

# Release Notes for CPAP FlowLite Software

---

## Version 2.15.6

- UART2 configured for SD card logging
- New support for external RTC part with battery backup.
  - External RTC addressed at 0x68 (7-bit address) on the I2C bus; accessing the external RTC during operation will use the I2C bus semaphore to prevent collisions with other I2C devices
  - On bootup, the system will compare the external RTC time to the time stored in FLASH and use the later time as new MCU time
  - External RTC time is accessible with command “time”, along with MCU time:  

```
Command: time
MCU Time is 12:55:27, date: 5/19/2022 (1652964927)
External RTC Time is 12:55:26, date: 5/19/2022
```
- Bug in Circuit Calibration limits: if the snorkel is mistakenly disconnected, the calibration passed because the `c_leak` was calculated to be  $> 0$  (`c_leak` was approximately 0.0001 with snorkel disconnected).
  - New `c_leak` minimum limit is: `#define C_FACTOR_GLOBAL_MINIMUM_LIMIT (0.0008f)`
  - Now a disconnected snorkel will cause the “CIRCUIT TEST” to fail until the snorkel is properly connected
- Send “popup\_screens\_present” info to the host (FlowWorks) so we can simulate popup screens
- Improve data logging:
  - Never log data that starts with “senddata” or “sendstatus”, which are meant for FlowWorks interaction
  - Remove trailing `<CR>` character (`'\r'`) from buffers sent to the datalogger, because it causes the datalogger file to double-space all the data

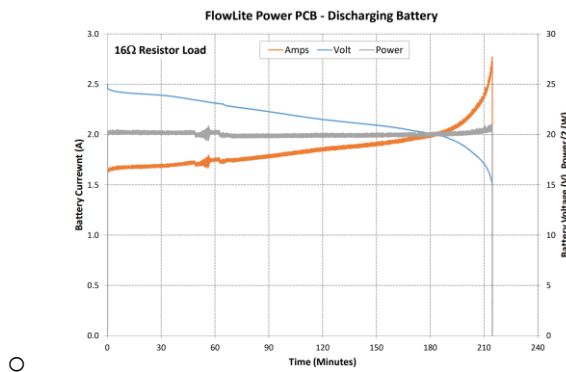
## Version 2.15.5

- Configure UART2, pins PB16 and PB17, to be used on SD Card for data logging.
  - Output all diag text (same as USB port) onto the data logger, plus do the “defaultStream” onto the data logger (running at 115200)
  - Remove the “PWRGD” signal and the “FP\_CS” signal to make room for the new UART; “FP\_NCS” is now assigned to PA-5.

- Add a “button(10)” to simulate pushing the Power Button on the new front panel – the FlowWorks tool will need this, to simulate the Power button.

## Version 2.15.4

- Shutdown system if battery voltage is below 19V for 10 seconds while AC is disconnected. At 20V, we put the system in “standby”, meaning the blower runs at low speed and everything else is shutoff (no O2 or pressure or heat is maintained when the battery is this low). Then when dropping below 19V we do a full shutdown using the new Power Board shutdown.



- (SPI Master baud rate changed from 200000 to 400000 – should have stayed at 400000)

## Version 2.15.3

- Supporting Screens for LCD (no setpoints displayed, no buttons displayed, updated alarm screens). Move the baby temperature and FiO2 and Pressure to proper position in “new” alarm screens.
- Occlusion alarm: if flow\_Insp is < 1.8, start Occlusion Alarm (low flow is associated with water blocking the circuit tubes)
- If there are I2C errors when reading flow, set the flow value to “-200” to force an error upstream (otherwise the flow just reads back 0); also will cause the POST test to fail if there are I2C errors
- Fix printouts in “CIRCUIT TEST” calibration to be useable in Software Verification tests
- Create a “forcePropValve” function to artificially create “High FiO2” or “Low FiO2” alarm conditions by forcing the prop valve to certain values, overriding the PID control loop

## Version 2.15.2

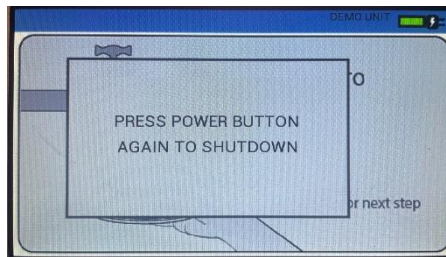
- Supporting New Power Board
  - Use the condition “both “nChg” and “nFull” are high when “NoAC” is low then there’s a problem with the battery” (from Chris) to create a “BATTERY ERROR” condition in the POST, and a new Device Failure alarm for the battery

- Improvements to heat PID control
  - Transition to  $t_{dist}$  control after 2 minutes, not 3, and transition instantly if  $t_{dist}$  is already at the 37C setpoint (fixes issue with quickly restarting therapy where temp is already at setpoint)
  - KP for heat plate PID increased from 200 to 600 to speed up response when control falls below setpoint
  - KI for heat wire increased from 1 to 3 (fixes issue with heat wire being very slow to reach 39C, hovering around 37-38C for a long time)
  - Add PID Integral limit negative – we already have a limit on how large the integral term can get in the positive direction. Since a negative signal has no meaning for a heater, we add the corresponding negative integral term limit, so when trying to cool down the heater, we don't build up a giant negative integral term.
- New Device Failure features:
  - Battery failure Alarm
  - Oxycell missing communications Alarm
  - New “Device Failure” alarm screen with failure description and “Next Steps” screens using canned text in “screens.c”
- Software verification
  - New command “forceBlower(x)” allows user to temporarily override the blower setting to force a “High Pressure” or “Low Pressure” alarm for testing
  - Command “blower” now prints out blower speed and ALSO “Blower On Time” for viewing

## Version 2.15.1

- Starting at version 2.15.x, software uses the Front Panel board and the new Power Board
  - New digital inputs from Power Board, read through I2C bus:
    - AC Unplugged – used to determine if AC connected
    - Power Button – monitored to decide to shutdown FlowLite
    - Battery Charging (not used)
    - Battery Fully Charged (not used)
  - Power Button Behavior:

- If Power Button is pressed and released, a pop-up screen appears and tells the user “PRESS POWER BUTTON AGAIN TO SHUTDOWN”

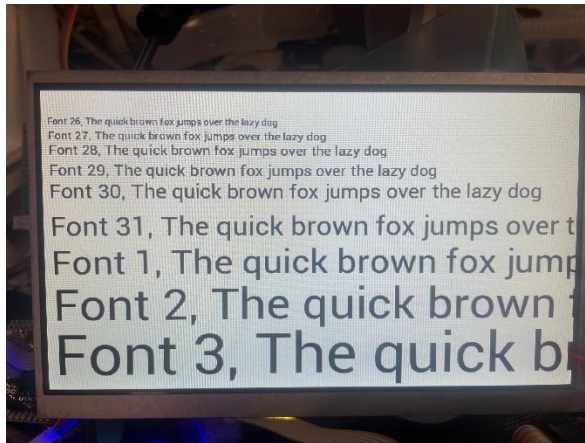


- If Power Button is pressed again, the software will power down the system within 2 seconds. If the Power Button is not pressed again, the pop-up screen will disappear after 10 seconds.
- If the Power Button is pushed and held for 4 seconds, the software will power down the system.
- If the software isn't running (bootloader running, problems, etc.), and the Power Button is pushed and held for 10 seconds, the system will power down even without Software control.
- Whenever the software powers down the system, it first saves information to FLASH (like “saveAll”), then pauses 1 second to allow the FLASH write to finish, then disables FLASH writing and disables task switching and powers down.
  - Input “SW\_COMMON” is now repurposed to be “SHUTDOWN\_24”, a digital output that can shut down the 24V directly (e.g. if I2C isn't working)
- Compiler optimization changed from “none” to “-O2” (“more optimization”)
- Front panel displays swapped: Pressure on the left, FiO2 on the right, corresponding knobs also swapped to match displays
- Maximum FiO2 setpoint is now 97%.
- Disable the Battery Monitor chip on the Power Board because it glitches when AC power is removed or replaced.
- Add new Oxycell Zero Failure code: 0x60. This occurs when the date on the Main Board hasn't been set.
- Increase Power Task execution rate from 511 msec to 51msec, because now it has to monitor the Power button and not miss a push.

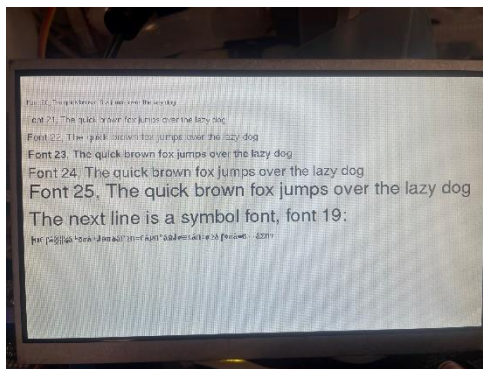
## Version 2.14.10

- For SPI bus semaphore, if it ever times out, return without continuing

- For SPI FLASH, if the “flash\_write\_enable()” fails in any way, return without continuing
- For designing screens, add debug capability to display fonts on LCD:
  - “setGraphics(14)” command will show



- 
- “setGraphics(15)” command will show



- 
- “setFont(x)” will display a line using whatever Font index is “x”. NOTE: this can cause LCD failures requiring reboot – it’s for debugging only

## Version 2.14.9

- External Battery Monitor enabled
- All SPI-FLASH semaphore takes are checked for timeout (meaning the SPI bus is being used by another task), and if there is a semaphore timeout the function returns a negative value. Previously, the “SPI timeout” indication was ignored for the SPI FLASH operations, setting up a possible conflict with the LCD or Front Panel holding onto the SPI bus.
- All SPI-LCD semaphore takes are checked for timeout (meaning the SPI bus is being used by another task), and if there is a semaphore timeout the function returns without continuing. Previously, some “SPI timeout” indications were ignored for the LCD operations, setting up a possible SPI bus conflict.

## Version 2.14.8

- Working release of Front Panel Board software, I2C Battery monitor enabled

## Version 2.14.7

- Working release of Front Panel Board software (requires Front Panel board)
- Fix for bug when removing temperature probes from circuit. Previously, the code would think that negative baby pressure required the blower to switch to “minimum”, causing the blower to fluctuate between 100 and 400 back and forth. Now, we just limit any baby pressure under -1.0 to -1.0, and that limits the maximum blower speed when a temperature probe is removed, the blower stays stable. (Tested by removing each temperature probe and observing low-pressure alarm)
- Change mute time to 60 seconds (was 120 seconds), per requirements for FIH
- Add “Device Failure” alarms to detect pressure sensors or flow sensors outputting illegal values and generate an alarm condition (still missing appropriate screens for LCD).
  - Pressure sensor failure: raw A/D reading outside range 300-3800 (in raw A/D counts, range 0-4095) is a failure
  - Flow sensor failure: I2C errors
  - Flow sensor failure: flow reading outside range -2.0 to +31.0 lpm is a failure
  - New state for Device Failure alarm (missing appropriate screen)
- SPI baud rate set to 400000 (was 200000 in previous version)
- Retry logic if LCD initialization fails. If LCD can’t be initialized, keep trying every 2 seconds (product needs LCD for correct operation). With certain Front Panel failures on the SPI bus, I discovered that retrying the LCD init function eventually succeeded, and the LCD started working.

## Version 2.14.6

- Working release of Front Panel Board software, now merged into “develop” branch
- Front panel enabled: use 5 front panel buttons and 2 rotary encoder wheels for all inputs
- Touch panel on LCD disabled
- Old encoder knobs disabled
- KNOWN ISSUE: LCD is not stable when connected to SPI bus through Front Panel board.
- SPI baud rate lowered to 200,000 (was at 400,000 before Front Panel board issues)

## Version 2.14.5

- Working release of Front Panel Board software
- Add diagnostics for LCD errors – when LCD board detects SPI errors, the errors print out on the terminal
- If there are any timeouts (semaphore timeouts) on the SPI FLASH communications, give up the FLASH operation, don't retry
- Handle maintenance screen (push buttons: “Mute”, “Cal”, “Mute”, “Cal”, “Mute”) when button sequence is entered.
- Buttons will now handle “Change filter” screen interaction and “Power Disconnected” popups.
- Remove “lock screen” stuff from code, since we not longer support lock screen button
- If we detect “Flash Busy” in the LCD screen software, do NOT update screen and do not do any SPI operations until FLASH is done
- Set SPI speed at “400000” instead of “200000”. (There is no effect on the LCD problems happening by changing SPI speed.)

## Version 2.14.4

- Working release of Front Panel Board software
  - Increment FiO2 by 5% increments, not 1% increments, to make knob more usable
  - Make sure the SPI sharing in the LCD code is robust, in case the RTOS isn't running yet
  - Left, Right, Calibrate, Run/Pause, Mute buttons working now with system
  - For buttons use “Debounced” registers to be sure not to miss a push, and to be sure not to register a push twice
  - Make button assignments compatible with FlowWorks software
  - Add debug commands “LcdInit” and “FPInit” to reinitialize subsystems if desired
  - Include “temp\_insp” from the Oxycell sensor in data sent up to FlowWorks host

## Version 2.14.3

- Working release of Front Panel Board software
  - Displays connected to FiO2 and Pressure, not flickering, not noisy (disable with “toggleKnobControl” command)
  - Buttons connected and active
  - SPI bus baud rate back to original 400,000

- If Front panel not detected (powered off, not working), don't try to use it
- FiO2 and pressure initialized from FLASH on power up, working OK
- Fix and test Gray Code converter
- Fix to eliminate display flicker and “knob noise”: in “front\_panel\_read\_register()” after reading out the 8 bits of the register, do a couple dummy writes “spi\_write8()” before raising the Chip Enable line. By doing this the display does NOT enter the “display test” mode, so no flicker, and no knob noise

## Version 2.14.1

- Initial release of Front Panel Board software
- Connects MAX6954 chip through SPI bus to control:
  - Rotary encoders for adjusting FiO2 and Pressure
  - 5 switches to control Left, Right, Mute, Cal and Run
  - Control 4 of the 7-segment displays for FiO2 setpoint and Pressure setpoint
  - The “IRQ” from the chip is used to detect a change of any switch or knob level

## Version 2.13.10

- Cancel blower purging 10 seconds after pausing therapy session. (Just cancel once in SETTING\_STATE.)

## Version 2.13.9

- Remove HAL timers from Alarm Delay and State Delay to improve reliability. (I was seeing these timers from <timers.h> seem to stop running in certain conditions – breakpoints in the timer service routine were never hit once in these “conditions”, that caused alarms to never cancel etc.)
- Fix bug in “PAUSE” state where sometimes the purge blower running after shutting down a session would continue forever. The code to shut down the blower never executed before.

## Version 2.13.8

- To prevent FLASH corruption from damaging board calibration, move the “write once” constants from main FLASH to a special, “write once” factory sector
  - Board serial number now stored in factory FLASH (address 0x66000)
  - Press\_insp scale factor, determined during pressure sensor calibration to precisely set pressure measurements, now store in factory FLASH



- Press\_exp scale factor, determined during pressure sensor calibration to precisely set pressure measurements, now store in factory FLASH
- The command “showFlash” will display the regular FLASH, and the factory FLASH at the end.
- The press\_insp and press\_exp scale factors can be written into factory FLASH using the commands pressInspScale(x) and pressExpScale(y), or by doing a complete “calibratePress” with a calibrated pressure meter.
- Set limits of Oxycell manual calibration to use calibration factor ONLY if the measured oxygen is below 23% and above 19%; otherwise the calibration is considered “in error”, so the manual calibration won’t be used and the raw oxygen sensor reading will be used.

## Version 2.13.7

- Fix for pulsing LED’s, no more flashing. (Put “pwm\_disable” before “pwm\_set\_parameters” and “pwm\_enable” afterwards.)

## Version 2.13.6

- Oxycell enabled
- Oxycell manual calibration tested and working to spec (delay after closing prop valve tested to reduce oxygen level to 20.7 to 20.9 for calibration)
- Oxycell Draeger calibration tested and working to spec after running CPAP at 100% O2.
- FiO2 control loop speedup, using measured oxygen flow to move prop valve to position faster (flow sensors respond quickly, oxygen sensor responds relatively slowly)

## Version 2.13.5

- Oxycell enabled
- Disable Occlusion Alarm – happening too often
- This release calibrates the Oxycell depending on the new position of the prop valve installed near the blower
- Tune FiO2 control filter for new pneumatics; reduce loop gain from .015 to .010
- If the flow\_combined is less than flow\_oxygen, we detect this and start to close the prop valve
- Inhibit “Set Pressure parameter...” messages for clarity
- Run blower longer after hitting “Pause” after therapy to clear out Oxygen
- If we see baby pressure < 0 (bad condition) limit the blower speed to blower(100).

- The FiO2 setpoint can be anything up to 100, but we will never servo on anything above 97. The Oxycell could easily be showing 98 at 100% Oxygen.

## Version 2.13.4

- Oxycell enabled
- Fix FiO2 control bug: when FiO2 setpoint is at minimum 21, don't run the FiO2 control loop at all unless the O2 reading is >22 (i.e. unless we're trying to decrease the oxygen level). This will prevent the prop valve from opening at low FiO2 setpoints.
- Improve FiO2 controls for low-pressure response. Noticed that at pressure of 3cmH2O the FiO2 would overshoot the setpoint by > 7 points. By reducing the gain for pressures < 5cmH2O, this reduces the overshoots and slows down the response time for FiO2 changes.
- Oxycell calibration during therapy: for special case where we're running with no Oxygen (FiO2 set at 21%), do the manual calibration instantly without trying to close the prop valve or wait.
- Fix Oxycell manual calibration for case where the factor is already in use (need to start with raw O2 reading to get proper calibration, not raw O2 reading times calibration factor).

## Version 2.13.3

- Enable Oxycell – when FiO2 setpoint is 98 or 99 or 100, set it instead to 97.5% for servo-ing control. The Oxycell could easily be reading 98.5 at actual 100% oxygen because of calibration error.
- Fix FLASH bug when we're at the last valid sector at 0x63000 at bootup. The bug was generating a Self Test error because the old logic didn't recognize that the FLASH sectors had to "rollover" back to 0x0.
- Fix LCD Screen (or 7-segment) display of Oxygen reading – we weren't using the average value before, we were using instantaneous and noisy Oxygen readings.
- Some Oxycell calibration improvements, for calibrating at atmosphere after a therapy session (purge longer, and retry if there's a calibration error from Draeger), and also for calibrating during therapy session at 48 hours (get better estimate of atmospheric pressure, improve FiO2 control when restarting)

## Version 2.13.2

- Enable Oxycell – leave Oxycell powered on even when resetting board (O2\_PWR pin defaults to always high)
- Calibration of circuit:

- Fail calibration if outside limits: `c_leak` calculation must be  $> 0$  and  $< 0.1$ . On the screen, the calibration will fail and the “setup” screen won’t appear until the user successfully calibrates their circuit
- get rid of all references to “`c_factor`” – it isn’t used, replaced by `c_leak` and `c_factor_insp`
- FiO2 PID loop:
  - several proportional valves display new response characteristics, such as cracking open at much lower signals than previously. Instead of using “45%” as the minimum, starting value of the FiO2 loop, we will now use “25%” so these new prop valves can be controllable.
  - Change FiO2 “ERROR SCALE FACTOR” from .005 to .015 to increase responsiveness. Observed before that increasing FiO2 from 29 to 50 took over 1 minute to increase at all, and after several minutes was nowhere near 50.
- Heater Plate PID loop:
  - Noticed heater plate control banging from “0” to “4000” to “0” back and forth when it gets near 71C. Decreased KI factor during warmup from 1 down to .2.
- Oxycell Calibration:
  - Noticed that O2 “Zero” procedure on Draeger board often fails because the O2 level after a session continues to drift up from 20.6 to 21 to 22 as residual oxygen creeps into the system. If the “Zero” fails, retry the Zero after blowing the fan for awhile (this seems to work). The “Failure Code” turns out to be “3”, meaning “Noise in Sample Data”.
- Oxycell startup: if the MCU detects that the Oxycell is already powered up and at full ISO accuracy right at bootup, it reduces the “warmup wait” time from 3 minutes to 20 seconds.
- Oxycell Pressure: according to Bjoern, we need to send our “`p_insp`” pressure to the Draeger board every time it changes. Their O2 measurement always needs to know the pressure, so from now on we’ll download the pressure every 1 second to the Draeger board (whenever we sense it has changed more than .2 cmH2O).

## Version 2.13.1

- Enable Oxycell – first release with Oxycell active.
- Oxycell calibration:
  - Oxycell will do a “Zero” operation (auto-calibration) when either 24 hours pass or if the internal “Module Status” requires a Zero operation.

- If 48 hours pass with no auto-calibration (possibly inhibited because therapy is active), the Oxycell “manual calibration” will be started during a therapy session, and a calibration factor will be selected based on assuming actual O2 concentration is 20.7%.
- When Calibration is in progress during a therapy session, the screen will not show actual FiO2, but instead will show the word “CALIBRATING”; the controls will be locked for the entire calibration event.
- When booting up, if the FLASH is read with an incorrect Checksum, indicating a bad FLASH sector, that FLASH sector will be erased, and the previous FLASH sector will be checked. If that checksum is correct, the previously saved FLASH sector will be used in this new session.

## Version 2.12.9

- Fix for circuit calibration when blower is on when it starts: be sure the blower is completely off before trying to zero the pressure sensors, or the c\_factor and other parameters will blow up because the pressures look negative
- Working Oxycell calibration during therapy: after 26 hours, where the system has been trying to calibrate at atmosphere, the system will now interrupt the O2 flow by closing the propvalve, flushing the Oxycell, then doing a calibration at pressure, and calculate a “calibration factor” which will correct oxygen concentration errors. When the calculation is complete, the FiO2 control loop is reestablished, the calibration pressure and calibration factor are stored in FLASH, and the therapy continues.

## Version 2.12.8

- Fix for calibration: instead of sampling flows and pressures in calibration routine, just do “os\_delay(25)” to let the “Pressure and Flows” task run (it runs every 19 msecs). This will prevent the interference between tasks sharing the ADC resources.
- Add Oxycell calibration during therapy, using the pressure and creating a manual calibration factor to use when we can’t shut down therapy to zero the Oxycell at atmospheric pressure.

## Version 2.12.7

- Fix for equation for expiratory flow:
  - If  $P_{ex} \leq 0$ :  $F_{ex} = 0$
  - If  $P_{ex} > 0$ :  $F_{ex} = 1.58 * P_{ex}^{0.554}$
- Fix for calibrate pressure routine (factory pressure calibration “calibratePress”): turn off regular pressure sampling while calibrating pressure.
- Fix for c\_factor calibration equation, was miscalculating “press\_ckt”:

- `press_ckt_exp = press_exp.avg + (0.00218 * flow_exp.avg * flow_exp.avg) + (0.0623 * flow_exp.avg) - 0.0817;`
- (Old equation with bug: `press_ckt_exp = press_exp.avg = (0.00218 * flow_exp.avg * flow_exp.avg) + (0.0623 * flow_exp.avg) - 0.0817;`)
- Fix for alarms:
  - if the FiO2 PID control is disabled, kill the “High FiO2” and “Low FiO2” alarms
  - if the Pressure PID control is disabled, kill the “High Press” and “Low Press” alarms
  - if the heat plate PID control is disabled, kill the “High Plate Temp” and “Low Plate Temp” alarms
  - if the heat wire PID control is disabled, kill the “Heat wire disconnected” and “Low Temperature” and “High Temperature” alarms.
- Disable the heat wire monitor, disable the “HEAT CABLE DISCONNECTED” alarm. There is a hardware problem that the INA219 chip cannot function when there is modulating PWM signals on the shunt resistor. It causes I2C errors in communications.
- New command “muxAddress(x)” switches the digital Mux to the address “x”, for instance “muxAddress(3)” would set MUX0 to 1, MUX1 to 1, and MUX2 to 0. “muxAddress” with no argument will read back the 3 address lines and the SW\_COMM line.

## Version 2.12.5

- New equation for expiratory flow:
  - $F_{ex} = 1.58 * P_{ex}^{0.554}$
- Reset Heat Plate PID loop to “warmup” if the heat plate temperature drops below 68°C and we haven’t reached our  $t_{distal}$  setpoint yet
- Reduce the number of “I2C Error” printouts
- Don’t try to read the flow sensors if a FLASH operation is in progress (reduces I2C errors that happen during FLASH writes)

## Version 2.12.4

- Disable touch screen initialization
- Add alarm for heat wire disconnected. When we have reached the “running” screen, monitor the heat wire current, and if the heater setting is  $> 3000$  and the heat wire current is  $< 100$ , declare the “HEAT CABLE DISCONNECTED” and put up alarm screen. This screen has precedence over the “LOW TEMPERATURE” screen, because the disconnected cable will CAUSE a low temperature. NOTE: if the heat wire was disconnected on startup, we assume the user doesn’t have a heat wire or temperature probes, so we disable the alarm.
- Change “LOW TEMPERATURE” delay from 120 seconds to 180 seconds before alarming

- Increase “SPI\_BUS\_DEFAULT\_TIMEOUT” from 83 to 166msecs, because we’re seeing intermittent “rd8 Xout” and other LCD timeout errors.
- Sample heat plate and t\_prox every 99 msecs to filter out lots of noise on the heat plate thermistor and on the t\_prox after the heat wire
- Sample heat coil current every 59 msecs, and filter heavily (50 samples average)
- Fix heat wire runaway condition when heat wire heater is disconnected for some time, then reconnected. This had been causing a thermal runaway (heat wire went up to 48C)
- Change PID parameters:
  - heat plate warmup KP: reduce from 2500 to 1000 to dampen oscillations
  - heat plate Integral limit: set to 20000 (wasn’t used before)
  - heat wire KP: reduce from 10000 to 2000 (reduce oscillations)
  - heat wire Integral limit: set to 10000 (wasn’t used before, prevents runaway when heater disconnected)
  - heat plate steady state KP: reduce from 2000 to 100
  - heat plate steady state KI: increase from 1 to 5
- Heat Plate PID loop now uses a full state machine to transition from “warmup” where it keeps the heater plate at 71C to “steady state” where it keeps the t\_distal at 37C.
- New command to download firmware: “bootloader”. This command will put the board in bootloader mode, and you can then copy the “FlowLiteATSAMD51J19A.uf2” file into the Window that pops up.

## Version 2.12.3

- Change heat plate thermistor to type “NTCS0805E3103FLT” where the “beta” value in our firmware is changed from 3984 (old thermistor) to 3430 (new thermistor) as shown in datasheet [https://www.mouser.com/datasheet/2/427/VISH\\_S\\_A0012884891\\_1-2572708.pdf](https://www.mouser.com/datasheet/2/427/VISH_S_A0012884891_1-2572708.pdf).

## Version 2.12.2

- Add new field in “pidStream” and “pidDump” commands to indicate errors:

LOOP	,	EN, SETPT, ACTUAL, CONTROL, ERROR, INTEGRAL, MAX, INTERLOCK
fio2_rate,		1, 44.00, 20.79, 60.96, 23.21, 0.00, 6500.0, 2
press_baby_pid,		0, 5.00, 0.49, 0, 4.51, 0.00, 511.0, 0
heat_plate_pid,		1, 37.00, 34.82, 4000, 37.18, 6389.04, 4000.0, 0
heat_wire_pid,		1, 39.00, 20.32, 0, 18.68, 2417.59, 10000.0, 3

In this example, note the interlocks present on “fio2” (interlock 2) and “heat\_wire” (interlock 3) due to the blower being off. Anytime the interlock is non-zero, the PID loop is experiencing a problem preventing normal operation.

\*\* For “fio2\_rate” and “press\_baby\_pid” the interlock codes are:

```
NO_GAS_INTERLOCK=0,  
GAS_PRESSURE_SENSOR_BAD_INTERLOCK=1,  
GAS_NO_BLOWER_INTERLOCK=2,  
GAS_FLOW_INSP_NO_GOOD_INTERLOCK=3,  
GAS_FLOW_EXP_NO_GOOD_INTERLOCK=4,  
GAS_BAD_OXYGEN_SENSOR_INTERLOCK=5
```

\*\* For “heat\_wire\_pid” and heat\_plate\_pid” the interlock codes are:

```
NO_HEAT_INTERLOCK=0,  
HEATER_LOW_BATTERY_INTERLOCK=1,  
HEATER_TEMP_PROBE_DISCONNECTED_INTERLOCK=2,  
HEATER_NO_BLOWER_INTERLOCK=3
```

## Version 2.12.1

- (First release on “develop” branch for board v5.0)
- Commands “pidDump” and “pidStream” will now display values of the PID loop like “actual” and “control” and “error” correctly, even with the PID disabled. Previously, these parameters were ONLY updated when the PID loop actually ran. (Note the PID loop doesn’t run if there are interlocks preventing it.)
- The “Low Temperature” alarm only should occur when in active running screen or alarm screen. Previously, it showed up whenever the “heatWire” or “heatPlate” PID loop was enabled. The high temperature alarm will still occur on any screen.
- Oxycell improvements: only start a “Oxycell Zero” cycle when the system is warmed up, the Oxycell status is “good”, and the Oxycell hasn’t calibrated within 24 hours. (Show these times in the “oxycellData” or “o2Status” command. Also, commands “o2Status” and “o2Calibration” will now use the Oxycell commands, whenever Oxycell is compiled in
- (Oxycell NOT in this release – this release uses the SST sensor)

## Version 2.11.10

- Don't allow Oxycell "Zero" during 3-minute warmup after power on. Include the 3-minute warmup status in the "oxycellData" command, so user can see why the Oxycell is showing "bad" status for 3 minutes
- Only check for Oxycell comms errors when power (O2\_PWR signal) is "on". Oxycell comms errors can be disabled with "o2Power(0)".

## Version 2.11.9

- O2\_PWR now working (removed pullup in firmware), confirm that 24V goes on and off using command "o2power(1)" and "o2power(0)".

## Version 2.11.8

- Battery Status I2C channel disabled

## Version 2.11.7

- O2\_STATUS on PA02 and O2\_VALUE on PB07 (O2\_STATUS was reading ADC1 in previous version, needs to be on ADC0).

## Version 2.11.6

- O2\_PWR on PB11 (was on PB17 in previous firmware).

## Version 2.11.5

- The SST Oxygen sensor is enabled, and the Oxycell sensor is disabled.
- The O2\_PWR output should definitely be high now, and has an internal pull-up.
- If the LCD isn't present, be sure never to check the touchscreen (would generate additional errors)

## Version 2.11.4

- The avgStream command now includes prop valve, blower, heat wire and heat plate setting

```
,P_INSP, P_EXP,F_INS,FL_OX,T_DIS,T_PRX, T_HT,T_PCB, FIO2, BABY, F_EXP, F_LEAK, O2_AVG, PROP_V,  
O2_SET,BLOWER,HT_WIRE,HT_PLT  
  
,001.34,001.03,05.67,-0.00,18.38,18.07,20.20,28.83,021.95,001.22,002.24,003.43,000.00,50.000,21.000,00200 , 00000,  
00000  
  
,001.36,001.05,05.66,00.00,18.39,18.06,20.21,28.83,021.95,001.24,002.27,003.39,000.00,50.000,21.000,00200 , 00000,  
00000
```

- The knob encoders now are fixed, incrementing and decrementing FiO2 and Pressure
- The battery monitor is now enabled, meaning the battery must be putting out > 18V to avoid I2C errors.



- The O2\_PWR signal (PB11) is now set “high” on power up.

## Version 2.11.3

- For testing board revision 5.0, disable the comms with the external Battery chip. All battery monitoring is invalid in this release. We don’t have the ability to detect “power unplugged”.
- Change “avgStreamRate()” command to accept floating point numbers, so e.g. “avgStreamRate(.2)” would output the stream every 5 seconds.

## Version 2.11.2

- Bug fix for board revision 5.0, print out floating point numbers correctly.

## Version 2.11.1

- Bug fix for board revision 5.0, using the ATSAMD51J19A part. The “ADC\_1” driver was never starting, so the conversion in main.c: “`adc_sync_read_channel(&ADC_1...`” was hanging up incorrectly.

## Version 2.11.0

- Build for board revision 5.0, using the ATSAMD51J19A part, implementing the following pin assignments
  - KNOB\_1 (FiO2): pin PB00
  - KNOB\_2 (Press): pin PB01
  - MUX\_1: pin PB04
  - MUX\_2: pin PB05
  - O2\_SENSOR: AIN\_9 on ADC1 (pin PA07); (was AIN\_7 on ADC0)
  - O2\_SENSOR\_STATUS: AIN\_0 (pin PA02); (was AIN\_13)
  - O2\_CAL (calibrate output signal): pin PA04; (was PB00)
- Added “sample\_adc\_1()” and “set\_adc\_1” to allow for sampling and setting ADC1, first used on board revision 5.0.
- Moved external IRQ functions from PB04 and PB05 to PB00 and PB01
- Add stub for new PWM to PB17 (O2\_PWR is current output on PB11). Since PWM can’t be assigned to PB11, I chose PB17. There is no active code to enable/disable O2\_PWR.

## Version 2.10.6

- Oxycell operation fixes (NOTE: Oxycell disabled in this build):
  - 3-minute warmup period – keep status at “0” for 3 minutes, ignore status from Oxycell for 3 minutes

- If losing comms, set date when comms restores, plus restart 3-minute warmup timer
- Keep updating screen FiO2 value when LCD hardware is missing. There was a bug where we only update FiO2 readings for the screen if the LCD is connected, which makes the Windows simulator break, because the FiO2 readings are frozen.

## Version 2.10.5

- To prevent I2C errors (from flow sensors or battery monitor) from causing lockup, need to add "schedulerState == RUNNING" test, so we never call "os\_sleep()" when the scheduler is suspended, which happens when we're resetting the I2C bus during I2C error conditions.
- Show "--" for FiO2 when Oxycell or SST not ready. Don't start session until O2 sensor is showing "Good" status.
- Instead of showing "RUN" on the ready screen, it will now show "WAIT..." until the O2 sensor is showing "Good" status. Also, if the O2 sensor goes to bad status, the PID loop will not update until the status changes back to "Good".
- Optimization set from "none" to "more" to save MCU memory space.
- Implement Oxycell interface (compiled out by default – SST Sensor is still the default)
  - Oxycell now programmed with date/time.
  - Command to zero Oxycell: "calibrateOxycell" - takes 30 seconds to calibrate at room pressure.
  - Set date to Oxycell on every power up.
  - Read back Oxycell date in "oxycellData" command.
  - Read back ambient pressure in "oxycellData" command (not integrated into product yet).
  - For oxygen\_status, always set to "0" or "Invalid" when zero-ing sensor, otherwise just use lower 2 bits to determine accuracy.
  - Oxycell Zero calibration is now automatic. In order to determine when to Zero the Oxycell we look at:
    - 1) > 24 hours elapsed since last Zero
    - 2) The unit has been powered on for at least 30 minutes
    - 3) The tube has been purged of oxygen so we can zero at atmosphere
    - 4) The blower is off
    - 5) The unit is NOT in operation
- Leave blower on after pressing "Pause". In order to evacuate the oxygen from the tubing after stopping a session by pressing "Pause", we now let the blower run at a reduced setting for 12 seconds in the "Setting" state, then shut off the blower completely. This was tested to reduce the oxygen to below 24%, even if the FiO2 setpoint was 100% before hitting pause.
- Limit prop valve. If the O2 tank cannot provide sufficient oxygen to achieve the setpoint, we detect this problem by seeing that the flow ratio of flow\_ox/flow\_insp is > 1.1. If this happens,

we hold the prop valve at the current value and don't open it anymore. This should deal with both weak oxygen concentrators and O2 tanks with low-concentration oxygen gas.

## Version 2.10.4

- To use the Atmel ICE to program the board, changes were made in this version.  
BOOTLOADER: because the bootloader differs between the different MCU's in use (different bootloader for -J19A and -J20A), and because erasing the chip using the "ProgramAtmel.bat" files could accidentally erase the bootloader, this version contains new, working program files to use with the Atmel ICE programmer:
  - ProgramAtmel.bat, using FlowLiteATSAMD51J20A.hex
  - ProgramAtmel\_J19A\_chip.bat, using FlowLiteATSAMD51J19A.hex
- To use the bootloader to load firmware, enter the bootloader by rapidly double-pressing the board reset button, and when the "SAMD51 (D: )" window appears, copy the appropriate \*.uf2 file into that window
  - FlowLiteATSAMD51J20A.uf2 for the -J20A MCU
  - FlowLiteATSAMD51J19A.uf2 for the -J19A MCU

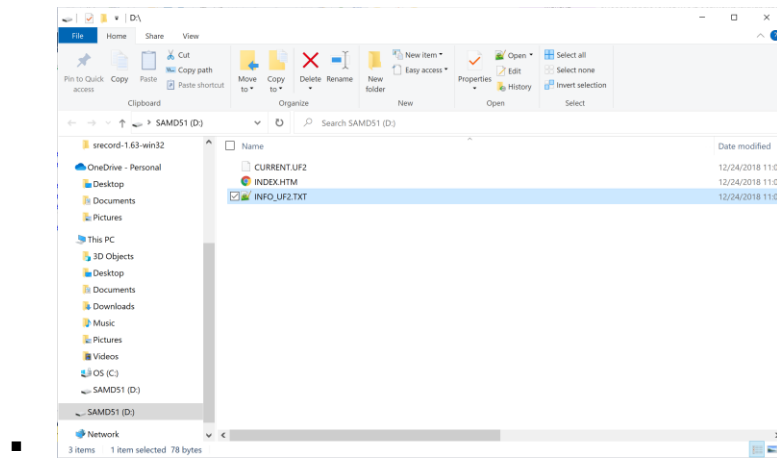
## Version 2.10.3

- BOOTLOADER
  - Include bootloader version 1.4 with pulsing white LED's
  - Don't lock "boot protect" fuses – we want to be able to update the bootloader as well as the main application using the USB, which won't work with the boot protection fuses set.
  - In bootloader mode, info in pop-up window will conform to EH product names

## Version 2.10.2

- BOOTLOADER
  - Firmware file "FlowLite.hex" will now contain both the application firmware AND the bootloader. This will be programmed correctly with the "ProgramAtmel.bat" file and the Atmel ICE device.
  - After programming with "ProgramAtmel.bat" the "bootloader protection" fuses are set so the bootloader cannot be modified or erased. Before programming, the "bootloader protection" fuses are cleared so the bootloader can be updated. This is all done with the Atmel ICE device.

- Once the bootloader is installed, the board may be programmed with the USB port, not requiring the Atmel ICE. Enter the bootloader by grounding pin P9-4 and resetting or power-cycling the board. If the USB port isn't connected, connect it to Windows. Now the programming window will pop up:



- Locate the “FlowLite.uf2” file and drag-and-drop the file onto the “SAMD51 (D:)” drive.

## Version 2.10.1

- NEW FEATURE: BOOTLOADER COMPATIBILITY
  - Move code up to address 0x4000 to make room for Bootloader at address 0x0
  - Don't use RX and TX pins for debug terminal – do everything over USB
  - As part of the standard Atmel build process, the output file “FlowLite.hex” is converted to the bootloader file “FlowLite.uf2”. When in “Bootloader mode”, the file FlowLite.uf2 can be dragged over to the new “D:” drive in Windows, and that will program the v2.10.1 application onto the board

## Version 2.9.7

- The temperature shown on the screens (setpoint screen, run, and alarm screen) is now the less of t\_dist and t\_prox. That means the temperature displayed will usually be t\_dist, the lower temperature coming out of the humidity chamber. NOTE: the temperature alarms are still using t\_prox to decide on high temp and low temp alarms, not t\_dist. T\_dist takes much longer to heat up, so using that for alarming will require longer delay times in the alarm timing.

## Version 2.9.6

- Change UART from synchronous to asynchronous. Synchronous severely limits the Rx capability of the UART because the Rx buffer just overwrites if data comes in faster than it's processed. In asynchronous mode, the Rx is interrupt driven and can run as fast as the baud rate

allows. Tx must use the “TX COMPLETE” interrupt to signal when the next output can start, or the output could just overwrite the Tx buffer.

- NOTE: unable to go above 115200 baud (just get gibberish at 460800), probably because the Arduino is limiting the baudrate with its serial<->USB converter.

## Version 2.9.5

- Compiled so all debug traffic goes out the UART on pins PA22 (Rx) and PA23 (Tx). Port is configured for 115200 baud. USB port in this version is unused.
- To connect, need an RS232 – USB converter cable or equivalent to connect to P9-4 and P9-5, and to the computer. (Suggestion: FTDI C232HD-DDHSP-0)

## Version 2.9.4

- Run FiO2 control loop at 1.1 seconds instead of 1 second to reduce overshoot.

## Version 2.9.3

- Bug fix: when AC power is connected, observed current into battery as “-9mA”, so change “AC Power Detected” as anytime the battery current is > -20mA (to properly display AC plug on LCD)
- Bug fix: put “AC Power disconnected” splash screen last in “screens.c” so no other screen can paint over this splash screen.
- Alarms: make sure when a parameter is over- or under-range, always draw that parameter in red, e.g. if there is a “High Temperature” alarm, the temperature numbers will appear in Red on the alarm screen
- Touch screen improvements: sample touch at 100Hz instead of 10Hz, to be sure to capture touch events as quickly as possible. Still run state machine at 10Hz as before.
- Screen refresh: make the screen refresh rate be 0.5 seconds, per the Product Spec (requires 0.5 to 2 second update rate)
- Doxygen documentation: add “Doxyfile” to configure Doxygen Wizard to product formatted documentation for all the comments in the code

## Version 2.9.2

- Startup screen improvements, closer to user study:
  - Add larger “Progress Bar” that updates as the Power Up self-test progresses. Instead of 1 black bar for the whole test, the progress bar animates the progress, is colored blue, and is thicker than before

- Remove “NEXT” button from startup screens. The screens progress by themselves, and don’t require a “NEXT” intervention. The “Startup complete” screen has a next button, but will also progress by itself to the first “Instructions” screen.
- If the user wants to enter the “Maintenance screen” code on the “Startup complete” screen, the system gives extra time to enter the correct button sequence
- Command “pidDump” will show the “fio2\_rate” max change, because we jump from maximum prop valve of “65” to “75” when we detect flow, indicating that the O2 tank is connected

## Version 2.9.1

- Alarm improvements (back on Develop branch):
  - When there are multiple alarms, and the alarm that is on the screen ends, be careful switching back to the “previous screen”, which in this case might be the previous alarm. Before switching back, check if the alarm conditions still exist for that alarm (new function: `machine_state_matching_alarm_state()`). If the previous alarm is no longer present, just go back to RUN or SETTING state.
  - Don’t present “LOW BATTERY” screen during “RAMP” state, because it prevents the system from ever leaving RAMP state. In RAMP state certain alarms are inhibited (because everything is changing), so disable the LOW BATTERY alarm during RAMP, then re-enable it when RAMP is complete.
  - Add a “catch-all” check for an alarm screen being presented after the alarm condition is gone. If an alarm screen is present with no alarms present, switch back to “RUN” or “SETTING” state. There should never be an alarm screen present without an alarm condition, but there may be race conditions when multiple alarms are present and we’re trying to recover states, so this protects against that occurrence.

## Version 2.8.17

- After muting an alarm 4 times, don’t allow any more mutes
  - When the alarm buzzer is sounding, pushing “MUTE” mutes the audible alarm for 2 minutes, then the buzzer sounds again. The user can press “MUTE” up to 4 times to silence the buzzer, but after 4 pushes the “MUTE” button disappears and the user must fix the alarm condition to stop the alarm
- Power On Self Test (POST) changes:
  - Add FLASH test in POST – if no valid FLASH sector is found (all sectors either blank or bad checksum), show a failure in POST
  - Remove O2 test from POST – the O2 sensor is never ready when the POST executes, so it always fails

- If any POST failure occurs, don't let the user click "NEXT" to move to instruction screens; instead, the only option will be to "RETRY", preferably after fixing the error. If a POST error occurs in any of these tests, the device will not allow the user to proceed:
  - FLASH test
  - Pressure test (tied to blower)
  - Flow test (tied to blower)
  - Temperature test
- Fix bug when "CRITICALLY LOW BATTERY" screen first appears, it may get overridden by regular "LOW BATTERY" screen – fixed now.
- When an alarm screen disappears and we try to return to the previous state, choose the RUN\_STATE if the pressure PID is running, and the "SETTING\_STATE" if the pressure PID is not running
- Change the heater plate high temperature alarm to 82C (80C can sometimes happen during unusual overshoot conditions)

## Version 2.8.16

- New alarms
  - Low temperature (35°C at baby) (alarm existed, but now a new screen exists)
  - Low plate temperature (35°C under humidifier chamber)
  - High plate temperature (80°C under humidifier)
  - Refill humidifier (not implemented)
- Circuit temperature probes alarm changed. Determine at "RAMP" state whether there is a circuit temperature probe. If not connected during ramping, assume the circuit is deliberately missing, and inhibit all "temp sensor disconnected" alarms, and eventually inhibit all "circuit heater disconnected" alarms (not implemented until circuit heater current hardware is available).
- Alarm behavior improvement: when one alarm goes away, be sure to remove the alarm screen. Previously, if another alarm was waiting in "pre-alarm" mode, it left the obsolete alarm screen up.
- Move "temp\_prox" and "temp\_heater" from gas\_sensors.c to heater.c, because that's where they are used (in the heater PID loops)
- "DEMO UNIT" banner present on all screens. This will be removed when the firmware is built for production. In production, besides removing the "DEMO UNIT" banner, the buzzer is unmuted, the watchdog is enabled, etc.



- For Power On Self test, add a “Memory Test...Pass” line. This indicates the FLASH has a valid sector (non-empty, and correct checksum). “Memory Test...CORRUPTION” indicates the FLASH has no valid sector, and items like “c\_factor” and pressure factors are missing. In production, this will eventually prevent the user from proceeding.
- Bug fix: when in “change filter” state, don’t FLASH red LED’s – it isn’t an alarm condition.
- Add “forcePlateTemp(x)” to command list. This allows testing of plate temperature alarms by forcing the temperature to a fake value “x”. (Debugging only)

## Version 2.8.15

- Multiple alarm improvements
  - When cancelling, start the “delay\_stop” timer even if another alarm is pending, so the screen being shown actually gets cancelled
  - Don’t flag high- and low-temperature alarms if the temp sensor is disconnected
  - Fix buzzer alert (although buzzer is disabled by default)
  - Don’t do FiO2 alarms when the FiO2 setpoint has just changed
  - Obey alarm precedence from [Alarm Priority List](#)
- Tweak FiO2 control loop for new O2 manifold (less dead space, 6cm tube)
- Increase Prop valve PWM clock (PWM2) from 1Mhz to 12Mhz, which increases the PWM Frequency from 100 Hz to 1200 Hz.
- (eliminate all FiO2 calibration – we’re not going to rely on FiO2 calculations at all)

## Version 2.8.13

- Break up “bad temperature” alarm into “high temperature” and “low temperature”. Both still have the same “High Temp” alarm screen for now. The low temperature alarm has a much longer delay time (120 seconds) to alarm, because the heater is slow heating up, but the high temperature alarm is faster (20 seconds), to give prompter reaction.
- FiO2 control changes:
  - In any state, bump up the “offset” by 0.8 until flow is sensed (>0.4 lpm)
  - Reduce the scale factor to 70% whenever we’re > RAMP\_STATE and FiO2\_error is < 8.
  - When we’ve established O2 flow, and setpoint > 23, bump up the prop valve max from 6500 to 7500 (out of 10000).
- For debugging, create new forcing functions to create dummy conditions:

- forceTemp(x) – forces the t\_prox to “x” to simulate over- and under- temperature (note this value will go into the PID loop for the circuit heater)
- forceO2(x) – forces the O2 sensor feedback to “x” to simulate over-FiO2 and under-FiO2 (note this value is NOT used in the FiO2 control loop, so the prop valve doesn’t see it - it is only used for display and alarms)
- forceBabyPress(x) – forces the baby pressure to “x” to simulate over- and under-pressure (note this value is NOT used in the pressure control loop, so that remains stable – it is only used for display and alarms)

## Version 2.8.12

- Add alarms
  - High FiO2 – 10 higher than setpoint for 5 seconds
  - Bad Temperature – t\_prox not between 35°C and 43°C
  - Temp Cable disconnected – if t\_prox or t\_dist is < -40°C, that means the cable is disconnected
  - Heat Wire disconnected warning – not implemented yet, waiting on hardware
- When battery is in “red zone”, less than 21V, disable the circuit heater and the heat plate to reduce power consumption
- Mute function: leave muted for 2 minutes (not 30 seconds); after 4 mutes (using mute counter), don’t allow mute anymore; when all alarms are cleared, restart mute counter
- Start prioritizing alarms based on chart from Robin
- In FiO2 control, do accelerated prop valve open when flow < 0.4 and any state >= RAMP\_STATE and when the FiO2 setpoint > 24 (don’t try to open prop valve if no O2 is commanded)
- For screens needing the “PAUSE” button, only display PAUSE if the are in “running” state (pressure PID enabled), otherwise display “RUN” in that location. (Including low-battery screen.)
- Change FiO2 knob click to be 1% increment. (No longer 5% increment.) This means going from 21% to 100% may take 4 rotations of the knob.

## Version 2.8.11

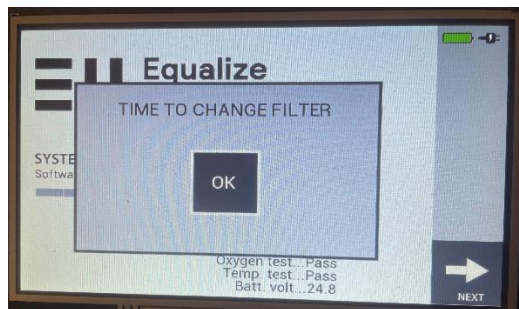
- Re-tune the FiO2 control loop for new Enfield valve (calibrated at 30 psi). Increase bias from 32 to 40 (nominally opens around 50), decrease error\_scale\_factor because valve is more sensitive, increase error\_scale\_factor when O2 sensor max’s out at 100%, because it means

we've overshot the prop valve setting and need to recover back down quickly, even though the measured FiO2 error may look small.

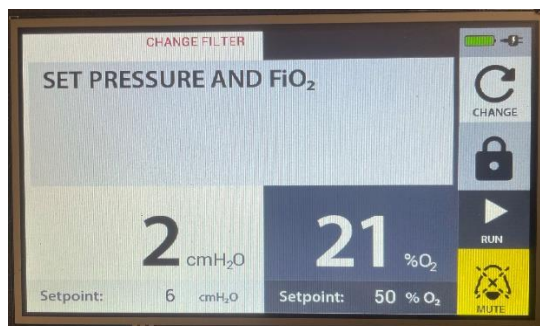
- For Occlusion alarm, de-sensitize the alarm looking for decreased pressure difference, because the flows and pressures become very noisy above about 5lpm flow.
- Change Green LED back to less-glitchy "hri\_tcc\_write\_CCBUF\_reg" instead of "pwm\_set\_parameters()"

## Version 2.8.10

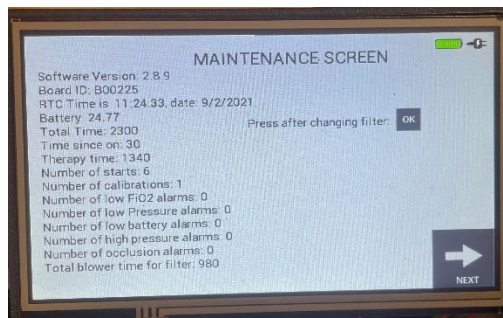
- Add "TIME TO CHANGE FILTER" pop-up screen when accumulated blower time reaches 1440 hours, per PRQ-0047.



- When screen is acknowledged with the OK button, the "CHANGE FILTER" warning appears on all screens in header. The pop-up screen will re-appear every 10 minutes until the filter is changed



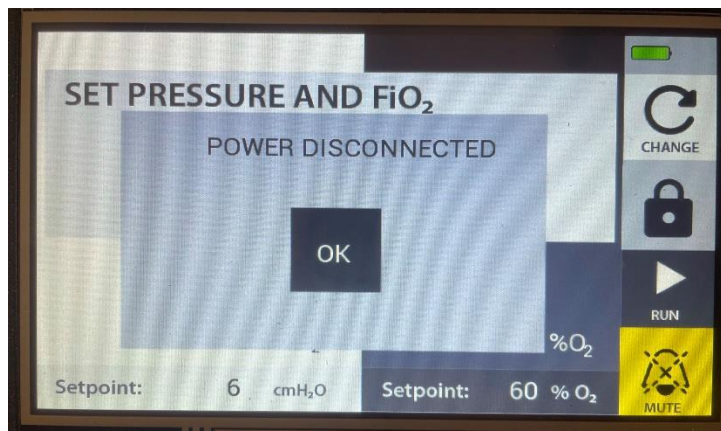
- To clear the "CHANGE FILTER" warnings, you must use the maintenance screen and press a button there. Look for the note "Press after changing filter:". This will clear the blower counter to "0" and restart the 1440 hour clock. (Note the debug command "setBlowerTime(x)" to artificially set the blower time to any value to test the "CHANGE FILTER" screen behavior.)



- For POST test, add a battery voltage test, failing if voltage is less than 20V. (per PRQ-0043)
- Don't allow maintenance screen on any of the "running" screens, because it completely blocks out other screens.
- Fixed a FLASH bug on power up (in retrieving timestamp from FLASH) that caused system fault due to sprintf() failure.
- Add terminal commands:
  - changeFilter – resets blower time to 0
  - clearUsage – resets all usage parameters in FLASH, useful before shipping
  - setBlowerTime(x) – set blower time to "x" for testing

## Version 2.8.9

- Whenever AC power is lost, a pop-up screen appears. (PRQ-0038 "Device shall alarm ... when under battery power").



- Touching anywhere on the screen clears the pop-up screen, and it reappears only if power is applied and removed again

## Version 2.8.8

- New default streaming output: product requirements stipulate “parameters shall be issued out a USB port uninterrupted and on a regular basis”. To satisfy this, the following output stream is on, by default, at 1Hz output rate:

Header (output one time):

```
, Date, Time, Brd_Id, P_INS, P_EXP, F_INS, FL_OX, TDIS, TPRX, T_HT, TPCB, FIO2, BA-  
BY, O2_AVG, PR_VLV, F_SET, P_SET, HTST, HTWR, BLOW, ALM, GUI
```

1Hz Repeating Data:

```
, 08/31/21, 11:33:12, B00225, -0.09, -0.02, -0.00, -0.00, 24.1, 29.0, 36.8, 31.1, 021.95, -  
0.09, 021.75, 032.00, 060.0, 006.0, 37.0, 39.0, 0094, 000, 003
```

- New default streaming can be disabled/re-enabled by entering “defaultStream”. “avgStream” is still available, and the rate of both these outputs is still controlled by “avgStreamRate(x)” to set the rate in Hz.
- Maintenance screen now supports the battery icon and the AC-connected icon

## Version 2.8.7

- New maintenance screen: The screen displays the PRD requirements:
  - (1) total days since first turned on (displayed now in seconds)
  - (2) total usage time (time in therapy, displayed in seconds; time powered on, in seconds)
  - (3) unique device identification; (Board serial number)
  - (4) total number of calibrations. (calibrations since power on)

In addition, the maintenance screen shows software version, RTC time, number of starts, number of alarms (tabulated by alarm type)

- Maintenance screen is accessible by a sequence of 5 screen touches on the “Setup Ready” screen after the POST. It can also be displayed using the “maint” command from the terminal.
- Screen is exited by pressed “Next” to go to the start of the setup screens.

## Version 2.8.6

- Usage requirements: The requirements are only for 5 things:
  - 1) days since turned on
  - 2) usage time
  - 3) Device ID
  - 4) number of calibrations
  - 5) GeoLocation
- Implemented in this version, using new “usage” task running under FreeRTOS:
  - Total “On” time (uses FLASH storage to maintain over power cycles)

- Time since last power “On” (no FLASH)
- Total “Therapy” time in use (uses FLASH storage to maintain over power cycles)
- Therapy time since last power “On” (no FLASH)
- Number of calibrations/Test Circuit pushes (uses FLASH storage to maintain over power cycles)
- Number of “Start” pushes (uses FLASH storage to maintain over power cycles)
- Serial number of Main Board (for Device ID, stored in FLASH)
- Additional: current Machine State, alarm totals for: LOW\_PRESS, LOW\_FIO2, LOW\_BATTERY, HI\_PRESS, OCCLUSION
- New “Usage” command on terminal, to print out:
 

```
Time is 16:37:28, date: 8/24/2021 (1629823048)

Board Serial Number: B00225

Total on time: 5160, Time since power on: 2400, Current machine state: 3
Number of starts: 8, Number of calibrations: 1, Total therapy time: 1060
Therapy time since on: 0

ALARM TOTALS

Low Fio2: 0, Low press: 0, Occlusion: 0, Low Battery: 0, High Press.: 0
```
- Occlusion alarm re-enabled. Added more filtering to “flow\_insp” to get rid of partial spurious occlusion alarms, increasing filter length from 10 to 20 samples.
- Add “Doxygen” style comments at beginning of all software files
- Button “1” in startup screens always skips POST and skips setup and “TEST CIRCUIT” screens and jumps to “ready to run” screen

## Version 2.8.5

- FiO2 control: Use local\_offset=1 as long as the flow\_ox sensor shows "no flow yet", which ramps up the prop valve quickly at startup. Disable this when flow\_ox reaches 0.4.
- Change FIO2\_SCALE\_FACTOR to 0.007 (was 0.01) for calmer reaction.
- GUI: Add "SKIP SETUP" button to "READY" screen to skip setup screens and skip "Test" calibrate screens - jump right to "Adjust" screen.
- Now using J20 MCU instead of J19 (CPU change on main board).

## Version 2.8.4

- FiO2 control: at low pressures, it was taking > 60 seconds to find the setpoint for FiO2, so when far away from setpoint, temporarily increase the “scale\_factor” of the control loop by 30%.

- Set the FiO2 bias to 32 (was 25) to speed up initial search for setpoint.

## Version 2.8.3

- BUG FIX: FiO2 Control Loop. When entering “LOW FIO2” alarm state during ramp up (because of slow control loop response), then the FiO2 finally catches up to the desired setpoint, the “LOW FIO2” alarm goes away, and the prop valve stays at its proper value. Previously, coming out of alarm caused a “reset\_fio2\_pid” which reset the prop valve to minimum, and the ramp had to start over.

## Version 2.8.2

- Improved FiO2 Control Loop: use the O2 sensor as the feedback for the FiO2 control loop, ignoring the flow sensor ratio. Use Costa’s control loop, scale factor 0.01, PID loop executing once per second (slow because of the slow O2 sensor response). Change bias from 40 to 25 (meaning start Prop valve at 25%, for more ramp room to “accelerate”)
- To prevent spurious “Low FiO2” errors, the error delay is increased from 10 seconds to 30 seconds. It takes > 20 seconds now to ramp from 35 to 95 FiO2.
- Change “avgStream” to show: `P_INSP, P_EXP, F_INS, FL_OX, T_DIS, T_PRX, T_HT, T_PCB, FIO2, BABY, F_EXP, F_LEAK, O2_AVG, PROP_VLV, O2_SETPT`

## Version 2.8.1

- New FiO2 Control Loop: use the O2 sensor as the feedback for the FiO2 control loop, ignoring the flow sensor ratio. If the O2 sensor is invalid or reading less than 18, switch back to using the flow ratios for the FiO2 controls.
- Disable Occlusion alarm for now – needs retuning because the flow\_insp is very unstable
- Because the O2 sensor is maximum 100%, we can only servo up to 99%. Otherwise the control loop would “freeze” at 100%, because the feedback from the O2 sensor can never show above 100. By servoing on 99 when asking for 100, we will continue to servo.

## Version 2.8.0

- New FiO2 Controls: Costa implemented changes to the FiO2 control loop to operate effectively at 3-20psi O2 tank inputs. The scale factor is now 0.01, the offset is 0, and the FiO2 averaging is 10-point averaging (the screen value of FiO2 is averaged over 100 points). These values can be changed at run time using “FiO2ScaleFactor(x)” and “FiO2Offset(x)” (values forgotten after reboot).
- New debug command “forceO2Calibration” skips the 10 minute wait and allows you to do an O2 calibration right away if you’re sure the sensor has warmed up.

## Version 2.7.8

- FiO2 display now always shows O2 sensor when O2 sensor is valid (status OK and value > 20). There is no attempt to mask out errors, e.g. if the O2 sensor says “78%” and the FiO2 calculation is “80%”, we still show 78 because that’s the O2 sensor reading.
- Fix for FlowWorks, so the simulation screen reflects this new FiO2 reading being O2 sensor instead of FiO2 calculation.
- Terminal Password implementation. Currently compiled out, so no password required. If “`PROTECT_TERMINAL_WITH_PASSWORD`” in main.h is set to “1”, the terminal will be completely disabled for inputs and outputs until the password “&Drev695!” is entered, then the terminal will be responsive like in development. Note the password will be remembered over sessions if “savePassword” is entered after entering the correct password.

## Version 2.7.7

- Feature change: the FiO2 value displayed on the LCD is now the O2 sensor. Previously, it was the FiO2 calculation (ratio of flow sensor readings). The O2 sensor reading is rounded before displaying. If the O2 sensor reading is bad (status bad, or less than 20), we revert to displaying the FiO2 reading instead.
- Power On Self Test (POST) change: the POST will no longer open the prop valve. The prop valve will be left closed (propvalve(0)).
- FiO2 Calibration improvement: we only want to calibrate FiO2 to the O2 sensor when the O2 sensor is stable. When the O2 concentrator is warming up, the O2 concentration gradually increases over about 2 minutes, making it unstable for FiO2 calibration. We use a statistic based on the sum of derivatives of the O2 sensor reading to determine when the O2 sensor has stopped changing, and only then do we allow the FiO2 calibration.

## Version 2.7.6

- Fix “AC power connected” icon. It turns out with a fully charged battery, the “charge current” is sometimes actually -3mA, meaning the battery is supplying some current even with the charger plugged in. From now on, “plugged in” means battery current > -8.0mA (not > 0mA).
- Improve FiO2 calibration: sense when O2 sensor is reading much higher than FiO2, indicating a possible calibration error (or possibly O2 tank being disconnected/reconnected during operation)

## Version 2.7.5

- For use with “FlowWorks” running on host, send up “lock screen” status and “mute alarm” status.



## Version 2.7.4

- Use standard screen table, so this works with either Arduino-programmed LCD or with “ProgramLCD” programmed device
- Fixes for “FiO2 calibration” against the O2 sensor. Problems occurred when we tried to calibrate too soon after an FiO2 adjustment. The FiO2 calibration must be delayed around 20 seconds every time the FiO2 setpoint is changed. In addition, if a large change is done to the FiO2 setpoint (more than 11), we wait 40 seconds to calibrate FiO2 to let O2 sensor catch up.

## Version 2.7.3

- This release is to be used with LCD units programmed with the new application “ProgramLCD” (see installer file here: <https://drive.google.com/file/d/1YEunGWWDYTHI8v9Md0UD2ZdAWyiCoVI4/view?usp=sharing>)
- Instead of hard-coding the image addresses in the firmware, the information is now all stored in the LCD FLASH along with the images. This should eliminate the old step of copying image addresses from the LCD programming tool into the firmware. Note the order of the images during programming is very important: the index of each file is how the firmware figures out which address to use for the screen.
- During bootup, the table containing the screens and locations is printed on the terminal, e.g.

```
Entering Machine state SETUP_STATE
screen 1: flash: 0x2000, width: 800, height: 480
screen 2: flash: 0x1a000, width: 800, height: 480
screen 3: flash: 0x32000, width: 800, height: 480
screen 4: flash: 0x4a000, width: 800, height: 480
(etc.)
```

## Version 2.7.2

- This release is NOT for the DEMO.
- NOTE: This release requires new screens, to be found here: <https://drive.google.com/drive/folders/1P3KZXNMDZpomw7nefTuw3aeivfjIdmBB?usp=sharing>
- New screens: “Occlusion Alarm” screen and “Low Battery” screen
- New button screens: “Lock”, “Unlock”, “Mute”, “Muted”, “Back”, “Change”, “Next”, “Pause”, “Run”, and “Skip”.
- Only show “Low Battery” screen when unplugged; when the CPAP is plugged in, never show “Low Battery” no matter how low the battery, since it’s charging.

- Instead of using 4 screens for each state (with every combination of “Lock” and “Mute” buttons), use 1 base screen, and overwrite the proper button images. This reduces the number of full screens stored in LCD FLASH memory from 36 screens down to 21 screens.

## Version 2.7.1

- This release is NOT for the DEMO. It is for our regular CPAP development stream, using a modified v4.5 board that accepts O2 sensor input.
- For the regular build we make the following modifications from the DEMO software
  - ◆ Enable Hi Pressure alarm
  - ◆ Enable Battery Alarm
  - ◆ Enable Occlusion Alarm
  - ◆ Enable Battery Monitor through Power Board (use opto-isolators for I2C protection)
  - ◆ For v4.5 board modifications:
    - O2 analog input on pin A7 (AIN7), O2 status input on pin B1 (AIN13), O2 calibrate pin on B0
    - Knob encoders changed from B0 and B1 to B5 and B6
  - ◆ Configure external interrupts for B5 and B6 to Interrupt 5 and Interrupt 6
  - ◆ Use 50-point averaging for O2 sensor and O2 status to smooth out readings
  - ◆ Set DC bias for prop valve control to “48” for Enfield valve. (We used “70” for SMC valve for demo.)

## Version 2.6.8

- DEMO BUILD:
  - ◆ BUG FIX FOR LED FLASHES: The LED’s were glitching brightly, briefly and intermittently. The fix is to load the PWM counter registers through a double-buffering technique (using “CCBUF”), instead of loading the counters directly (that caused a race condition). Verified this prevents the intermittent glitchy flashing. Leaving “pulsing” enabled now instead of flashing.

## Version 2.6.7

- DEMO BUILD:
  - ◆ NEW SCREENS: The rebuilt screens are controlled with this firmware build. They include all lock and mute screens for setpoint, ramp and run screens. (Needed to adjust the position of the battery graphic and the firmware version printout.)

- ◆ BUG ADDRESSED: Because the pulsing LED's cause glitches, just flash the LED's on and off. (Note this is a temporary solution.)
- ◆ WATCHDOG: Now disabled for development and demos. (Now you have time to calibrate the touch screen after loading new screens.)

## Version 2.6.6

- DEMO BUILD:
  - ◆ BUG FIX: Buzzer will sound as long as both unmuted and any alarm screen is present; buzzer won't silence until the alarm screen goes away
  - ◆ FEATURE CHANGE: Mute button now toggles mute condition. If mute is on, pressing mute button again disables mute.
  - ◆ FEATURE CHANGE: Instead of "pulsing" the LED's, the LED's will now flash on and off. (This is to avoid the glitchy "flash" that occasionally happens during pulsing.)
  - ◆ Disable watchdog for development and demo.

## Version 2.6.5

- DEMO BUILD:
  - ◆ BUG FIX: Include "baby temperature" on the "ramping" screen (it was completely missing before)
  - ◆ DISPLAY IMPROVEMENT: Include degrees symbol "°" after "baby temperature" so it shows for instance "38°" instead of "38".
  - ◆ BUG FIX: When power on self test completes, the "SELF TEST COMPLETE" text no longer overwrites the "SELF TEST STARTING" text.

## Version 2.6.4

- DEMO BUILD:
  - ◆ BUG FIX, KNOB ENCODERS: For the external IRQ peripheral, debouncer settings are frequency: Divided by 64 and Low Frequency Samples: 7. It looks like the encoder noise could be close to 1ms pulses, so setting the debouncer to about 2Khz. The glitching is almost gone now.

## Version 2.6.3

- DEMO BUILD:
  - ◆ BUG FIX, KNOB ENCODERS: Add firmware debouncing to both knob inputs. For the external IRQ peripheral, debouncer settings are frequency: Divided by 4 and Low Frequency Samples: 7. Also, add 2usec delay after switching MUX to read encoder (MUX delay is about 500 nsecs).

- ◆ For the tasks, instead of having them all multiples of 20msecs (so they all try to run at the same interval), make them odd multiples, like instead of running the state machine at 100msecs, run it at 101 msecs.
- ◆ The units “%O2” were touching the numbers, e.g. with “24” the 4 touched the “%” sign, so moved it over a bit

## Version 2.6.2

- DEMO BUILD:
  - ◆ NEW FONT FILE: Changed address where font file is stored (for large numbers on run screens and alarm screens). Now able to display large numbers correctly.
  - ◆ SWAPPING SCREENS: since there are different Alarm screens for muted, unmuted, locked, unlocked, the screens are now updated whenever the lock or alarm condition changes.

## Version 2.6.1

- DEMO BUILD:
  - ◆ NEW LCD SCREENS: This build requires new screens to run properly. The 27 new screens are from India, labeled Screen\_1 through Screen\_27, with about 6 screens still missing as of today.
  - ◆ Link to the screens I’m using:  
<https://drive.google.com/drive/folders/1UrHonXwzStTXon6qzlbzrp2qBs9QDQtI?usp=sharing>.

## Version 2.5.13

- DEMO BUILD:
  - ◆ BUG FIX for Pressure overshooting: fix for Pressure control with demo unit plumbing: change parameters for Baby Pressure PID loop, KP is reduced from 10 to 1, and KI is reduced from 5 to 3. This gives decent pressure response with no overshoot.
  - ◆ BUG FIX for POST failing. Make sure POST (Self Test) passes even if the thermocouples and circuit are all disconnected – the test only checks that the flow\_insp increases when the blower is on, and the pressure raw readings are between 800 and 1500, and the heater and PCB temperatures are normal. Previously, this test assumed the circuit was connected to pass.

## Version 2.5.12

- DEMO BUILD:
  - ◆ Fix for FAILING Flow test during Power on Self Test: the flow sensors weren’t being updated to track the blower in time, and the flow test was failing on startup. Also, flow\_ox

was sometimes returning a slightly negative number, which was failing. Flow test should always pass now

- ◆ Fix for alarm “beep” happening when adjusting the knobs quickly; the logic was letting a “Buzz” get through even before the 5 seconds had elapsed; now fixed.

## Version 2.5.11

- DEMO BUILD:
  - ◆ Fix for Knob Encoders jumping when spinning them too fast. The debounce filter in Atmel Studio IRQ configuration is now disabled. The problem was the “debouncing” circuit was filtering the signal SO much that the interrupt was delayed so much it was out of sync with the 2nd encoder output. This happened when spinning the knobs too fast.

## Version 2.5.10

- DEMO BUILD:
  - ◆ After 40 seconds in “RAMP” state (“Adjusting to New Setpoint”) the alarms are enabled. So if the pressure or FiO2 setpoint is never reached during ramp-up, the alarms will start after 45 seconds.
  - ◆ Fixes for resetting Flow Sensors (still not a reliable solution to I2C errors)
  - ◆ Limit “Watchdog Feed” to once every 100msecs
  - ◆ Increase resolution of LED pulsing – instead of using 0-100, use 0-10000 for smoother pulsing.
  - ◆ KNOWN BUG: LED pulsing sometimes shows a brief “flash” – very intermittent

## Version 2.5.9

- DEMO BUILD:
  - ◆ Make regular LED brightness (yellow and green) dimmer (20% of maximum instead of 50%).
  - ◆ Make alarm LED brightness still at 50%.

## Version 2.5.8

- DEMO BUILD:
  - ◆ Move “Back” button to lower left of screen, opposite “Forward” button; only on setup screens; icon for “Back” button is temporary pending new screens; old “Back” button is now masked off to be invisible
  - ◆ Allow “Mute” even when screen is locked (other buttons and encoders still locked)

- ◆ Improve knob encoder response (more debouncing of Encoder signal); still a very-intermittent problem with slowly rotating the knob causing a value jump.
- ◆ (Remove large amounts of unused code in main.c)

## Version 2.5.7

- DEMO BUILD:
  - ◆ Improve knob encoder response (use only rising edge of encoder pulse for translation); still a very-intermittent problem with slowly rotating the knob causing a value jump.
  - ◆ New “LOCK” button on the run screens to Lock the screen. When locked, the “LOCK” button will show “UNLOCK”, and the other buttons and knobs will be ignored. Only the “UNLOCK” button can unlock the screen. Eventually, we will get an icon for instead of just the words “LOCK” or “UNLOCK”.
  - ◆ The “RE\_CAL” button on the first run screen is now in button position “1”, and button position “2” is reserved for the “LOCK” button.

## Version 2.5.6

- DEMO BUILD:
  - ◆ Disable “High Pressure” alarm for DEMO unit (compiled out)
  - ◆ Ability to abort “Power on Self Test” by clicking “next”

## Version 2.5.5

- DEMO BUILD:
  - ◆ Mute button is active for each screen, pushing will mute all buzzes for 30 seconds.
  - ◆ Mute button disappears from screen when “mute” is active; reappears in 30 seconds
  - ◆ During Splash and POST, LED’s will be “bright blue”
  - ◆ FiO2 actual display will be rounded to 1% (instead of rounding to 5%); update rate for numbers is 2 times per second

## Version 2.5.4

- DEMO BUILD:
  - ◆ For Ramping and for Alarm states, the LED’s will “pulse”, meaning ramp up and down in a sinusoidal fashion, instead of blinking.
  - ◆ FiO2 prop valve tuning:

- Reduce the propvalve ramp\_rate from 0.5 to 0.2 to get better control
- Start the loop at prop valve “70” instead of “0” to get to the setpoint sooner.

## Version 2.5.3

- DEMO BUILD:
  - ◆ Mute button silences alarm for 30 seconds (instead of forever), then alarm comes back on.
  - ◆ Knobs can now be set with “pressSetpt()” and “fio2Setpt()”, so they can be started at a known value instead of “minimum”. Default on settings are 5cmH2O and 21%O2.
  - ◆ Low pressure alarm is 2.0cmH2O below setpoint for 5 seconds. Cancel low pressure alarm when above 1.5cmH2O below setpoint for 5 seconds.
  - ◆ Low FiO2 alarm is 10% below setpoint for 10 seconds.
  - ◆ Cancel low FiO2 alarm when above 7% below setpoint for 5 seconds.

## Version 2.5.2

- DEMO BUILD: Includes
  - ◆ No O2 sensor capability or checking
  - ◆ Encoder knobs working using PB00 and PB01
  - ◆ LED Color styles implemented, including blue on setup, pulsing green on ramping, solid green on steady state running, red flashing on alarms, and yellow on pause
  - ◆ Occlusion alarm disabled
  - ◆ Mute button implemented (currently mutes permanently for each alarm)
  - ◆ Heat plate now servoing on “35 degrees” for demo safety (not 74 degrees)
  - ◆ On Power Up Self test, which is now enabled on every power up, use battery voltage instead of O2 report – Self Test always passes

## Version 2.4.14

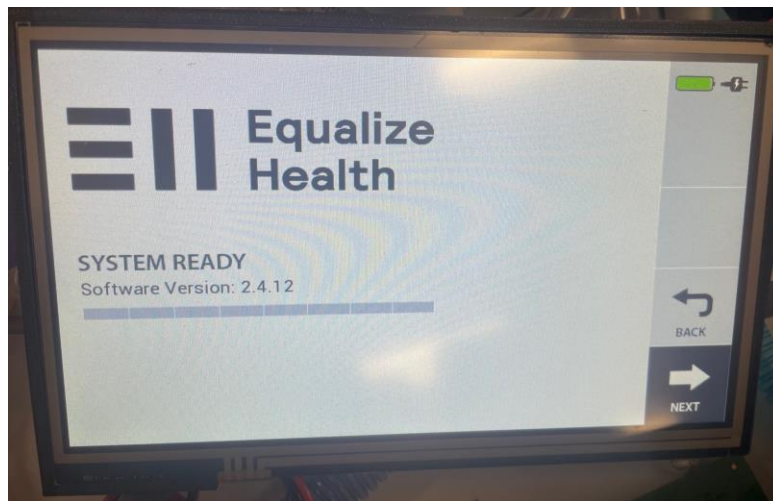
- BATTERY MONITOR DISABLED: disable the I2C to the Power board to monitor battery level and AC plugged in because of the I2C errors. The battery level and AC power status are now invalid
- I2C ERROR MONITORING: add printouts if there is I2C bus contention

## Version 2.4.13

- ENCODER KNOBS: now working. Can be enabled with “toggleKnobControl” and they will become live and control FiO2 and Baby Pressure.

- ◆ Must enable Interrupt 5 settings and Interrupt 6 settings in Atmel Start (for EXTERNAL INTERRUPTS) since new pinouts has PB05 using interrupt 5 and PB06 using interrupt 6. (Had to re-generate project in Atmel Start).
- ◆ Make sure FiO2 knob goes in even 5% increments (e.g. 21, 25, 30, 35, ..., instead of 21, 26, 31, 36, etc...)
- ◆ Be sure the mux switch common SW\_COMMON\_PIN is PB23, not PB07 (lingering mistake from using O2 Modbus control and changed SW\_COMMON to PB07 for that)

## Version 2.4.12

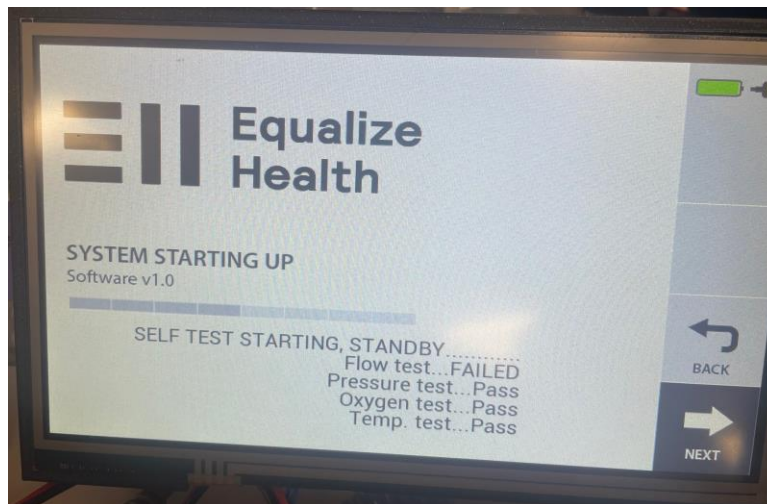


- LCD SCREENS: Now display software version on display after power up.
- KNOBS: Fix encoder pin assignment. (Was PB00 and PB01, should be PB05 and PB06)
- FIO2 ALARM: disable for 8 seconds after changing FiO2 setpoint
- Change initial FiO2 ramp rate from 1.0 to 2.0 to speed up initial ramp up.
- FIO2 LOOP: disable loop if flow sensors are giving bad readings (e.g. when I2C errors are happening)

## Version 2.4.11

- POWER ON SELFTEST: Not compiled in yet. This self test will execute after every power up or reset, and the self test results will display on the LCD screen when this is eventually compiled in. It runs the blower at “blower(250)” and checks the 2 flow sensors, the 2 pressure sensors, the oxygen sensor, and the temperature sensors of the heat plate, and the temp\_insp and temp\_exp.





- **FIO2 FILTERING:** To make the screen display more reasonable, a new heavily-filtered version of the FiO2 reading is used. This filtered version is also used for alarms.
- **FIO2 CALIBRATING:** With the faster prop valve controls, the delay before starting the O2 calibration is increased from 10 seconds to 15 seconds to let the O2 sensor catch up.
- **STACK OVERFLOW:** A stack overflow was seen on the State Machine task on one of the test platforms, so the stack for the State Machine task is increased by 2x.

## Version 2.4.10

- **FIO2 CONTROL:** New ramping function for FiO2 from Costa. Controls are “pvramp()” to set the number of counts per increment of ramp and “fio2tol()” to set the FiO2 tolerance band. Current defaults are “pvramp(0.5)” and “fio2tol(0.5)”.
- **OCCLUSION ALARM INHIBIT:** Per Robin, the occlusion alarm will be inhibited when the blower control has increased > 10 counts in the last 10 seconds. Once the blower is no longer increasing, the occlusion alarm will be active. This is necessary in the example where a hose is completely disconnected and the blower ramps up a lot, then reconnected and the blower comes back down. Without this inhibit, the occlusion alarm gives false alarms.
- **MULTIPLE ALARMS:** Since many of the alarms, e.g. FiO2 and occlusion or FiO2 and Low Pressure occur at the same time, we should handle simultaneous alarms.
- **LCD SCREENS:** Add button “RE-CAL” to basic run screen (button 2 position), so user can re-run the circuit calibration. This is only accessible from the initial “Run” screen where all systems are off. Also, numbers on alarm screens (e.g. baby pressure and FiO2) are now active and updating.

## Version 2.4.9

- **LCD SCREENS:** Make all fields active on alarm screens, including FiO2 and Baby Pressure actual and setpoints, temperature and battery icon

- LCD SCREENS: LCD screens are now active on this build; if the LCD is missing, the code will flag that and continue, if the LCD is present it will be active and usable
- I2C ERRORS: Instead of flooding the screen, the I2C errors will display intermittently, so the terminal screen doesn't flood
- CONFIG FILES FOR MPLAB IDE: config files needed to use MPLAB IDE tool, and to use J-Link debugger

## Version 2.4.8

- I2C ERRORS: Remove the board reset function after getting 100 I2C errors. Now the board will never reset, no matter how many errors occur on the I2C bus.
- LCD SCREENS: The "Power Connected" icon is now active: if AC power is connected, the icon will display, if AC power is removed, the "Power Connected" icon will disappear, and only the battery indication will display.

## Version 2.4.7

- BATTERY MONITOR: Set limits for battery voltage:
  - ◆ Below 23.0 volts, declare an "attention" area (color code Yellow)
  - ◆ Below 21.0 volts, declare a "battery warning" area (color code Red)
  - ◆ These colors and codes will show up when using LCD display
- LCD DISPLAY: Fix baby pressure and FiO2 displays to look clean
- LCD DISPLAY: Show baby temperature in real time on screen
- NOTE: LCD display is disabled by default, until everyone has a v4.5 board or above

## Version 2.4.6

- HEATER CONTROLS: Heat plate temperature limit changed from 79 to 74, per Robin and Costa. The firmware shouldn't let the heat plate above 74, there will be a mechanical shutoff somewhere around 80.

## Version 2.4.5

- HEATER CONTROLS: Implementation of Robin's heater requirements:
  - ◆ Heat wire should servo on t\_prox, setting at 39 degrees
  - ◆ Heat Plate should servo on t\_dist, setting at 37 degrees

- Heat plate shouldn't exceed 79 degrees, so while ramping up the heat plate, we will actually servo on the heat plate thermistor at "79 degrees", and when we're within 2 degrees of 37 on t\_dist, we'll switch over and servo on t\_dist at 37 degrees.
- Change "ki" for heat wire from "0" to "1". We were never actually reaching 39 because the integration term was 0.
- Remove printouts of "Prev\_flow: .. prev\_p\_exp:" used for Occlusion alarm debugging
- Fixes for FLASH behavior: make sure we release the semaphore before doing "flash\_wait\_busy()", because it needs semaphore, too. Also, use semaphore in "flash\_erase\_sector()", since it was missing.

## Version 2.4.4

- OCCLUSION ALARM: Implementation of Tanner's occlusion conditions:  
The occlusion alarm should sound while both of the following are true:
  1. The average inspiratory flow decreases by a minimum of 1 LPM over the course of 3 seconds and remains there for at least 5 seconds
  2. The % difference between the inspiratory and expiratory pressure decreases by a minimum of 10% over the course of 3 seconds and remains there for at least 5 seconds.
- NOTES: When the alarm is met, the message appears "Got Occlusion Indication wait 1 seconds" and the red LED's will flash if connected. There is a 5 second delay to "kill" the occlusion alarm, when this happens the message appears "Alarm ending delay complete". There are numerous debug statements on the terminal meant to help debugging the occlusion alarm which will be removed later. If there is a problem, recording all the debug statements will help in trouble-shooting.

## Version 2.4.3

- FiO2 CALIBRATION (to O2 sensor) CHANGES:
  - ◆ Allow any calibration factor from 21 to 100 (before, I was limiting it to 80-120%).
  - ◆ Redo the FiO2 calibration whenever the user changes the FiO2 setpoint. Wait 10 seconds after the FiO2 setpoint is reached by the PID loop, then run the FiO2 calibration with the O2 sensor. The O2 sensor takes > 5 seconds to catch up to the FiO2 readings
  - ◆ Requirements to do FiO2 calibration:
    - 1.1 FiO2 actual is within 0.4 of FiO2 setpoint
    - 1.2 10 seconds elapsed since last FiO2 setpoint change
    - 1.3 Blower > 200
    - 1.4 O2 sensor reading > 20.0 (when warming up, the O2 sensor reads typically 1 or 2)
    - 1.5 flow\_ox > 0.35

- ◆ When the O2 calibration factor is found, limit all further FiO2 settings to that factor. For example, if the calibration factor is 80, then don't allow any FiO2 setting above 80.
- I2C ERRORS: Only keep track of I2C errors from the Flow sensor board. I2C errors from the Flow sensors will generate a board reset when they total 100. But I2C errors from the battery monitor or from the power monitor or from the blower will NOT generate errors, no matter how many.
- FIO2 PID LOOP: Use a bias of 4100 for the prop valve instead of 3000. At low O2 tank pressure, the prop valve has no effect below about 4500 setting.

## Version 2.4.2

- NEW FEATURE: Set the PID maximum for each of the 4 PID loops using "kmax(x,value)". For instance, to set the heat\_plate maximum to 10000 for the PID loop, use "kmax(2,10000)", where: `x=0:fiO2,x=1:press,x=2:heat_plate,x=3:heat_wire`. If you do a "saveAll" after entering the maximum, the value you entered is saved to FLASH.
- BUG FIX: the blower limit is now held for 5 seconds after the flow condition that caused the limit is removed. Also the blower "integral" term is limited, so when the condition is active, the integral does NOT increase. Both of these together should ensure that the blower should not increase beyond whatever the blower setting is at when the flow condition is sensed.
- BATTERY MONITOR ON POWER BOARD: is now enabled on the I2C bus. If the cable to the Power board (connecting to P12 on the main board) is missing, this will generate a stream of I2C errors.

## Version 2.4.1

- NEW FEATURE: Watchdog implemented. If code hangs up for more than 4 seconds, the "watchdog" should bite and reset the processor. The watchdog is fed in the "Idle Task", which only runs when all the other tasks are not running. (Terminal command "triggerWatchdog" will put the code into an unrecoverable loop, and the watchdog should bite and reset the board after about 4 seconds.)
- NEW FEATURE: Battery backup of RTC. The time should survive power cycles after the watch battery is in place and the "BOD" fuses on the MCU are set. (In the Microship Studio project, go to "Device Programming", click "Apply", select "Fuses" and change "BOD33\_DIS" to unselected, and change "BOD33\_ACTION" to "The BOD33 puts the device in backup sleep mode". Then click "Program" and the battery backup should work.)
- BUG FIX: the blower speed wasn't updating normally, so remove the "return early" code.

## Version 2.3.13

- BUG FIX: Zero pressure sensors works now. There was a conflict between different measurements of the ADC, so there were intermittent bad readings causing the problem

(reading other ADC channels when trying to read pressure). Need to put the “semaphoreTake()” BEFORE the set\_adc(channel) selection, not after.

- Remove “Baby Pressure Offset”, which was the offset (usually about -0.3) that Baby Pressure showed with the blower off. Now, this will not be used, and the Baby Pressure value will be straight from the equations.
- O2 SENSOR: O2 calibration will not be allowed until the O2 Sensor has been on with status “good” for 10 minutes. The 10 minute timer starts with the O2 sensor status is OK (status 1) and the O2 reading is greater than 5 (for the first 1-2 minutes, the O2 reading is usually 1 or 2).
- BUG FIX: getting “Timer Task” Stack overflow errors, increase stack size from 64 to 100.
- TIMING: reduce the “sample\_adc(50)” to “sample\_adc(20)”. I don’t believe that a large sampling of the ADC is needed.
- I2C ERROR IMPROVEMENTS: when reading the Flow sensors over I2C, do a “ENTER\_CRITICAL()” function to prevent task swapping or RTOS interference. Increase delays in “flow\_reset\_sensor” to put 200msecs after releasing the RESET line for the Flow sensor power.
- CALIBRATE BUG: when “calibrate” starts, disable the FiO2 and Baby Pressure PID loops. Previously, if calibrate was attempted while under PID control, the PID loop would try to keep taking over from the calibration task.

## Version 2.3.12

- DEBUG PROP VALVE: Set DAC0 (pin PA02) for every update to the Prop Valve. PWM is still running (pin PA12), so select desired control signal using jumper on JP1. DAC0 is selected by jumping pins JP1-1. PWM is selected by jumping JP1-2. (See silkscreen on back of board.)
- BUG FIX: In the Flow Sensor reset logic, hold reset for 160msecs instead of 80, and after resetting, delay 250 msecs before resuming I2C operations.
- I2C ERROR RESET: If >100 I2C errors are recorded, then consider that the flow sensors are unrecoverable and do a soft MCU reset.
- DEBUG FREERTOS: Collect RTOS task statistics, display with new command “runTimeStats”. New timer “TIMER\_1” runs at 20Khz to collect task statistics.

## Version 2.3.11

- BUG FIX: FiO2 will calibrate to the O2 sensor if the flow\_ox is > 0.3. It was previously set to calibrate only if flow\_ox > 2.0, which meant at low FiO2 settings it never calibrated.
- FiO2 CONTROL LOOP: change bias of PID loop to be FIO2\_MIN of 0 (instead of 50), FIO2\_MAX is 75 (instead of 95) and FIO2\_BIAS is 30 (instead of 63). (FIO2\_BIAS is where the control loop starts hunting)

## Version 2.3.10

- FEATURE CHANGE: FiO2 will now calibrate to the O2 sensor if the O2 sensor reading is > 25 (changed from > 60).
- BUG FIX: programming “.bat” file now working again

## Version 2.3.9

- FEATURE CHANGE: The battery monitor on the Power board is disabled in firmware. To remove electrically from the circuit, disconnect cable from P12 (disconnects I2C). Should improve I2C error problem.
- FEATURE CHANGE: New equations for press\_baby, press\_ckt\_exp, press\_ckt\_insp and press\_ckt.value, calculate c\_leak and c\_insp in "calibration" when c\_factor is calculated.

```
press_baby.value = press_ckt.value - (c_leak * flow_leak.value * flow_leak.value);
```

```
flow_leak.value = flow_insp.avg - flow_exp.avg;
```

```
float press_ckt_exp = press_exp.avg + (0.00218 * flow_exp.avg * flow_exp.avg) + (0.0623 * flow_exp.avg) - 0.0817;
```

```
float press_ckt_insp = press_insp.avg - (c_insp * flow_leak.value * flow_leak.value);
```

```
press_ckt.value = (press_ckt_exp + press_ckt_insp) / 2.0;
```

*MUST BE DONE AS PART OF "CALIBRATION" TO CALCULATE c\_Leak AND c\_insp*

```
calx->press_ckt_exp = calx->press_exp = (0.00218 * flow_exp.avg * flow_exp.avg);
```

```
calx->c_leak = calx->press_ckt_exp / (calx->flow_leak * calx->flow_leak);
```

```
calx->c_insp = (calx->press_insp - calx->press_ckt_exp) / (calx->flow_insp * calx->flow_insp);
```

- In "configASSERT()" print out error message.
- Increase FreeRTOS heap size from 19200 to 24000
- Create binary semaphore for ADC. Prevents different consumers of ADC (on different threads) from interfering with channel selection of the ADC.
- FILTERING CHANGE: Increase standard Flow averaging to 10 (instead of 5); increase standard Pressure averaging to 10 (instead of 6).

## Version 2.3.8

- FEATURE CHANGE: O2 analog sensor enabled (use command “o2Status” to read out values)
- O2 CALIBRATION IMPROVEMENT: To toggle the “O2 Calibration” output, which must be high for 1 second, change the code to increase the width of the pulse:

```
gpio_set_pin_level(O2_CALIBRATE_PIN, true);
```

```
os_delay(2100); // was os_delay(1100)
```

```
gpio_set_pin_level(O2_CALIBRATE_PIN, false);
```

- O2 CALIBRATION PROTECTION: When calibrating the FiO2 to the O2 sensor every 10 minutes, put restrictions on to prevent bad calibration:

- `flox_ox` > 2.0
- O2 reading > 60
- blower speed > 2000
- FiO2 within 2% of setpoint in PID loop
- Resulting calibration must keep factor in range  $80 < o2\_factor < 120$ .

## Version 2.3.7

- BUG IMPROVEMENT: Only do averaging of 6 samples for `p_insp` and `p_exp` for regular measurements and PID loops, but during `c_factor` calibration and zeroing pressure sensors do 30-sample averaging for smoother, more accurate results.
- BUG IMPROVEMENT: For I2C errors, When reading I2C for flow, if the result is an error, try waiting 400usec and turn on the flow sensor measurements again (command 0x1000 to the SFM3400)

## Version 2.3.6

- BUG IMPROVEMENT: For I2C errors, Put 2usec delay in reading flow sensors (I2C address 0x40).
- Add Flow Sensor soft reset in "init\_flow" for resetting I2C errors.
- Add blower control limits during `press_baby` PID loop: If `p_limit` > `setpt+10` (per Robin's equation), try to limit blower speed (not tested).
- Put PID "max" value into "pidDump" for debugging blower speed limits
- New "delay\_usec(n)" function in `utils_user.c`.

## Version 2.3.5

- BUG FIX: Working LCD screens and touch buttons with Rev. 4.5 main board. SPI bus working with LCD and with external FLASH, nor FLASH corruption problems detected. New SPI bus semaphore so different RTOS tasks can share the SPI bus peacefully. Big and small numbers now updating on screens. NOTE: LCD screens DISABLED in current software to be backwards compatible with old boards.
- BUG PROGRESS: Getting I2C bus errors (scrolling off screen on terminal). Fix for now is the reset flow sensor board and reset I2C bus on main processor, whenever more than 8 I2C errors are detected. Increase I2C Bus semaphore timeout from 40 to 209.
- IMPROVEMENT: Remove FiO2 averaging (per CG). Change pressure PID coefficients to:  $P=10, I=5, D=0$ . Change FiO2 coefficients to:  $P=1.0, I=1.0, D=1.0$  for new Enfield

proportional valve. Change flow averaging from 50 samples to 5. Change pressure averaging from 50 samples to 3. Change FiO2 averaging from 50 samples to 5. (per CG)

- BUG FIX: Encoding knobs now working. To make them “active” for the setpoints, you must execute the “toggleKnobControl” command, then the knobs directly control pressure and FiO2. In code, “Knob1” is replaced with “Knob\_fio2”, “Knob2” is replaced with “Knob\_press”.
- BUG FIX: Serial number for boards should now be stored and retrieved correctly

## Version 2.3.4

- BUG FIX: in c\_factor calculation, make flow\_leak = flow\_insp-flow\_exp. In all previous firmware versions the bad equation was:

- `calx->flow_leak = get_flow_insp() - get_flow_ox();`
- Corrected equation:
  - `calx->flow_leak = get_flow_insp() - get_flow_exp();`

(NOTE: this bug fix is only for the c\_factor calculation during calibration; “flow\_leak” was calculated correctly in the regular program flow.)

## Version 2.3.3

- NEW FEATURE: New operating system “FreeRTOS” implemented. Tasks all running on separate threads, scheduled by RTOS so each runs for an equal part of the time, unless it is suspended. All tasks on equal priority except “Alarm” thread at higher priority. No statistics implemented yet.
- NEW FEATURE: Alarms now implemented:

Implement 4 alarms:

- 1) low pressure ( $\text{press\_baby} < \text{setpt} - 1.5$ )
- 2) high pressure ( $\text{press\_baby} > \text{setpt} + 1.5$  OR  
 $\text{press\_baby} > 20$  OR  
 $\text{p\_insp}$  almost equal  $\text{p\_exp}$ )
- 3) low FiO2 ( $\text{fio2} < \text{setpt} - 7$ )
- 4) low battery ( $\text{vBatt} < 20$ )

All alarms have a 5 second delay before firing. All alarms have a 3 second delay before cancelling.

All pressure alarms are only active during "Run" state when the pressure is servoing under PID control loops, not during setup or ramping (alarms are disabled during setup and ramping, except battery alarm is always active)

- BUG FIX: Battery voltage now sent up to Host App



## Version 2.2.11

- NEW FEATURE: Battery monitor on Power board is now communicating with the main CPU. Using the “battery” command we can now display current, voltage, temperature and charge:  
Batt. voltage: 25.13V, current: 0.23ma, charge: 95.9%, Temp: 27.3degC  
Battery registers are being read once per second, then converted to floating point numbers as needed.
- BUG FIX: When hitting the “Exit” button on the “Low Battery” alarm screen, return to the “Run” screen.

## Version 2.2.10

- BUG FIX: When entering kp, kd, and ki parameters (e.g. kp(1,3.4) for baby pressure PID proportional value), the values are setup for FLASH storage. If the command "saveAll" is sent before rebooting, the values are stored in FLASH. The values entered in kp(), kd() and ki() will be used even after reboots or power cycles, as long as the “saveAll” command was entered.

## Version 2.2.9

- NEW FEATURE: By enabling the Oxygen Sensor (setting O2\_SENSOR to “1” in main.h) the Modbus O2 sensor interface is enabled over the USART\_0 port, using pins PA22 and PA23. The O2\_SENSOR = 0 by default, so the O2 sensor is normally disabled and the Modbus is not configured by default.
- NEW FEATURE: When the oxygen sensor is enabled, the FlowWorks Hostapp will display O2 readings on the Pneumatic screen (requires v1.1.10 or above).

## Version 2.2.8

- BUG FIX: FiO2 calculation is unstable at low flows (low blower settings), so don’t calculate FiO2 until blower above 100.
- FEATURE: state machine cycles through available Alarm screens (using “Next Alarm” button) and coordinates with FlowWorks on Windows.
- FEATURE: add yellow, green and red LED’s to states “ramping up”, “running” and “alarms”.

## Version 2.2.7

- BUG FIX: Make FiO2 Control loop more responsive by changing PID parameters (using 50PSI pressure)
- When Calibration finishes, move state to “RUN” state so user doesn’t have to push “Next”
- INTERLOCK: Only allow FiO2 loop (Prop valve to open) when blower has started moving
- When FiO2, Baby Pressure are within limits (5% and 0.35cmH2O), move from “ADJUSTING TO NEW SETPOINT” screen to “Running” screen.

## Version 2.2.6

- BUG FIX: fix assignments of Pressure and FiO2 setpoints; use 3.0 and 21.0 as default (if nothing in FLASH), don't read setpoints from knobs unless knob selection is enabled.
- NEW BEHAVIOR: flow of states between "ready to run" to "ramping" to "running" now smooth; "Pause" button now shuts down system
- NEW BEHAVIOR: the "Self Test" screen will now start a "c\_factor\_calibration". The old behavior was to start a "power\_on\_self\_test", which was wrong.

## Version 2.2.5

- BUG FIX: when using the HeatPlate PID loop with maximum value of 10000, the standard board smokes. Reduce the PID maximum value to 2000 to prevent board damage.
- ENHANCMENT: handle states of "running" and "pause" in the state machine

## Version 2.2.4

- NEW BEHAVIOR: The output stream from "avgStream", will now include 5 extra parameters at the end: HeatWireSetpt, HeatWireActual, HeatWireControl, HeatWireError, HeatWireIntegral. (NOTE: this may be removed in an upcoming release)
- NEW BEHAVIOR: The 2 output streams "avgStream" and "pidStream" can now work together or separately. If both are requested, they will alternate (avgStream,pidStream,avgStream, etc.). The output rate can still be set with "avgStreamRate(hz)".
- FIX: the "Self Test" feature will no longer be "blocking". It will be run as a state machine, reentrant, updated every 20msecs, allowing all other normal functions to run at the same time.

## Version 2.2.3

- NEW FEATURE: Create state machine and LCD screen drivers, move through screens with button control.
- HOST SUPPORT: Host can change screen by sending button changes, e.g. "button(4)" for "Next", and Host can read back screen status to know which screen to display.
- BUG FIX: In the "avgStream" make sure that FiO2 is always less than or equal to "100".

## Version 2.2.2

- NEW FEATURE: Support Windows Host App (called "FlowWorks") by uploading data and status on demand to support extensive GUI interface showing sensors and controls.
- NOTE: FlowWorks on the laptop won't work with any firmware version less than 2.2.1.

## Version 2.2.1

- BUG FIX: heat wire PWM now working. Now all 6 PWM controls work together.

- Convert “calibration” and “wait for blower to stop” routines to all “re-entrant” code, so nothing stalls (hogs CPU) waiting for completion
- NEW FEATURE: sendStatus and sendData commands coming from Host App “FlowWorks” now supported

## Version 2.1.20

- FEATURE CHANGE: Max plate heater control value in the heat plate PID is increased from 5000 to 10000 to get the full 6A of current running to the heater plate when under PID control.
- FEATURE CHANGE: Now support knobs using encoder instead of potentiometer. Signals KNOB1 and KNOB2 will now generate interrupts (`irq_FIO2_PB00_encoder` and `irq_PRESS_PB01_encoder`), and SW1 and SW2 are connected to the other 2 encoder signals, which are combined to sense “CW” and “CCW” rotation pulses.
- FEATURE: The “power” command now will include the battery voltage of the on-board watch battery (read through the ADC)
- KNOWN LIMITATION: The “heat wire” PWM control is not working, so it is disconnected. The heat wire will NOT work in this release.

## Version 2.1.19

- FEATURE CHANGE: Instead of printing out the column headers in "avgStream" every 15 lines, only print them once on startup, and never again. Put a comma at the start of every line printed in avgStream to make Excel imports cleaner.
- Example Log from TeraTerm with timestamps and commas (no headers):

```
[2021-04-01 21:36:28.216] ,000.53,000.14,01.83,00.00,20.76,19.91,20.34,27.50,021.07,000.38,000.65,001.18
[2021-04-01 21:36:28.415] ,000.53,000.14,01.83,00.00,20.77,19.90,20.34,27.50,021.07,000.38,000.65,001.18
[2021-04-01 21:36:28.614] ,000.53,000.14,01.83,00.00,20.77,19.90,20.34,27.50,021.07,000.38,000.65,001.18
[2021-04-01 21:36:28.813] ,000.53,000.14,01.83,00.00,20.77,19.90,20.34,27.50,021.08,000.38,000.66,001.17
[2021-04-01 21:36:29.016] ,000.53,000.14,01.82,00.00,20.77,19.91,20.34,27.49,021.08,000.38,000.65,001.17
[2021-04-01 21:36:29.216] ,000.53,000.14,01.82,00.00,20.77,19.91,20.34,27.49,021.08,000.39,000.66,001.17
[2021-04-01 21:36:29.416] ,000.54,000.14,01.82,00.00,20.77,19.91,20.34,27.49,021.08,000.39,000.66,001.17
[2021-04-01 21:36:29.616] ,000.54,000.15,01.83,00.00,20.77,19.91,20.34,27.49,021.09,000.39,000.66,001.16
[2021-04-01 21:36:29.816] ,000.54,000.15,01.83,00.00,20.76,19.91,20.34,27.49,021.10,000.39,000.66,001.16
```

## Version 2.1.18

- BUG FIX: Make the interval in printouts in “avgStream” more precise. There was a bug where a 5Hz output rate (“avgStreamRate(5)”, the default) was resulting in 220msecs period instead of 200 msecs period. This is fixed, and should give more accurate intervals for all settings of avgStreamRate(x).

## Version 2.1.17

- IMPROVEMENT: Change the blower open loop startup accel from .22 to .57 to reduce startup lag. Change the closed loop accel from .04 to .77 to speed up blower speed changes. This

decreases the blower lag on startup from 7 seconds to 2 seconds, and speeds up the movement between blower speeds.

- INTERLOCK IMPROVEMENT: Add a feature to limit the blower output depending on the pressure setpoint. By measuring typical blower vs. baby pressure values, I created a zone of possible blower settings, and limit the output of the pressure PID loop to these values to prevent blower run-away when the circuit is opened.
- BUG FIX: the zero\_press\_baby calculation is fixed to work now
- INTERLOCK: if the temperature sensors are disconnected, the heater PID loops are disabled for both the heat plate and heat wire
- INTERLOCK: if the blower isn't running, the heat wire PID loop is disabled (need air moving through the tube to reach the temperature probe near the baby, or the heat wire could run away)
- FEATURE: the up-arrow will now go through the last 10 commands (the down-arrow moves the opposite way in the buffer)

## Version 2.1.16

---

- BUG FIX: if the temperature probe is removed, or the circuit is completely removed from the humidifier, while the baby pressure is being controlled by the PID loop, we will now just hold the blower at whatever value it was at until normal pressures are restored, then continue the PID loop. We use the baby pressure  $< -1.0$  as the indicator that the pressure is not valid to indicate we should NOT run the PID loop for baby pressure, just leave the blower at whatever value it was at.
- BUG FIX: as a backup to the previous fix, the PID integral term for baby pressure will be limited to 511. There were situations (like opening the circuit) where the integral term would keep increasing out of control, which means the PID loop took minutes to respond while the blower was at maximum and the integral term was greater than 3,000.

## Version 2.1.15

---

- BUG FIX: wait for blower speed to drop to 0 before continuing. Avoids problem of trying to zero pressure sensors before blower has a chance to drop to 0.
- LED strip color fully adjustable. Using command "leds(x,y,z)" the three colors R, G, B, can be adjusted, with the range in percent, 0-100. For instance bright white is "leds(100,100,100)", dim yellow is "leds(10,10,0)", medium blue is "leds(0,0,50)". The 3 LED controls are now controlled by PWM signals.

## Version 2.1.12

---

- Add "F\_LEAK" to avgStream printout (at the end)

- Add “zeroBabyPress” command so if the baby pressure at “no pressure” is -0.3, this command will figure out the offset and use that to make the calculation of baby pressure more accurate
- (Move the state machine from “main.c” to new module “states.c”)
- (Fix bug in “calibratePress” routine.)

## Version 2.1.9

---

- New repo using AtmelStart tool: EqualizeHealth\_CPAP
- RTC and date and time implemented (using internal 32kHz oscillator)
- Power monitoring:
  - Bus: 24.15V, shunt: 3.711mV, current mA: 74.22mA, power: 275.42mW
- Command “calibrateFiO2” figures out the bias in the Prop Valve control for a given O2 tank pressure, typically around 73% at 50PSI.
- Tuning of FiO2 PID loop for less overshoot

## Version 2.1.4

---

- New terminal interface
- Features:
  - Help screen, type “?” or “help”
  - Input ignores case (capitalizations don’t matter)
  - Arguments for commands are given within parentheses, e.g. “PressSetpt(2.5)” will interpret the argument as value 2.5 for the baby pressure setpoint.
  - “Backspace” and “up-arrow” supported, allows backing up to delete inputs, and up-arrow will return last command
  - Help screen shows firmware version (e.g. Version: 2.1.4) and Board Serial Number (e.g. B100104)
  - “rawStream” will output raw sensor readings at 50mS interval (e.g. 958 for pressure); inputting “rawStream” a 2<sup>nd</sup> time toggles the output off
  - Typing anything will momentarily halt the stream
  - “avgStream” will output converted sensor readings at 200mS interval (e.g. 1.57 for pressure)
  - All output devices can be driven directly:
    - blower(200) – runs the blower (open-loop) at 200 out of maximum 511

- propValve(50) – opens the O2 proportional valve at 50%
- heatPlate(5000) – turns on the heater for the humidity chamber at 5000 out of 10000
- heatWire(6000) – turns on the heater for the wire in the circuit at 6000 out of 10000
- For PID control, first set the setpoint for the 4 parameters:
  - FiO2Setpt(50) – sets the desired FiO2 reading to 50% oxygen (range 20 – 100)
  - heatPlateSetpt(50) – sets the desired heat plate temperature to 50 degrees C.
  - heatWireSetpt(30) – sets the desired circuit temperature to 30 degrees C.
  - PressSetpt(2.8) – sets the desired baby pressure to 2.8 cmH2O
- To engage active PID control:
  - togglePIDHeatPlate – will actively servo on the heatPlateSetpt
  - togglePIDHeatWire – will actively servo on the heatWireSetpt
  - togglePIDFiO2 – will actively servo on the FiO2Setpt using the proportional valve
  - togglePIDPress – will actively servo on the Baby Pressure using the blower
- “pidStream” will output the control and setpoint values for all the PID loops
- “pidDump” will output the PID loop values once, plus the PID constants: kp, ki, and kd.
- “showFlash” will show all the values that are primed to be stored in FLASH, or have been loaded from FLASH at startup
- “saveAll” will save the parameters in FLASH, e.g. c\_factor, pressure offsets (from zeroSensors), and setpoints
- “SerialNumber(B100104)” will set the board serial number to the argument (e.g. B100104) and will be permanently stored in FLASH after the command “saveAll”.
- Reboot – will allow the board to be reset from the terminal (e.g. to abort all PID functions)
- “calibrate” – runs the “zeroSensor” command to calculate the pressure sensor offsets at no pressure, then runs the blower at several different speeds and calculates c\_factor for the given disposable rig. These calibrated values will be permanently stored in FLASH with the “saveAll” command
-