LCD.
7. Pour the 100mL of water into the graduated beaker.
8. Press and release the TARE button.
9. Disconnect the USB cable.

4.2. Validation

1. Find a means of obtaining a stream of water with uniform flow that can be easily redirected. I used a kitchen faucet with a retractable head. Using the graduated beaker, adjust the flow so that the beaker is filled to 150mL in about 15 – 20 seconds. Adjusting the flow may require some patience.
2. Place the empty graduated beaker on the scale with the funnel resting on top and turn on the power switch.
3. Press and release the TARE button when instructed to do so by the LCD.
4. Press and release the UNIT button when instructed to do so by the LCD.
5. Prepare a timer set for 20 seconds, or use a second hand on a clock.
6. Direct the flow into the beaker and simultaneously start timing.
7. After 20 seconds, quickly direct the flow away from the beaker.
8. Press and release the TARE button.
9. The LCD will display the measured volume and Qmax. Make notes of these values.

   Qmax from LCD: ________ Volume from LCD: _________
10. Inspect the fluid level in the beaker to confirm that the volume reported on the LCD is within 5mL of the visually measured value.

    Volume of fluid measured with graduated beaker: _________
11. Press and hold the TARE button continuously until the LCD reports DONE. If the TARE button is released too early, the measured volume value will increase.
12. Turn off the unit.
13. Connect the USB cable to a PC running the Arduino IDE and activate the serial monitor.
14. Type the command “X U<Enter>” to send the flow data to the PC. The data is in records of the form: Time Volume Corrected Volume Flow
15. Scroll back through the data to find the first time at which the volume was greater than 5mL and record this time.

    Start time: ___________
16. Scroll forward to find the time at which the volume stopped increasing. Record this time and the corresponding volume:

    End time: ___________ Final Volume: ____________
17. Confirm that the value (End time – Start time) is within the range 19 – 21 seconds.

    End time – Start time: _____________
18. Compute the average flow rate based on the visual measurement using the graduated beaker by dividing the total measured volume by 20 seconds.

    Average flow based on visual measurements: __________
19. Compute the average flow rate measured by the system by dividing the Final Volume by the value (End time – Start time), and confirm that this value is within the range of .9 – 1.1 times the average flow based on visual measurements (10%).

    Average flow from system = Final Volume / (End time – Start time): ____________
20. Confirm that the value of Qmax reported on the LCD is within 10% of the average flow. Qmax will likely be slightly higher than the average flow due to the fluctuations in the flow.
21. Use the DELETE command to clear out the test data.

5. Instructions for Use

5.1. Sample collection

The LCD guides the user through the operations required.

1. Place the unit on the closed toilet seat lid with the graduated beaker and funnel on the scale.
2. Turn on the power. The following message appears on the LCD:

3. Press and release the TARE button, then press and release the UNIT button per the following message:

4. Then the following message appears:

5. When finished filling, press and release the TARE button. The LCD will show the measured volume:

6. Check the volume visually in the graduated beaker. The displayed volume can be adjusted up by pressing and releasing the TARE button and down by pressing and releasing the UNIT button. When the volume agrees with the visual measurement, press AND HOLD the TARE button. It is important to hold the TARE button firmly down to avoid changing the displayed volume. After the button is held down for several seconds, the final measurements are displayed, including:

    Qmax in mL/sec
    Total volume in mL
    Time from the start of flow to the peak flow, and total time

7. Dump the sample into the toilet and rinse the funnel/graduated beaker into the toilet. Wipe the scale down. Periodically wipe/rinse with isopropyl alcohol to suppress odors.
8. Battery charging: With the power switch turned on, connect a USB cable that is attached to a power source. The system draws about 50mA (I think), so, with a 3500 mAh battery, the battery should last ~70 hours, or 840 five-minute sessions.

5.2. PC Software

5.2.1. File transfer

1. Turn on the unit and wait for the start-up screen.
2. Press the blue button for about one second and release it.
3. Wait for the LCD screen to indicate the IP address assigned to the unit, for example:

4. Run the Java program uxfer.jar. Running this program requires Java to be installed on the PC. Use the command prompt (Windows key+r then cmd). Navigate to the directory that holds this program (for example, cd \Uses\<yourUsername>\ Documents\uflow) and use the command:

   java -jar uxfer.jar
5. The program will request the IP address. Enter the address showing on the LCD screen.
6. The program should begin transferring both the UFLOW.CSV and SUMMARY.CSV files, assigning file names U<date>.csv and S<date>.csv, where <date> is the current date.
7. Alternatively, the USB cable can be connected and a terminal program such as TeraTerm or HyperTerm can be used to transfer the data, using the commands "X U" and "X S".

5.2.2. Data display

1. On the PC, run the program ushow.jar.
2. Select the desired Uxxxxxx.CSV file. Be careful to select a .CSV file with a name starting with U and not choose one starting with S (for Summary type of file).
3. The program will generate plots of the flow data with 5 traces per plot, indicating the maximum flow with vertical lines (see Figure 2 above). If the visual measurement of volume differs from the scale measurement by more than 5mL, the flow data is re-scaled to correspond to the visual measurement. These plots might be useful for confirming that Qmax is being appropriately chosen, and that there are no artifacts in the data.

   The program also plots weekly histograms of the maximum flow values, showing the number of records having each value of Qmax (see Figure 3 above). This plot is the primary output of the system, allowing assessment of the change in flow over time.

   Any of the plots can be saved as a PNG file by right clicking on them.

6. System Design

The block diagram of the system is shown in Figure 24 below:

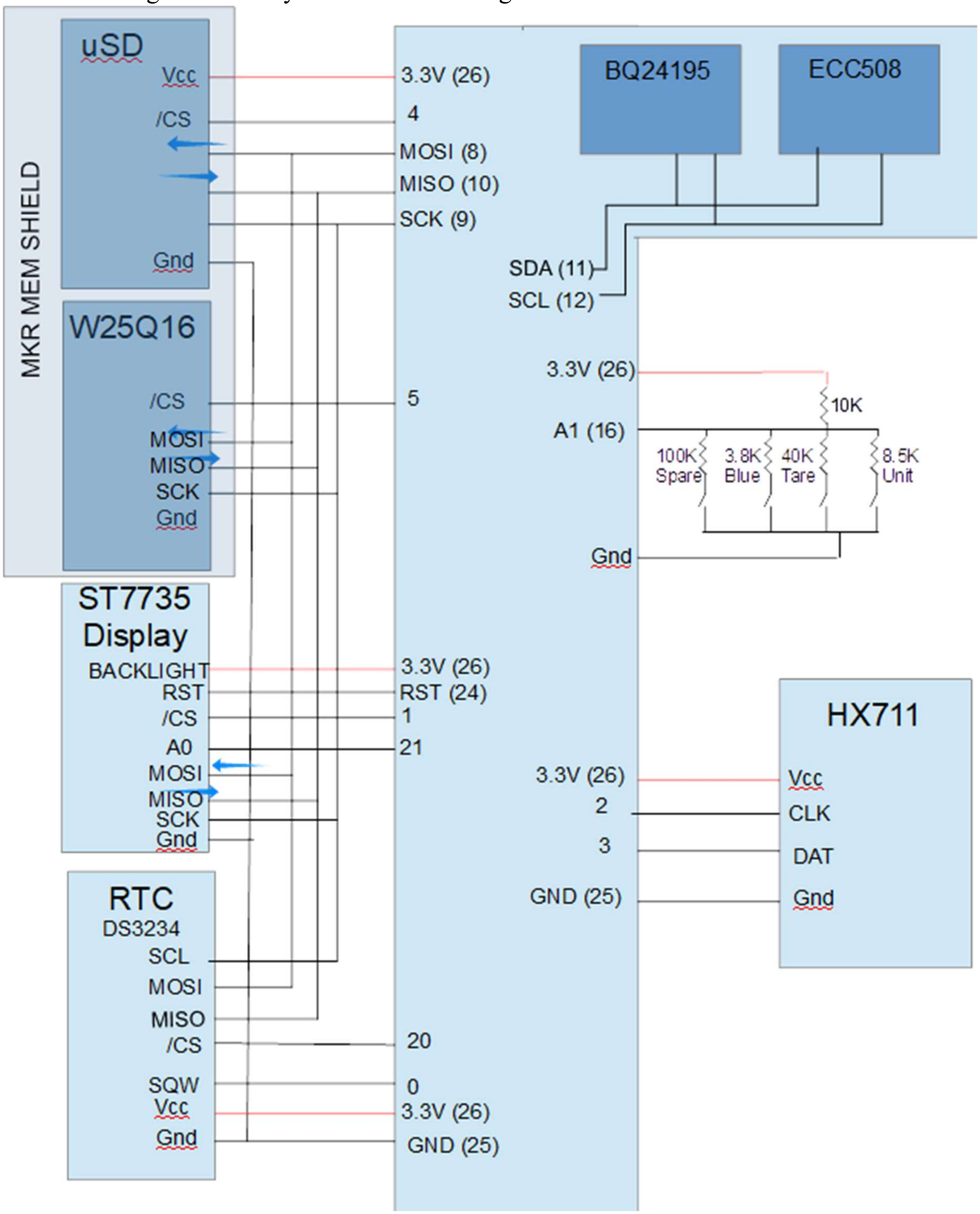


Figure 24. System Block Diagram.

The schematic and layout were developed using DipTrace. The schematic and layout files are on Github.
The electrical schematic is shown in Figure 25. The Mem Shield is not shown since it mounts directly with the MKR WIFI 1010.

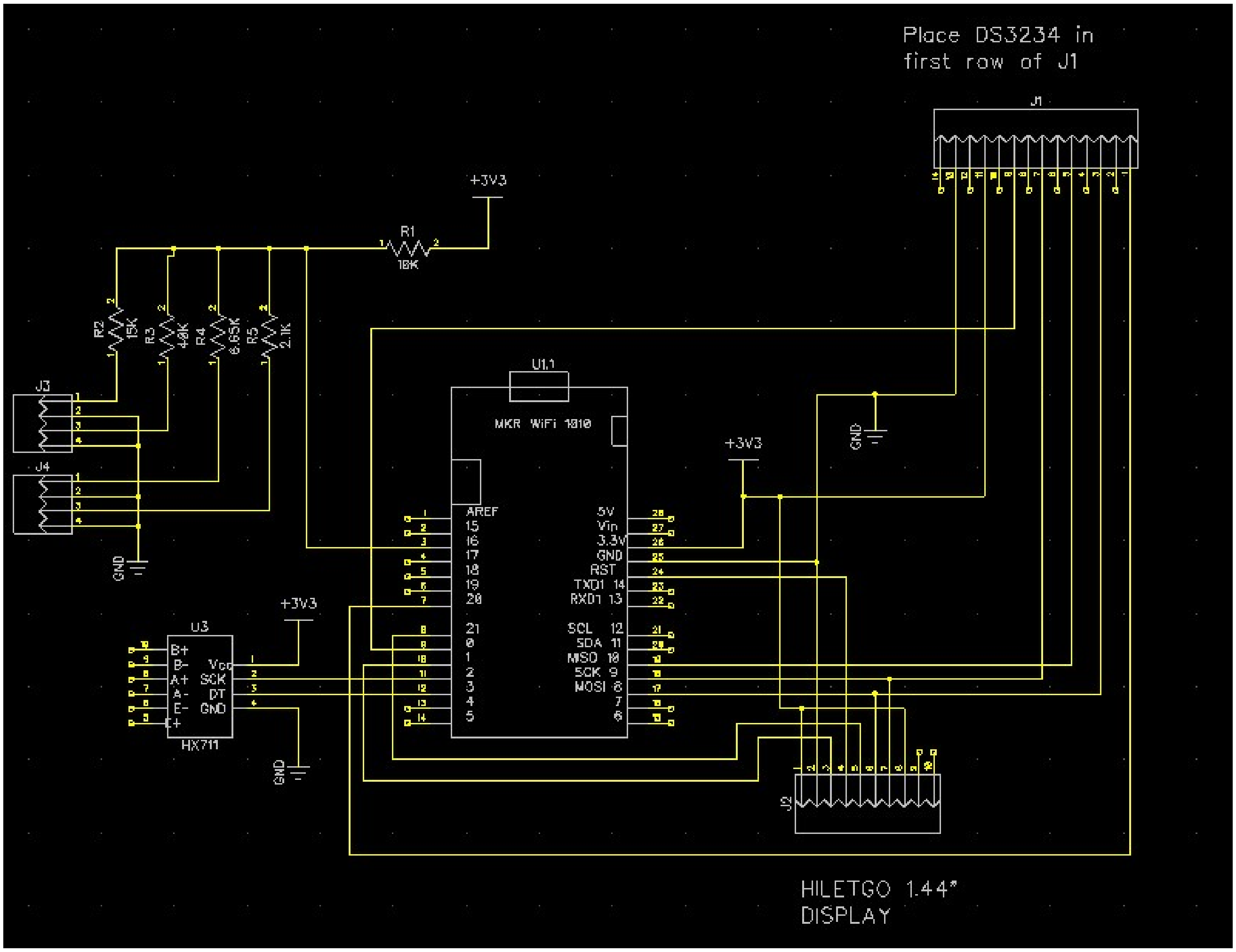


Figure 25. Electrical Schematic

The PCB layout is shown in Figure 26 below. There are fewer than 30 connections, so it would be practical (but rather tedious) to mount the components on perf board and hand-wire the connections with AWG 30 wire.

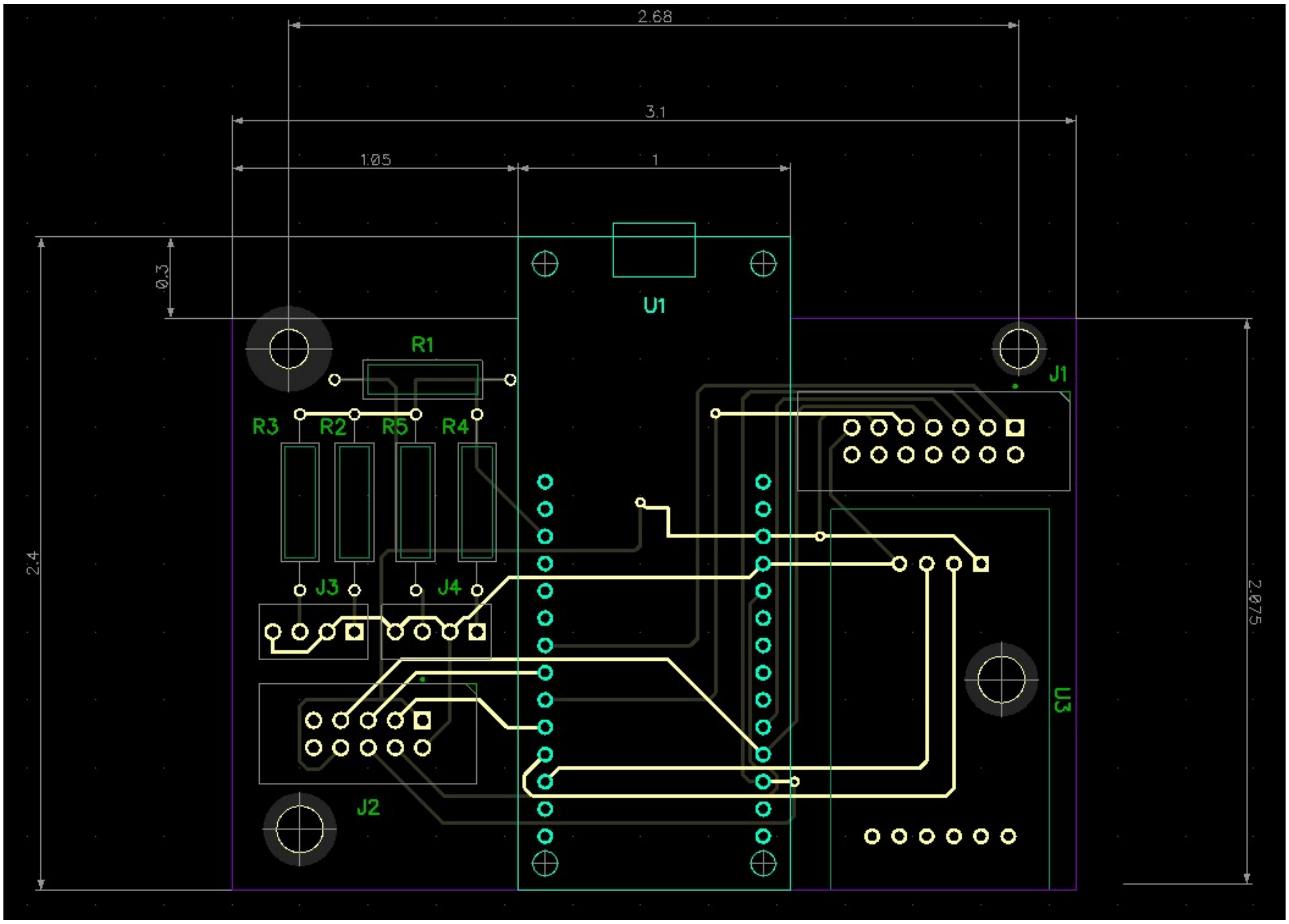


Figure 26. PCB Layout.

7. Software Description

The software includes a program for the MKR1010 wifi platform and Java programs on a PC to collect the data and to display it. The MKR 1010 software was developed using the Arduino IDE, version 1.8.14. The Java programs were developed using Jetbrains IntelliJIDEA 2022.2 Community Edition.

7.1. Arduino MKR1010 software

7.1.1. Tasks

The MKR1010 software uses Cooperative Multitasking to separate the code into 3 tasks:

- MainStateTask – handles the data acquisition, LCD messages, and downloading data,
- ButtonTask – checks for button press events and length of time that a button is pressed,
- SerialTask – handles commands from the USB port.

The MainStateTask runs every 100msec.
The ButtonTask runs every 20msec.
The SerialTask runs every 60msec.

The Arduino setup() function performs the following operations:

- Set pin 19 (A4) to output to allow checking the acquisition timing.
- Intialize the serial (USB) port for 115200 baud, waiting for 1 second to detect that the USB is connected, and declaring serialConnected false if it is not connected.
- Attempt to communicate with the wifi module. If this fails, set an error flag. (This just checks that the module is functioning – it is not checking for a wifi connection.)
- Check that the firmware is compatible with the wifi driver code. If this fails, set an error flag.
- Initalize the SPI port
- Initialize the LCD
- Check that the uSD card is present. If not, set an error flag.
- Initialize the DS3234 RTC. If the RTC date is not valid or earlier than the compile date, set the RTC date to the compile date. If the RTC is not running, start it. Set the square wave output of the RTC to 8.192 kHz, but don't enable it yet.
- Using a rising edge interrupt on the RTC square wave input pin, count rising edges for 500 msec and confirm that the proper number of edges has been seen, within 0.25%. If this separate RTC time base does not agree with the 1msec clock on the MKR1010, set an error flag – the system will stop later with a message on the LCD. Turn off the square wave from the RTC module.
- Read the configuration file to get the scale calibration constant. If no file is found, create this file.

The MainStateTask and ButtonTask each use state machines (implemented with switch statements) as shown below:

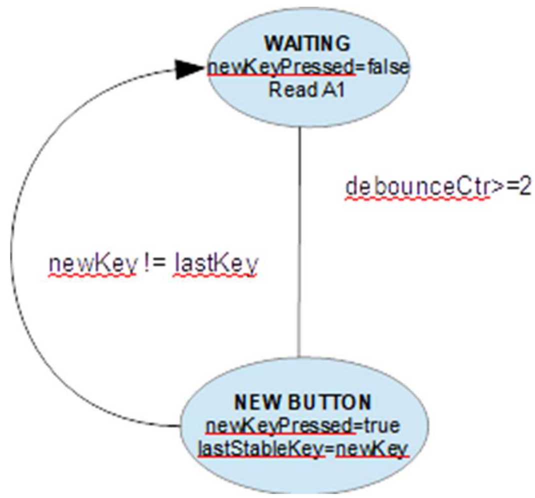


Figure 27. Button task state machine

Button presses usually cause actions when the button is released. In the VOLUME state, the TARE button must be held down continuously to transition to the PROCESS state.

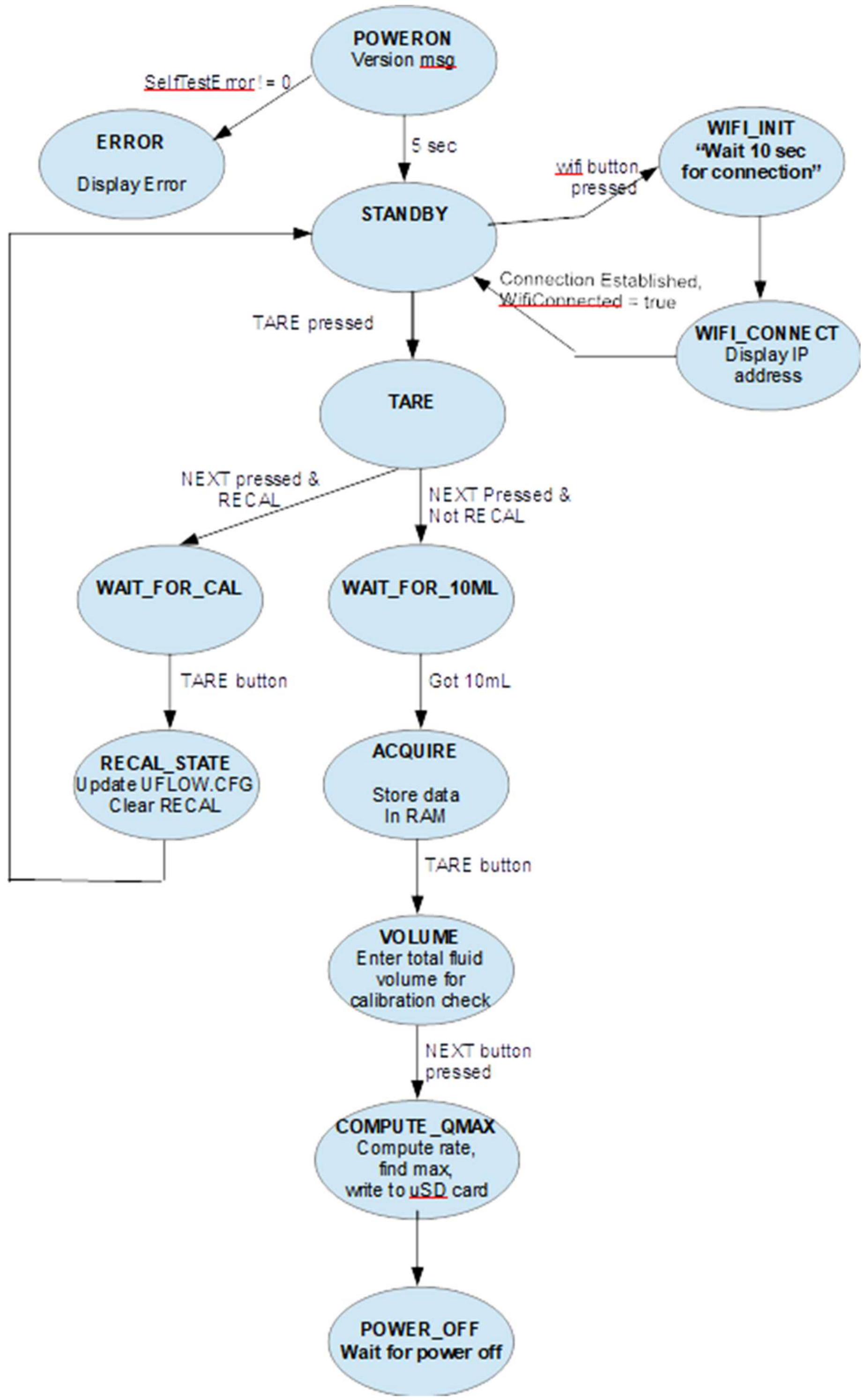


Figure 28. MainStateTask State Machine

The MainStateTask issues messages on the LCD screen guiding the user through the operations to acquire data or perform a recalibration operation. After displaying the splash screen with the version message, it checks if error flags are set – if so, it enters the Error state. After performing a tare operation, it then begins storing data collected at 100msec intervals into RAM. Data is stored in a 2-second FIFO until 10mL is detected, then data fills a RAM buffer (up to a maximum of 60 seconds). In the VOLUME state (entered when the TARE button is pressed after filling), the user checks the volume measured by the system against volume showing on the graduated beaker, and the displayed volume measurement can be adjusted up or down in 5mL increments by pressing and releasing the TARE or UNIT buttons, respectively. Finally, when the proper volume shows on the LCD, the user presses *and holds* the TARE button to find Qmax and write the data to the uSD card.

From the initial state after power-on, if the blue button is pressed and released to enter the WIFI_CONNECT state, the system attempts to connect to the wifi identified in the Arduino_secrets.h file (Sect. 3.12). When the connection is established, the IP address is displayed on the LCD, allowing a PC program to connect to the device.

The SerialTask checks for commands from the USB (serial) port or from wifi if it is connected. Output is delivered to the serial port and to the wifi port if it is connected. Commands include:

- TIME - Set time on RTC (time data is entered on the command line)
- RECAL – set a flag that causes the MainStateTask to follow a recalibration sequence, in which a tare operation is performed, the user is instructed to add 100mL of water, then the calibration constant is computed and stored in the file UFLOW.CFG on the uSD drive.
- X – sends the data file UFLOW.CSV (with command X U) or SUMMARY.CSV (with command X S). The UFLOW.CSV file has all the volume and flow data as well as the computed values such as Qmax, and the SUMMARY.CSV file has only the computed values. The SUMMARY.CSV file might be useful for computing other statistics on the measurement data using Excel.
- DELETE – causes the files UFLOW.CSV and SUMMARY.CSV to be deleted.
- BUTTONS – allows testing the operation of the buttons by displaying the analog value measured on the button port, and the button corresponding to this value.
- H – displays a help message showing the commands available.

The MKR 1010 USB port connects to the PC as a serial port at 115200 baud. The Serial Monitor of the Arduino IDE can be used for this communication, or some other terminal program can be used, such as TeraTerm or HyperTerm.

7.1.2. Algorithms

1. **Acquisition**: Collect volume (weight) data at 100msec intervals into 2 second FIFO until 10mL is measured. Then continue collecting data into a RAM buffer up to a maximum of 6000 samples. Volume is stored in units of 0.1mL. Stop collecting when TARE is pressed.

2. **Processing**
   Outlier Elimination
   The scale measurement reports wildly incorrect measurements on rare occasions (e.g., 15 times in about 23,200 samples). To avoid problems from these outliers, the volume data is first corrected to remove outliers, which are defined as volume measurements that are smaller than the previous measurement by more than 2mL or greater than 10mL above the previous measurement. These limits are extended for additional points. When a sample within the expected limits is found, the intervening samples are interpolated. The original volume data for these outliers is saved in a list, and the original is output to the Uxxxxx.csv file in addition to the filtered data.

   I suspect that these occasional bad data points may be due to the fact that the HX711 ADC clock is not synchronized to the processor clock, so the processor might be reading the ADC when new conversion results are being transferred to the HX711 output register. If this is the problem, it could possibly be fixed properly by re-writing the HX711 driver to use interrupts to read the data immediately after the DT line goes low. I didn't want to get into this, especially since the DT line is also used for data transfer.

   Flow Calculation
   The flow is estimated by computing the derivative of the volume data. Fluid volume data is stored as unsigned integers with a units of .1mL. If the derivative were computed from $(x_i - x_{i-1})/\Delta T$, the flow rate would suffer a maximum quantization error of +/-0.05mL/.1sec = +/-0.5mL/sec. So, this estimate of the derivative is averaged over 4 samples:
   Flow = ¼*(x[i-1]-x[i-2] + x[i]-x[i-1] + x[i+1]-x[i] + x[i+2]-x[i-1])/ΔT
   = ¼ * (x[i+2] – x[i-2]) /ΔT
   This estimate reduces the maximum quantization error to ±0.125mL/sec. However, there were periodic fluctuations in the derivative measurements with a period of 6 – 7 samples. I suspect that these fluctuations are due to waves on the fluid surface. So, the above estimate is averaged over 7 samples, resulting in the computation

$$
\begin{aligned}
Flow_n = (\, & Vol_{n+5} - Vol_{n-2} \\
& - Vol_{n+4} - Vol_{n-3} \\
& - Vol_{n+3} - Vol_{n-4} \\
& - Vol_{n+2} - Vol_{n-5} \,) \cdot Qscale
\end{aligned}
$$

   Where Qscale = (1/10)*(1000/(100*4*7), which accounts for the volume being stored in units of 0.1mL, and the 100msec data rate.

   Start Time
   The start time is defined as the time at which the volume first exceeds 2mL.

Qmax Calculation
The maximum flow rate is searched over the data from the start time to the end time (the time at which the flow no longer exceeds 2mL/sec. However, to disqualify short bursts at the end of the session, a peak is disqualified if the flow drops below 1mL/sec within 1 second of the time of the peak.

Time Calculations
The Total time is defined as the time between the start time and the time at which the flow is no longer greater than 2mL/sec.
The Tpeak time is the time between the start time and the time of Qmax.

7.1.3. File Structures

Configuration file UFLOW.CFG

All entries are in ASCII format with Name field, tab, then value

Field Description	Name	Format	Example
1	Software version	VERSION %4.2f	2.01
2	Scale Calibration	CALFACTOR %d	4900
3	16 bit checksum	CHECKSUM %d	43327

Summary file SUMMARY.CSV

Record Structure (in ASCII, all on one line for each session)

"DATE", Date (in format mm/dd/yyyy),
"TIME", Time (in format hh,mm,ss),
"Qmax", Max flow rate in mL/sec, floating pt, after adjusting total volume
"Tpeak", time in seconds from 10mL level to peak flow, floating pt.
"Ttotal", time in seconds from 10mL level to flow less than 2mL/sec, floating pt.
"Volscale", volume in mL as determined by the scale, before adjusting for the value observed on the graduated beaker
“Volvisual”, volume observed on the graduated beaker
“Calib”, scale calibration constant used

Flow data file UROFLOW.CSV

For each session, the file includes

The data above in the Summary file.
Records of one line for each measurement (100msec intervals) as follows.
There are no name fields in these records – just the data

Time in seconds
Corrected volume (with outliers removed) in mL
Volume in mL
Flow in mL/sec