ABCD Operating Manual

Index:

Specifications:

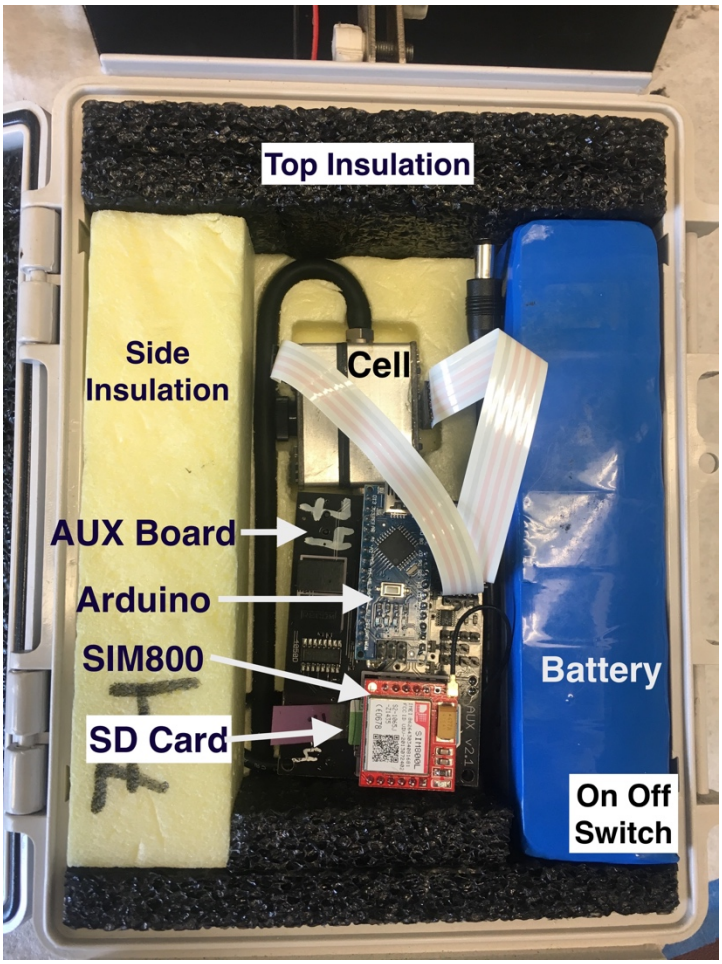
Specifications:
Operational Flowrate range: 100-300 cc/min
Spot size:
Filter Type: EMFAB
Filter Cut Punch: Number 4 hole punch

Components:
Microcontroller: Arduino Nano
Pump: Schwarzer SP-140FZ
FR Sensor: OMRON D6F 01A1 Mass Flow

Power:
Input Power: 9-12V
Power consumption: ~.6 W
Battery: 12V 10AH Li-Ion
Solar Panel: 18V 5W poly
Power Adapter: 12.6V charger with overcharge protection

Instrument schematics, and component nomenclature:

Schematics:



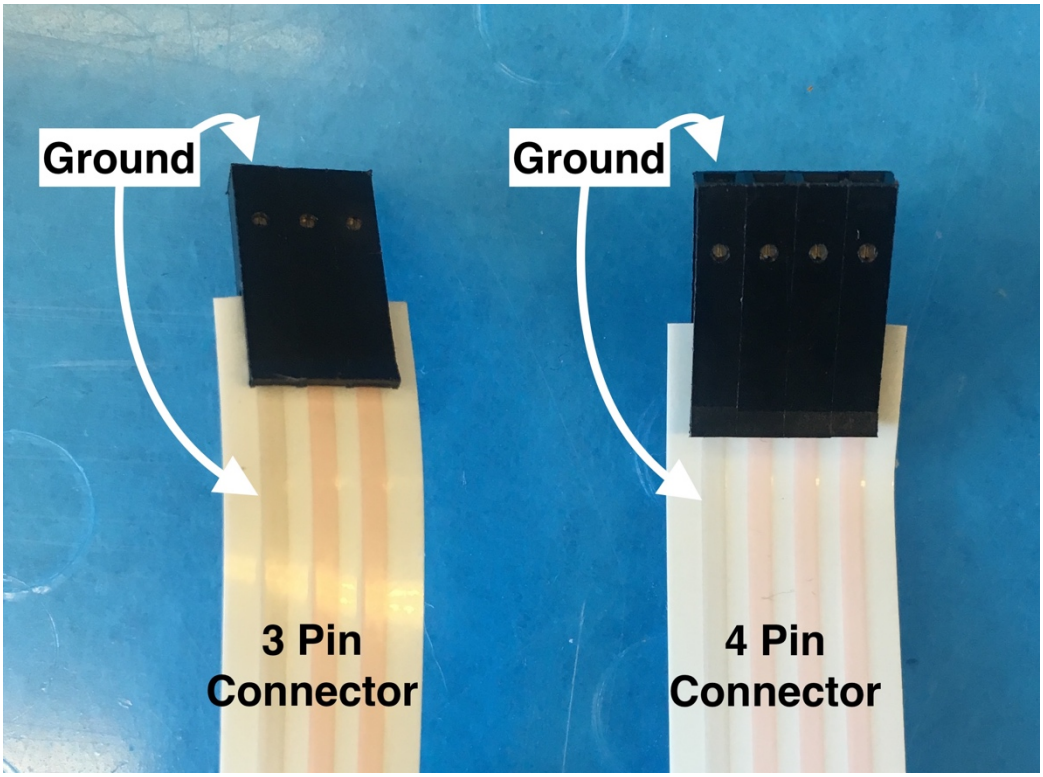


Figure 1: Cell, AUX Board, Arduino, SIM800, SD Card, Battery

Figure 1 shows the top view of the inside of the instrument. Labels are provided for main components and insulation sections.
Figure 1b shows the connector types used in the instrument. Ground is identified by the black line in the connector cable.

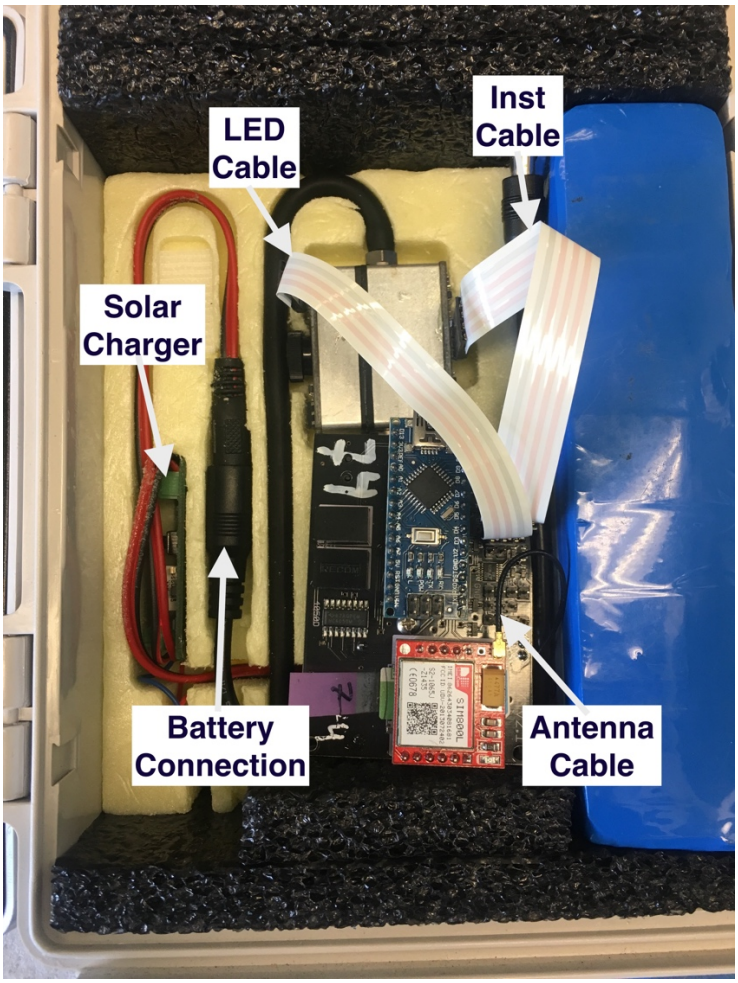


Figure 2: Solar Charger, Battery Connection, LED cable, Inst Cable, Antenna Cable

Figure 2 Shows the top view as shown in Figure 1, however the Side insulation is removed exposing the solar charger and power connections.

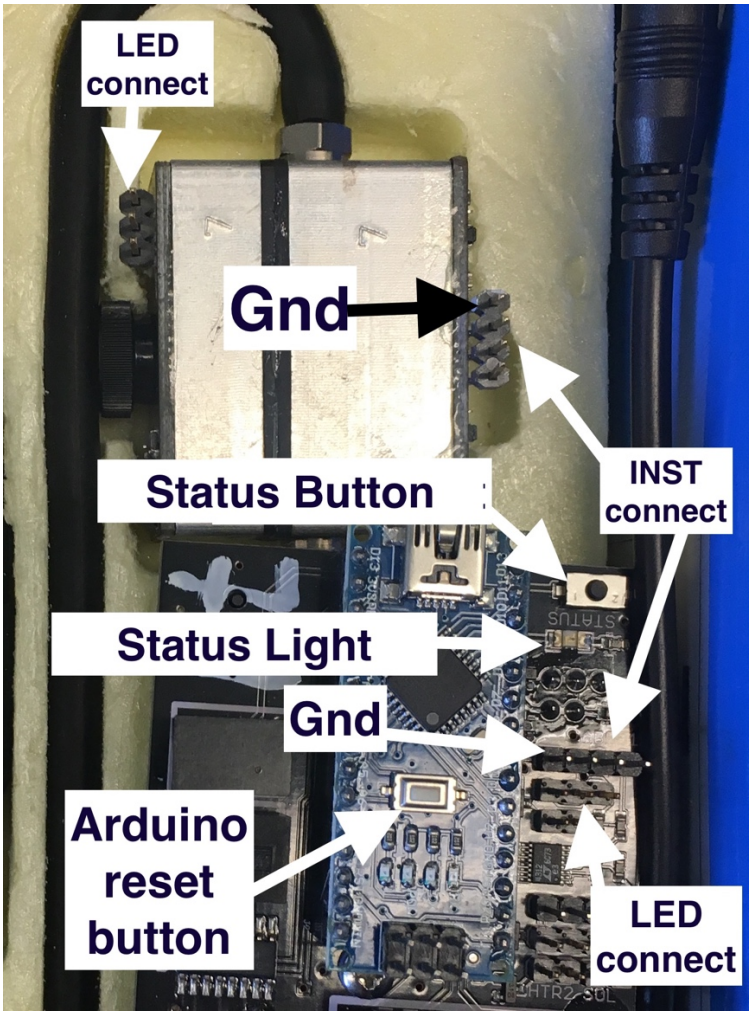


Figure 3: LED connect, INST connect, Status Button, Status Light, Reset button

Figure 3 focuses on the nomenclature of the Pinouts for the cell and Aux board. The Status Button and Status light are shown. The Arduino reset button is shown.
Connections are also shown here: Note that while the Instrument connection requires that the cables be set properly for the grounds to be oriented properly, the LED connection does not have a set orientation for ground since both of the outer pins are ground.

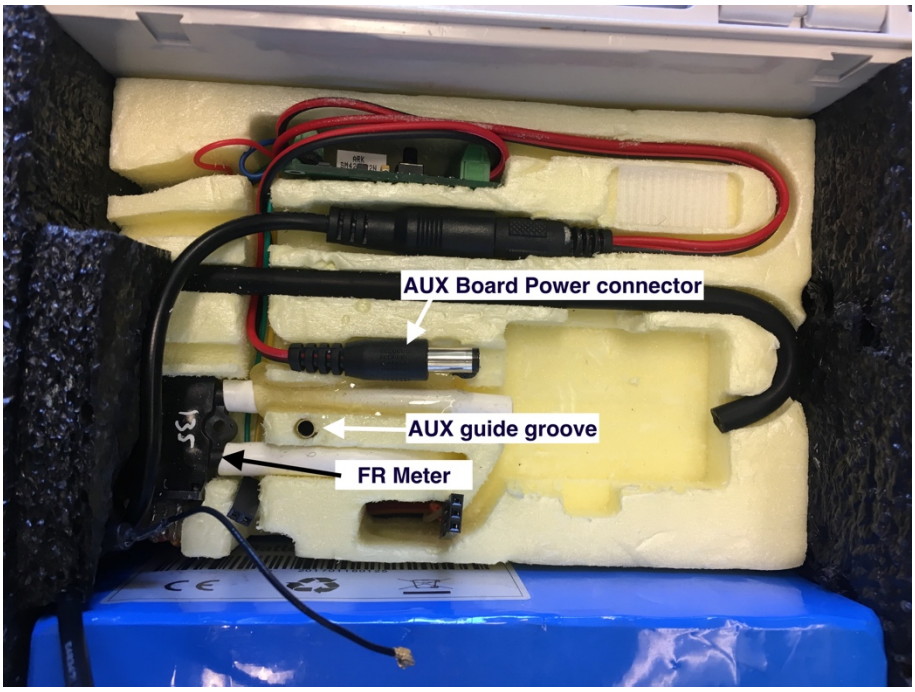


Figure 4: Aux Board Power connector, Aux guide Groove, FR Meter

Figure 4 shows the top view of the instrument with the Aux board, Cell, and Side insulation removed. The flow meter is exposed. The AUX guide groove is used to register the Aux board into place.

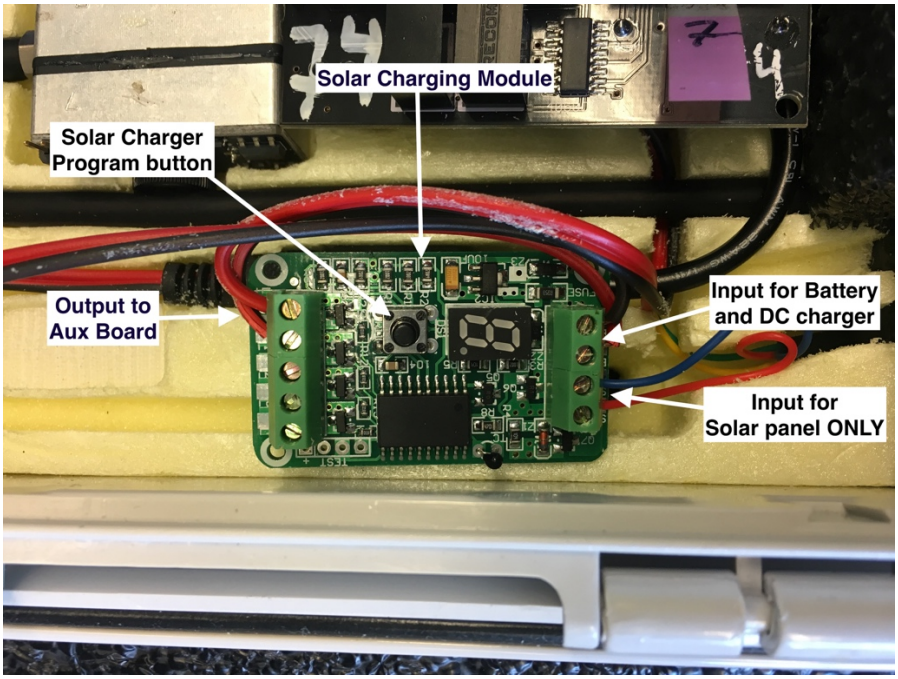


Figure 5: Solar Charging Module

Figure 5 show the Solar charging module pulled out of it's slot.

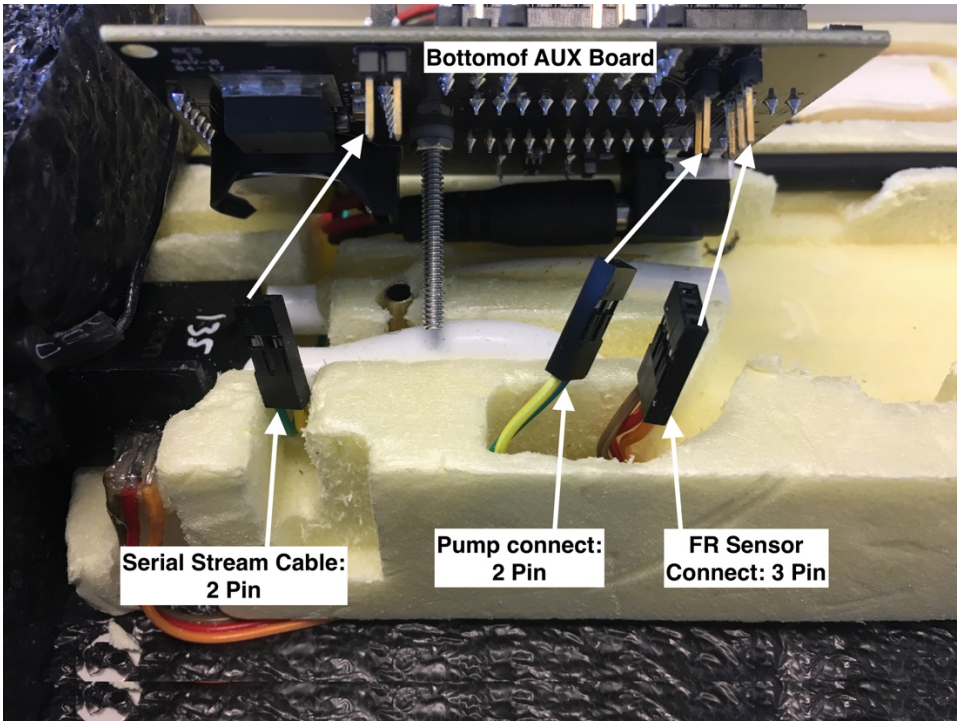


Figure 6: AUX Bottom connections: Serial cable, Pump cable, FR sensor cable

Figure 6 shows the bottom of the AUX Board and the bottom side connections. Note that the ground pins for the FR sensor and Pump are towards the middle of the AUX board.

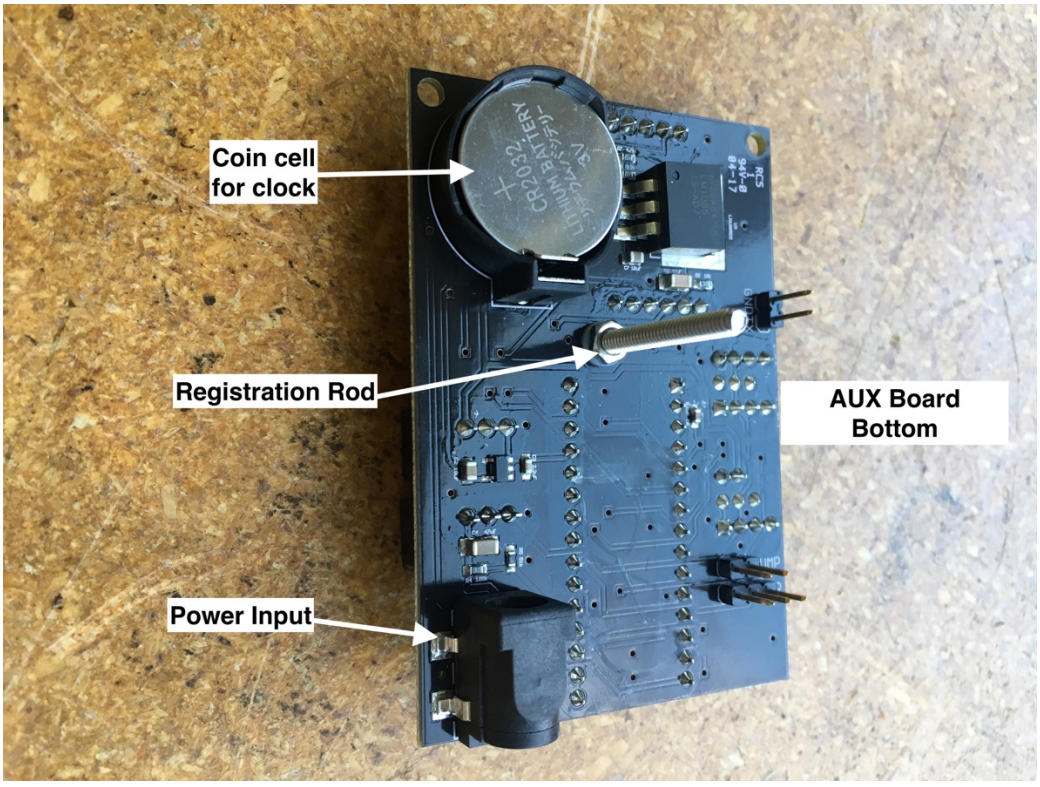


Figure 7: Aux Board Bottom

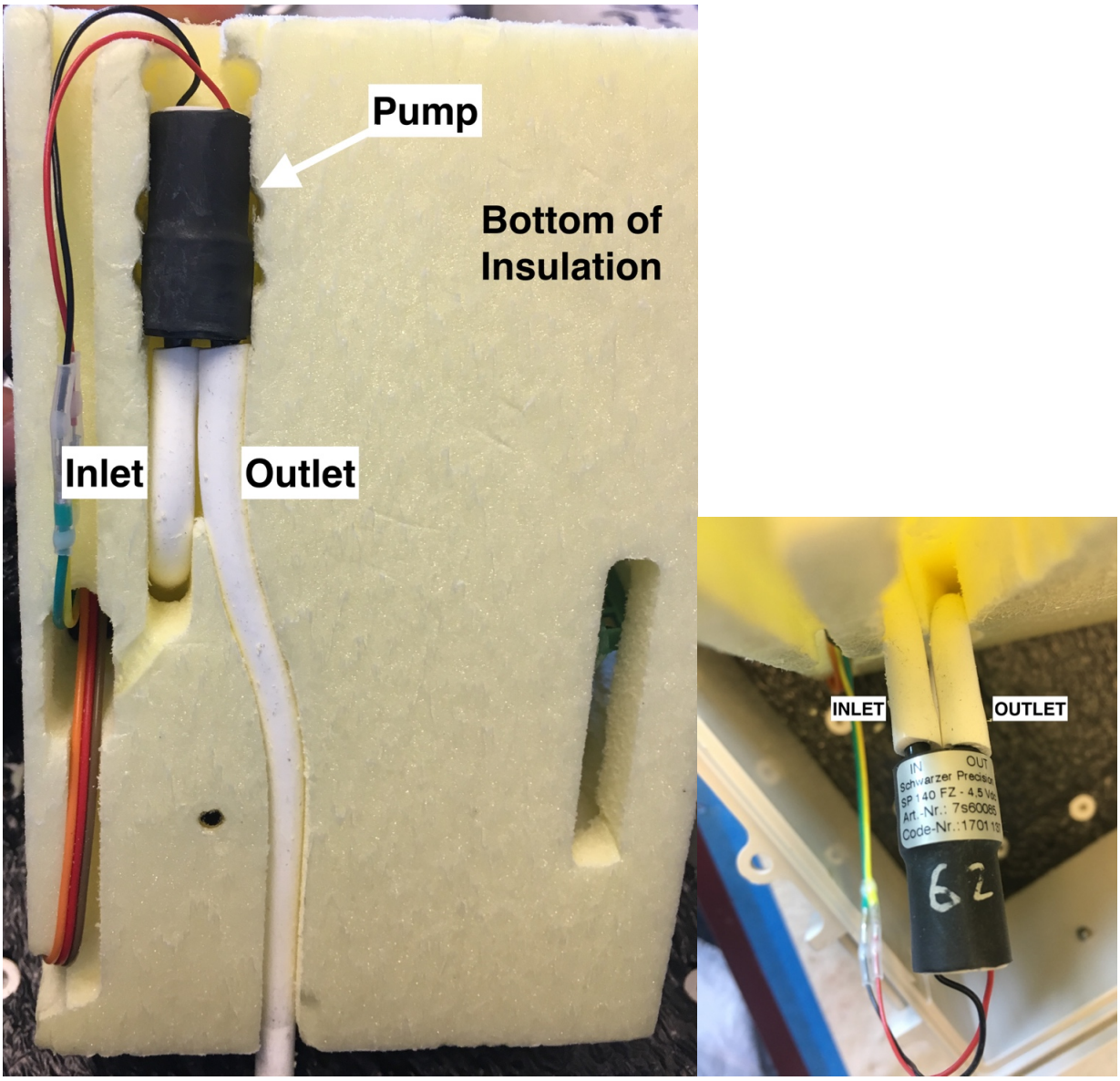


Figure 8: Bottom insulation: Pump, inlet, outlet

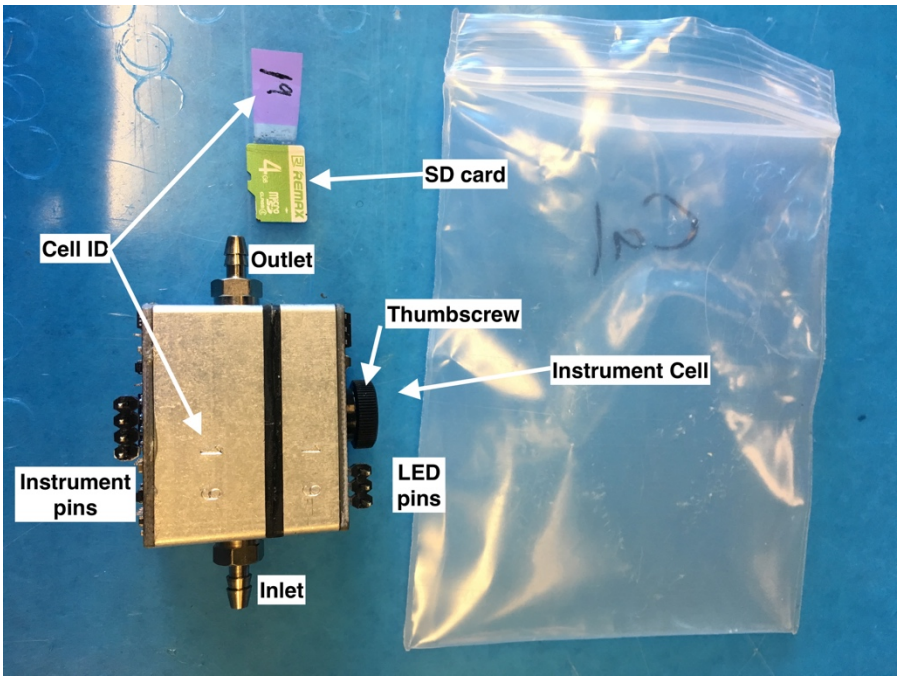


Figure 9: Cell

Buttons: UI buttons and switches available are the Status button, Arduino Reset button, Solar Charger module program button, and the Instrument on off switch.

Status button: (shown in Figure 3) is used to interact with the instrument while it is running. It is used differently for different codes, however in the main code it is used to reset attenuation, set the flip orientation of the cell, and parameter reset instrument parameters.

Arduino Reset Button: (shown in Figure 3) is used to reset the Arduino. This will restart the program it is running and revert to startup procedures.

Solar Charger program button: (shown in Figure 5) is used to program the power settings on the solar charging module.

On Off switch: (Shown in Figure 1) turn on or off the instrument: Note, for AC powered instruments, the power must be disconnected as well.

Connections:

Top Connections:

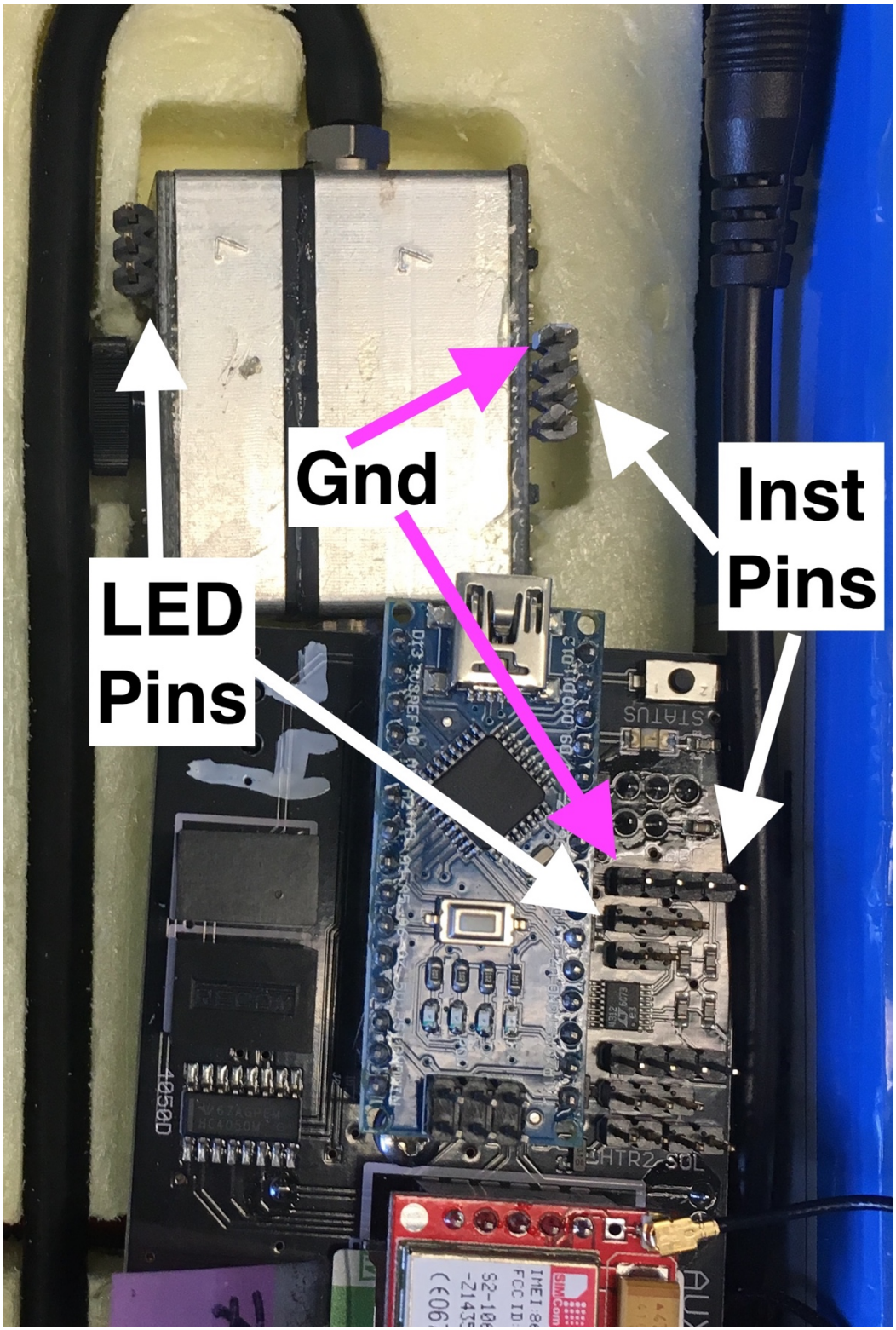


Figure 10: Top Connections

The Cell is connected from the Top. There are two connections, one for the instrument PCB and one for the LED PCB.
The LED connection (3 pins) does not have a set wiring orientation, just connect the cable from the AUX board to the LED PCB. There is no orientation on this connection since the two outer pins are both ground.
The Inst connection (4 pins) requires proper wiring orientation. Ensure that the ground from the aux pin is connected to the ground on the Inst PCB.

Bottom Connections:

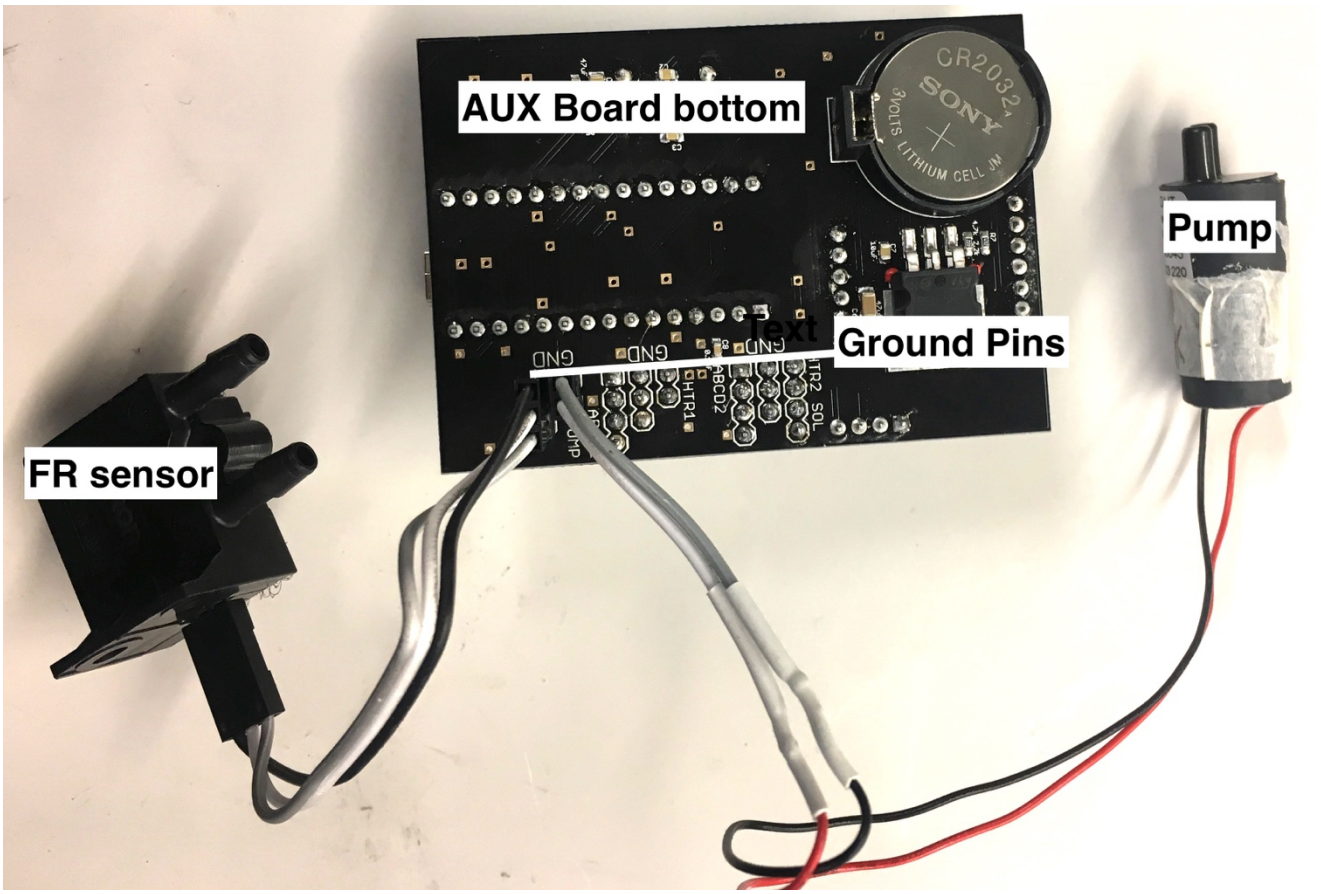


Figure 11: Bottom connections

The Pump and FR sensor, and Serial connections are connected to the bottom of the AUX board. The black wire on the cable is always the Ground cable. The ground connection on the AUX board is always towards the middle of the AUX board as shown in Figure 11.

Startup Procedures:

1) Ensure that the instrument is connected properly to the cell, ensure that an SD card is installed. Turn on instrument
2) The instrument Status light will blink 3 times to indicate that the instrument is turning on. After which the light should turn off and stay off.
    a. If the stats light stays red, or turns red at any point, it indicates an error exists that must be diagnosed. Common errors are due to improper connections or no SD card present.
    b. Diagnostic instructions can be found in the troubleshooting section of the manual.
3) When the instrument Arduino LED (on the Arduino board itself) begins flashing at the sample rate (Default 2 seconds), the instrument is operating normally.
    a. Startup generally requires about 10 seconds without network transmitting, and takes around 45 seconds with network transmitting.
4) If the cell has new filters, the ATN must be reset to 0: Hold down the status button until the status light turns on, and immediately release the status button. More detailed information on ATN resetting can be found in its own section in this manual.
    a. If the cell has new filters, but a different cell ID is being used, the instrument will detect a new cell and automatically reset ATN to 0.

Software:

Arduino Software:
Our instrument is Arduino based. It is the software used to program and upload firmware to the instrument, and also allows for easy monitoring of live serial data from the instrument. Please download the most current version of the Arduino for either OSx or Windows at: https://www.arduino.cc/en/Main/Software

Drivers:
In order to communicate with the instrument via USB, the driver for the Arduino must be installed on your computer. This is necessary if you intend to either change settings of the instrument or connect to collect a live stream from the instrument. (Note: for serial streaming via two-wire, driver is not necessary).

The Drivers are available in the “ABCD software package” in the folder CHdriver both for Windows® and OSx®

Libraries:
Before uploading firmware (required for changing major settings) to the instrument, it is required that the libraries are available to the Arduino software. To make them available, please copy and paste the contents in the folder “Libraries” from the “ABCD software package” into your Arduino library folder. After installation of the Arduino

software, by default, the Arduino library folder is located at:
/Documents/Arduino/libraries

Connecting to Arduino:

Connection to the Arduino can only be made after the Arduino Software and Drivers have been installed (as described above). Connect the Arduino to the computer using a USB mini cable. Open Arduino software.
Select COM port: In the toolbar of the Arduino software, navigate to tools → Port and select the proper port

For OSX, select "/dev/cu.wchusbserial1420"
For windows, it will be a COM port. The COM port may change, however you can figure it out through process of elimination.

Select board: In the toolbar of the Arduino software, navigate to tools → Board and select "Arduino Nano"
The Arduino is now properly connected to the computer

Serial Monitor:

Arduino Serial Monitor provides a live view of the serial data streaming from the Arduino. It also allows communication to the Arduino if the Arduino is programmed for it. Live data from the instrument will be broadcasted here.

Ensure that the Arduino is connected, and the port has been selected as instructed above before opening the Serial Monitor. To open the serial monitor, select the magnifying glass in the top right corner of the Arduino software interface (or open through the toolbar by navigating to tools → Serial Monitor). In the serial monitor, ensure that the baud rate (bottom right) is set to 57600 baud.

Uploading Code:

Before uploading code to the instrument ensure that you have already installed the Arduino software, Drivers, and Libraries onto your computer as detailed above. Open the code you want to upload by opening any .ino file in the code folder. Follow the above instructions to for "Connecting to Arduino." To upload code select the right pointing arrow at the top left of the Arduino software screen (or upload through the toolbar by navigating to Sketch → Upload).

There are 5 codes available for uploading to the instrument with the following purposes:

ABCD2.x.x_MainCode: This is the main operating firmware of the instrument. In the top of this code, there are user settings which can be adjusted to change flow rate, sampling periods, network settings ect. This is detailed in the "Instrument settings" section.

ABCD2.x.x_SetTime: This code is used to update the time on the Real Time Clock (RTC) of the instrument. The RTC is always powered from a coin cell battery, however sometimes the RTC can lose its time and require to be reset. An error on the instrument will alert you if this is necessary. This may also be useful if you are in a different timezone or need to adjust the instrument for daylight savings time. To use this code, upload the code onto the Arduino while holding down the STATUS button on the instrument. A message will be transmitted through the serial monitor indicating that the time has successfully been reset.

ABCD2.x.x_FRcal: Every instruments flow measurement inherently has deviations from each other due to each flow sensor reading slightly different, and also possibly due to each pump. Each instrument is to be independently calibrated for it's operating flowrate. This Code is used to calibrate the flowrate on instruments. See FR calibration for more details.

ABCD2.x_SIM800Set: Network Settings are saved to the EEPROM of the SIM800 module. The settings include proper operational parameters and the endpoint where to send the data. This code programs the Arduino to communicate with the SIM800 and properly set the parameters. A SIM Card MUST be installed into the SIM800 while programming. The status light will stay on until the programming finishes, after which it will turn off which indicates its ready. A blinking status light indicates that the programming procedure was unsuccessful.

ABCD2.x.x_GPRSdiagnose: A more advanced SIM800 programmer that requires a knowledge of using AT commands. This code allows you to communicate with the SIM800 using AT commands which are passed through the Arduino. There are also hot words that run common procedures on the SIM800: "wake", "sleep", "rst", "setup", "prog". "prog" runs the same programming code that SIM800Set code does to set the endpoint and parameters. "setup" initializes the SIM800 the same way it would be initialized during normal operation.

Instrument settings:

Instrument settings can be changed by modifying the file ABCD2.x.x_MainCode and re-uploading the code to the instrument. This requires that you have installed Arduino, drivers, and libraries as described in the software section above.
Modifying the code involves simply changing the options in the "Settings" portion of the code which is the first section (starts around line 15 – ends around line 85). For on or off settings, 1 represents on, 0 represents off.

Settings explained:

Instrument Specific Settings:
InstrumentID: Set to the ID of the instrument. The instrument will use this ID as the filename for the data saved to the SD card.

FRcal: If **FR_Specific_Cal_enable** = 1, and **FR_EEPROM_cal** = 0, set this to the appropriate flow rate calibration byte value as provided or determined from a flow rate calibration.

Operational Options:
SIM800_present: If the SIM800 module is installed, set this to 1. This is important since the instrument will turn off the SIM800 module during data logging. If the SIM800 module is installed but this setting is off, the module may intermittently attempt to connect to a network which causes power surges that may affect the data. If the SIM800 is not installed, set this to 0, since the instrument will be attempting to communicate with the module if it is set to 1 (this will cause startup delays).

GPRS_enable: Option to transmit data to the database (1), or turn off SIM800 (0). **SIM800_present** must be set to 1.

TempCor: Correct for temperature variations using cell specific calibration factors. These calibration factors are stored in the SD card associated with cell. It is highly recommended to set this option to 1.

Flow Rate settings
FR_Specific_Cal_enable: Choose between using an instrument specific flowrate calibration (must also properly configure **FR_EEPROM_cal** and **FRcal**), or use the generic factory flowrate calibration provided from the sensor manufacturer (may be off by up to 15 cc/min).

FR_EEPROM_cal: is configurable when **FR_Specific_Cal_enable** is set to 1. Use this option to select between pulling the instrument specific calibration byte from EEPROM (1), and inputting the byte manually (0).

FR_EEPROM_cal = 0 requires a specific flowrate byte for a specific operational flow rate – ie. If you change the flow rate, you must also change the flowrate byte.

Selecting 1 (EEPROM) requires that the flowrate byte has already been programmed into the EEPROM of the Arduino. EEPROM sometimes gets

corrupt so it is safer to input the byte manually. See the FR calibration section for more information.

Selecting 0 requires that you know the flowrate calibration byte to use. This is either provided with the instrument, or can be determined by following the FR calibration instructions. Make sure to set **FRcal** if selecting 1.

FRi: The setpoint flowrate In cc/min (note that the FRcal byte must be calibrated to the specific Fri)

Duty Cycle settings
Sample_rate: The sample rate in miliseconds. Ie 2000 will give 2 second (.5 hz) data.

pumpSpeed: The starting PWM applied to the pump. The pumps speed will automatically adjust, however settings this speed close to what will produce the ideal FR (FRi) will reduce startup time.

Logging Settings
blockAvgSec: The block average size for sending GPRS data to network. 60 will produce 1 minute block average data for network transmission.

GPRSDataPoints: The number of data points sent out in per transmission.

avgRep: Number of final data points to average to report for block average data (sig,ref,ATN). Must be at least 1. This will increase the sketch size. If many options are enbabled this may not fit. Probably best to keep at 1.

secSave: Save every reading to SD card (1), or save BA data to SD card (0).

Serial Monitor settings
DataPrintOnly: (1) Only print formatted data to the serial monitor. This may be warranted for serial logging since you may only want data being passed through. (0) Serial stream data as well as error messages and start up functions.

Serial_BA: Select whether to output block average data (1), or every data point (0).

ErrorPrint: Print diagnostic errors to serial. This also saves errors to the SD card in a file called error.txt

Enabled logging
SD_enable*: Choose to log to SD card (1) or not (0).*

Flow_enable: choose to calculate BC by using sensor measured Flow data (1) or by using FRi value (0). Useful in certain lab settings where flow rate is externally controlled and measured.

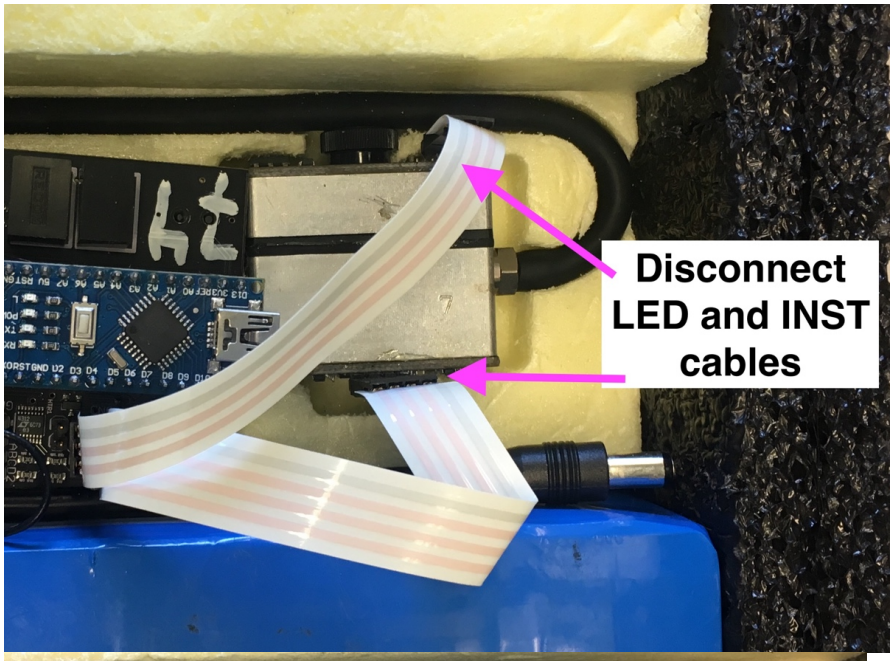
Flip_enable: Allow for cell flipping(1), or disable (0). For more information about cell flipping can be found in the Cell Flip section of this manual.

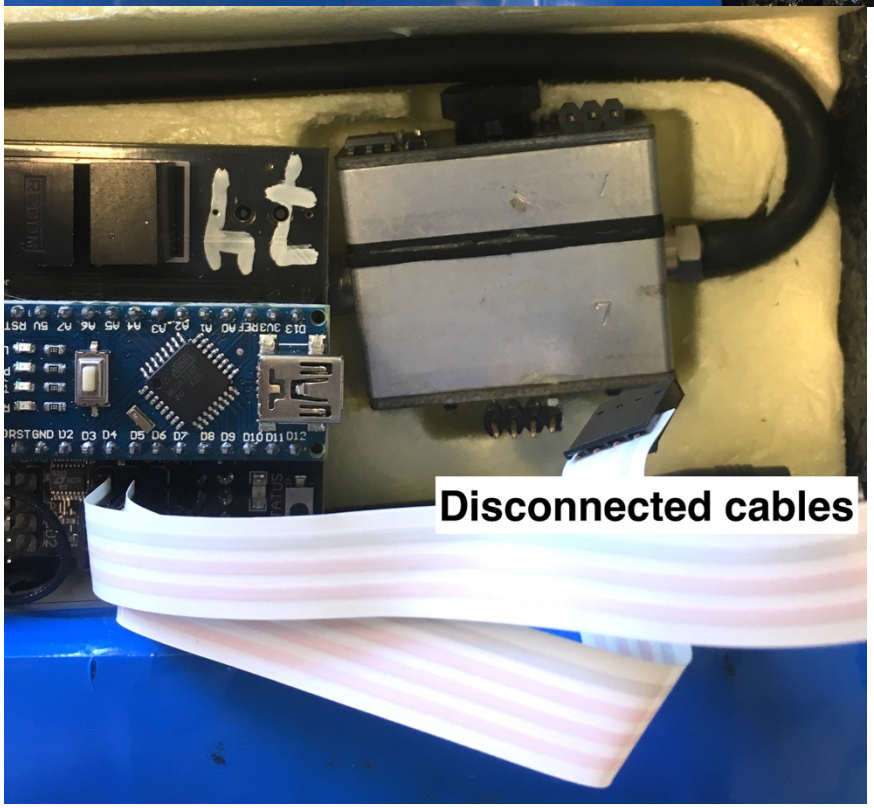
Cell:

Changing filters – schematic of cell and order of things
Filters should be cut using a number 4 punch. Filter material is EMFAB

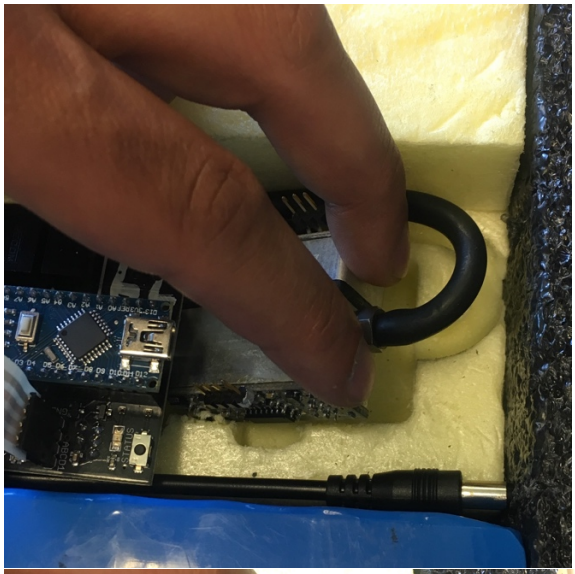
Steps for filter change:

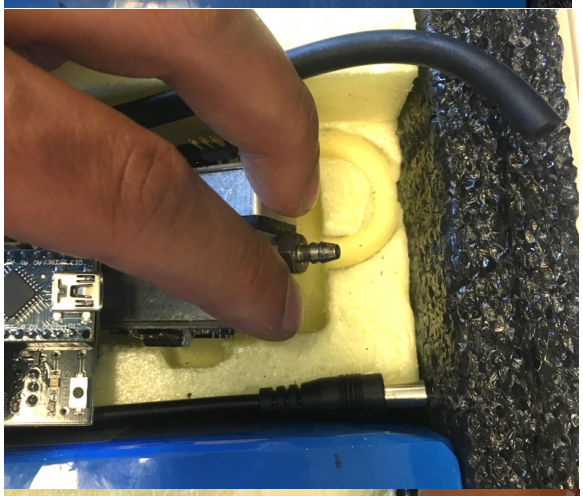
1) Disconnect LED wire and INST cable from Cell

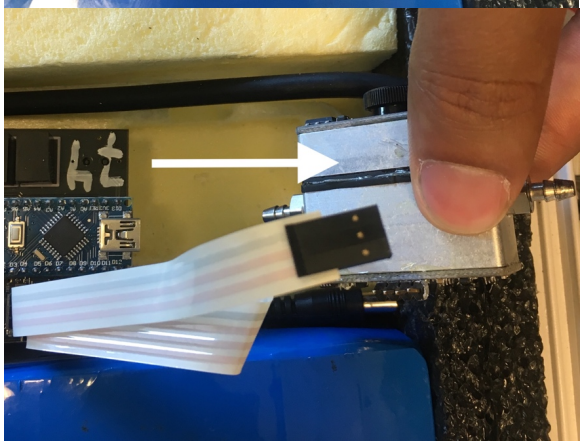




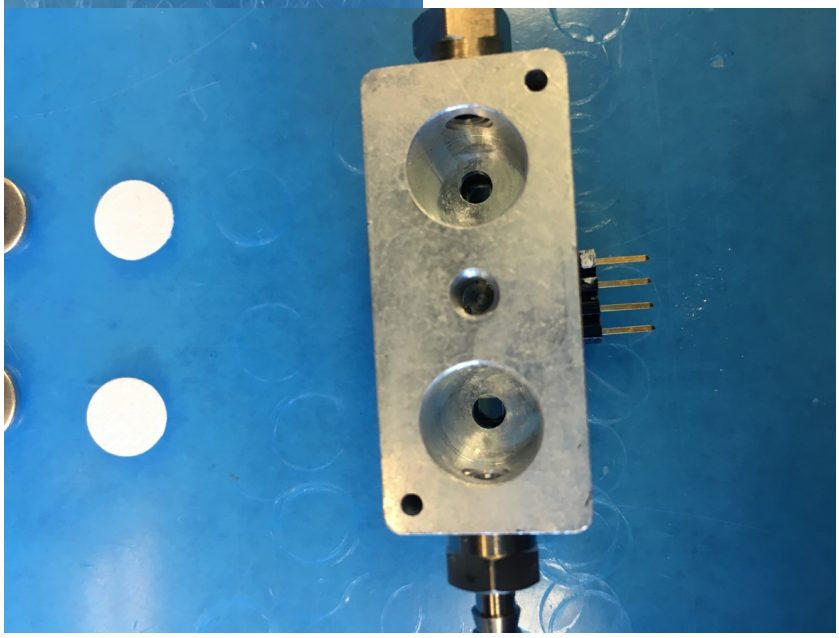
2) Use two fingers to pull cell inlet side out. Disconnect the inlet tubing. Then pull cell out away from outlet.



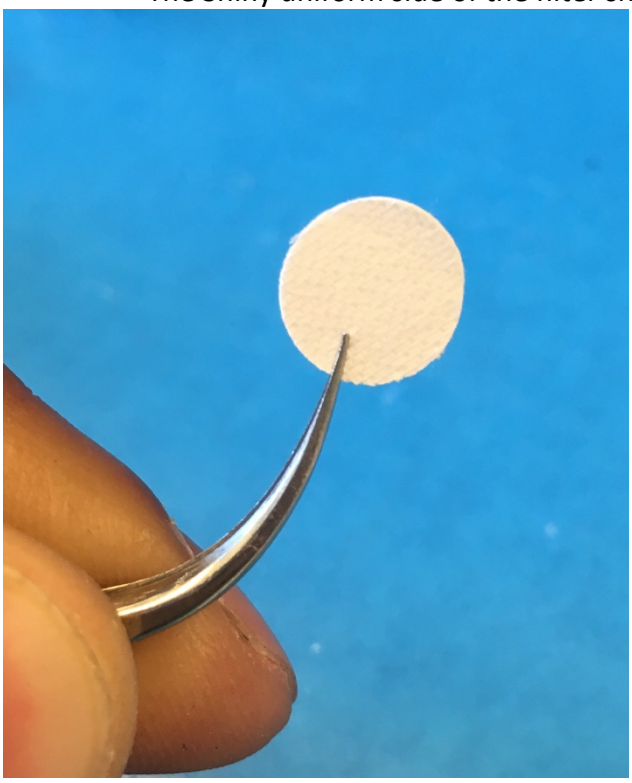


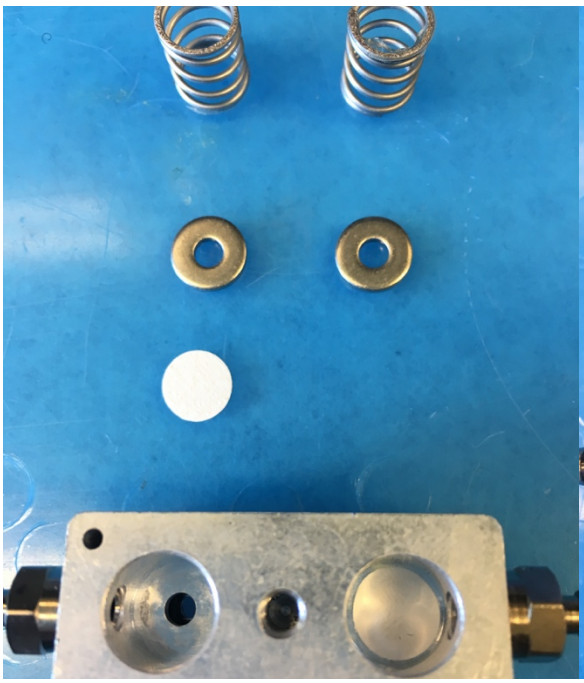


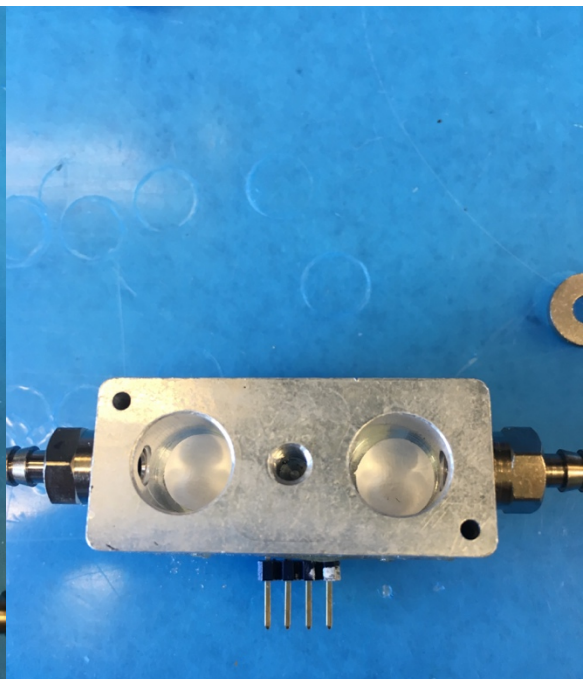
3) Remove thumbscrew using an alenwrench if too tight. Remove the springs, washers, and old filters



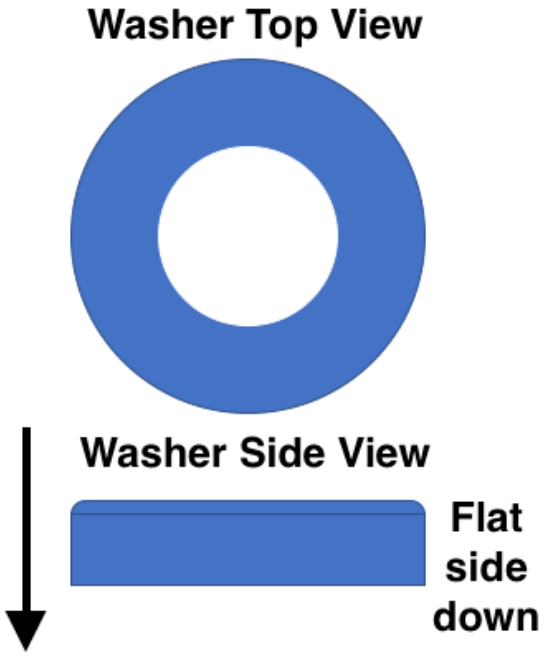
4) Use tweezers to hold filters from the edge and place them inside the cell cavities. The Shiny uniform side of the filter should face downwards out of view.



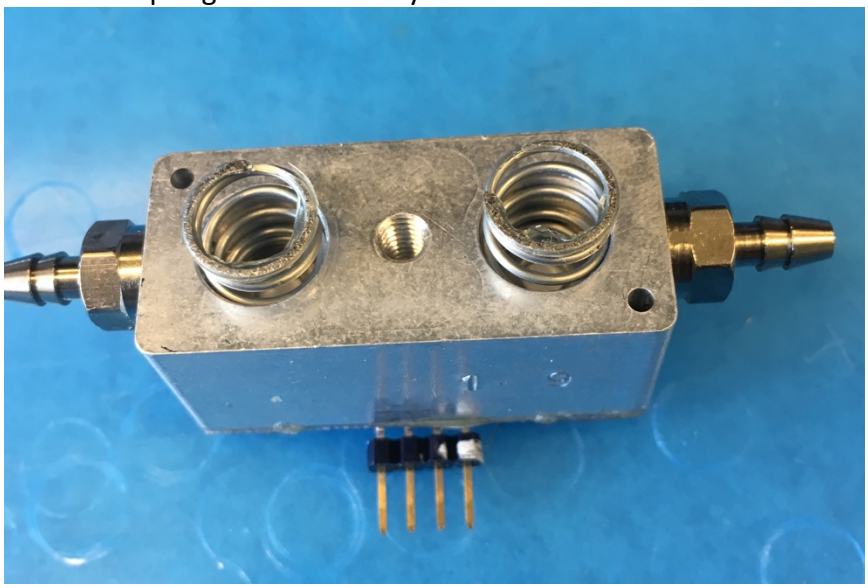




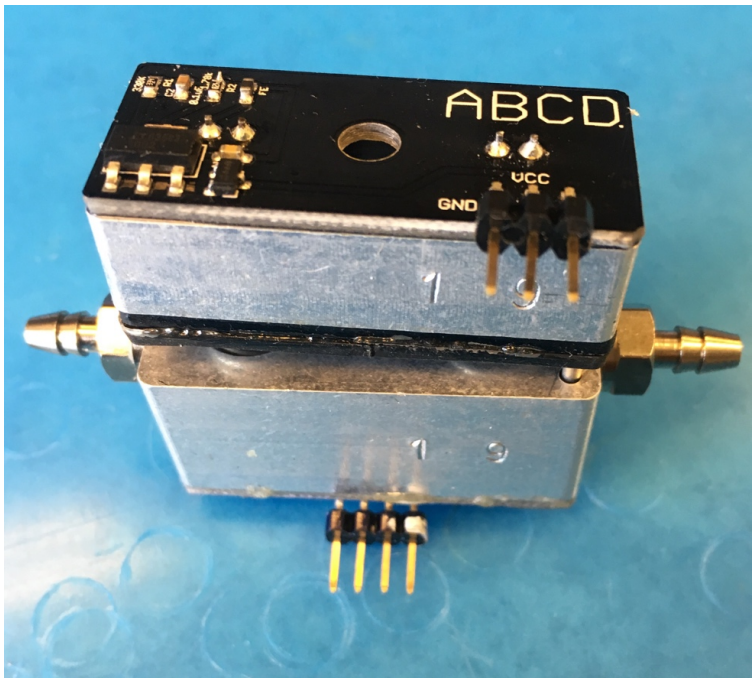
5) Use tweezers to drop the washers down into the cavities. The flat side of the washer should face down towards the filter.

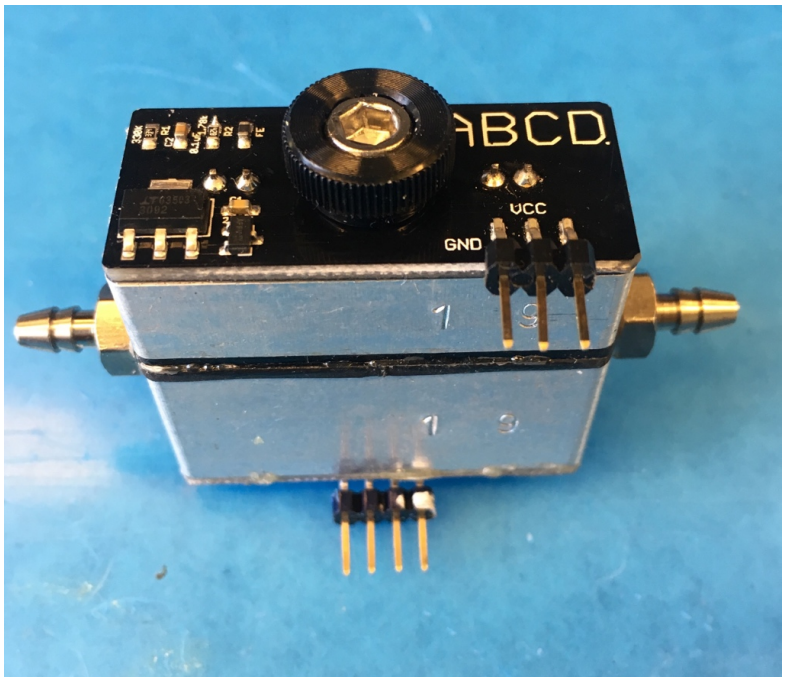


6) Insert the springs into the cavity

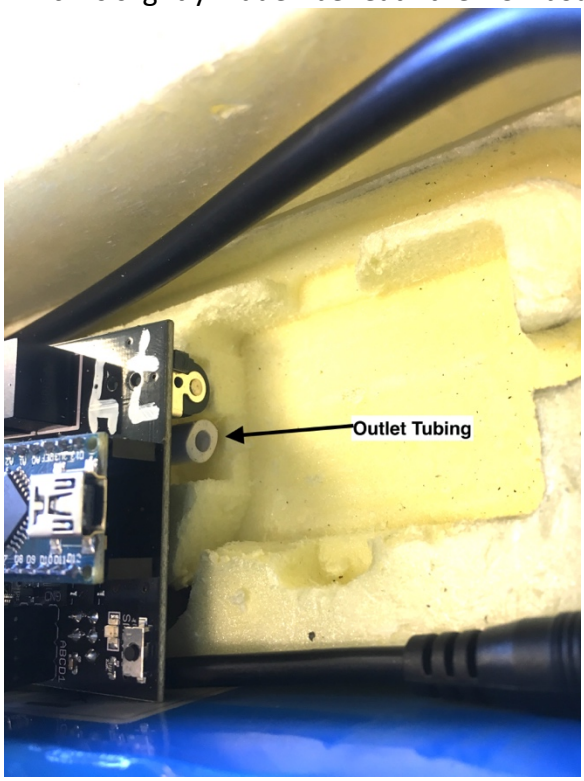


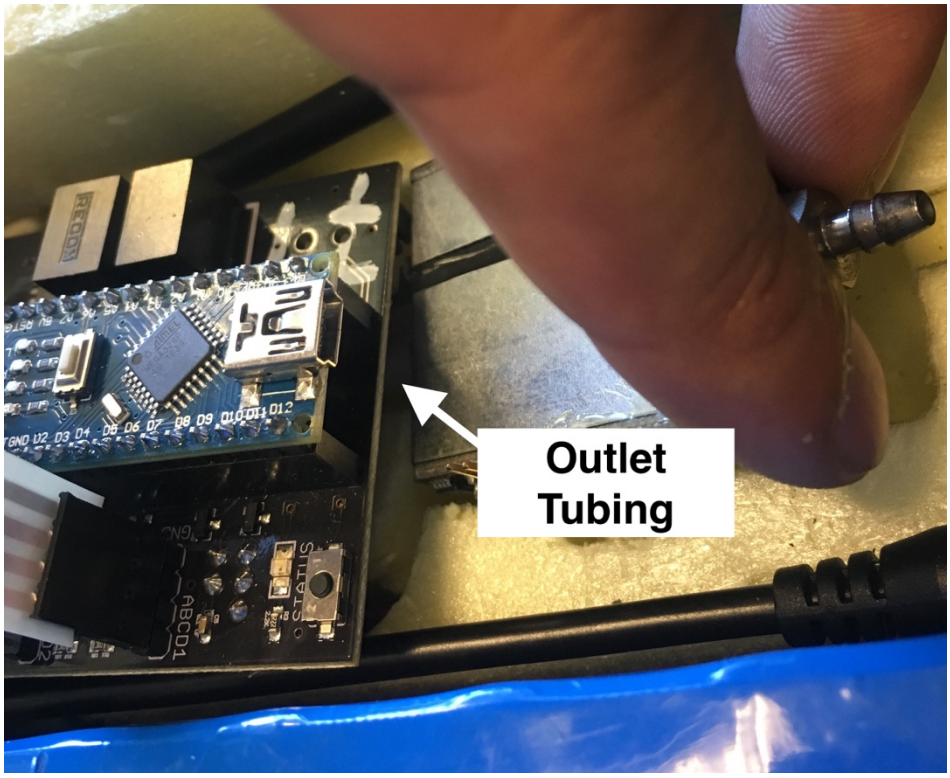
7) Register the LED shield into the Instrument manifold. Insert thumbscrew and tighten. It is good practice to perform a leak check to ensure that the cell holds vacuum.



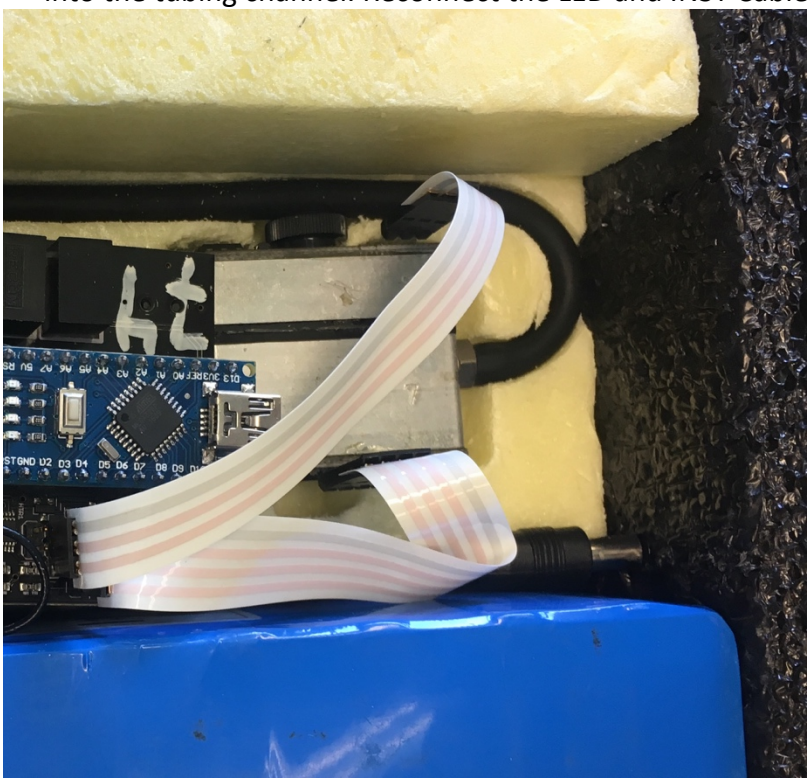


8) Place the cell inside the instrument by guiding the outlet barb into the outlet tubing which is slightly hidden beneath the AUX board.





9) Reconnect the inlet tubing, set cell into the insulation cavity, and set the inlet tubing into the tubing channel. Reconnect the LED and INST Cables



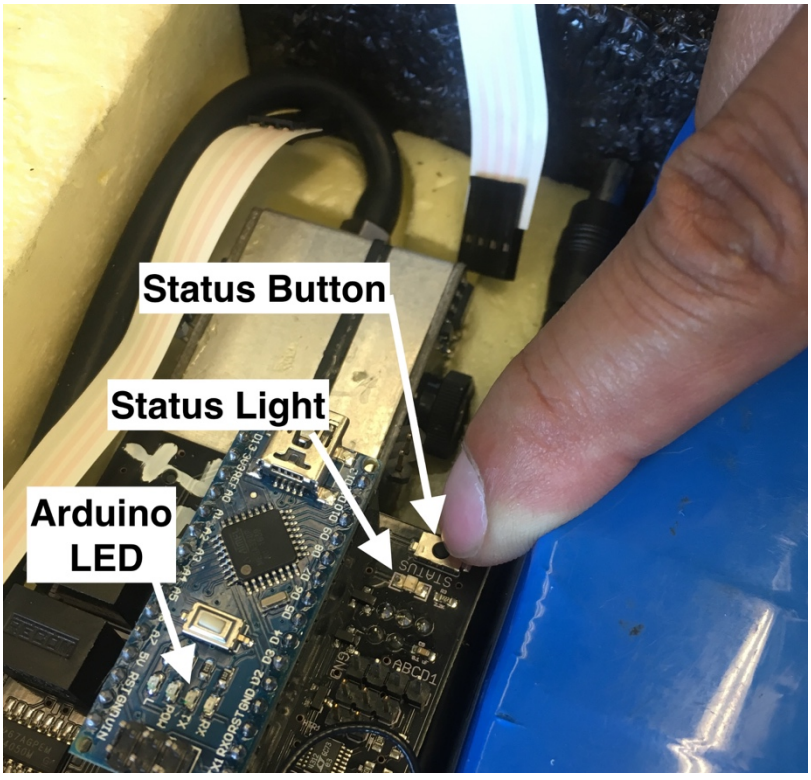
ATN reset:

Attenuation (ATN) is a number that is representative of the BC filter loading. Changes in ATN are relative to a starting point which should be defined as 0. ATN is calculated from the relative magnitudes between the signal and reference cells. Since the LEDs, photodetectors, and filters in each cell have intrinsic differences, even clean filters will produce an initial ATN that is offset from 0. Since we are interested in keeping track of ATN relative to 0, we reset the ATN to 0 offset. There are two ways to do this:

1) **Cell ID change:** If a new cell is inserted into the instrument that is different from the last cell used (different cell ID), the instrument will detect a new cell is being used and automatically reset the attenuation.

2) **Button change:** If replacing only the filters in the same cell that was previously used in an instrument, the instrument needs to be instructed to reset the attenuation. To reset attenuation with the button, first wait till the instrument is operating normally and data logging (The LED on the Arduino will be flashing at the set sample rate – ie. 1 hz). Press and hold the status button until the status light turns on, and immediately let go when it turns on. The status light will blink rapidly (5 times in one second) to confirm that the attenuation has been reset.

Note: If cell flipping is enabled, the ATN will automatically be reset when the Cell is flipped.

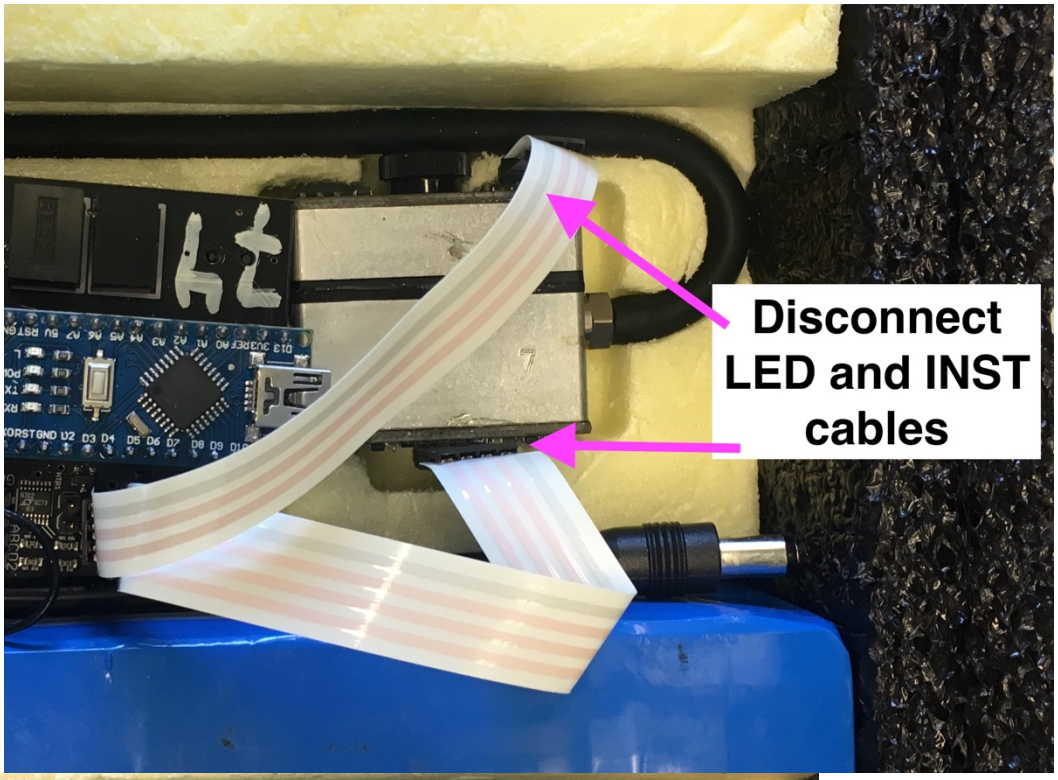


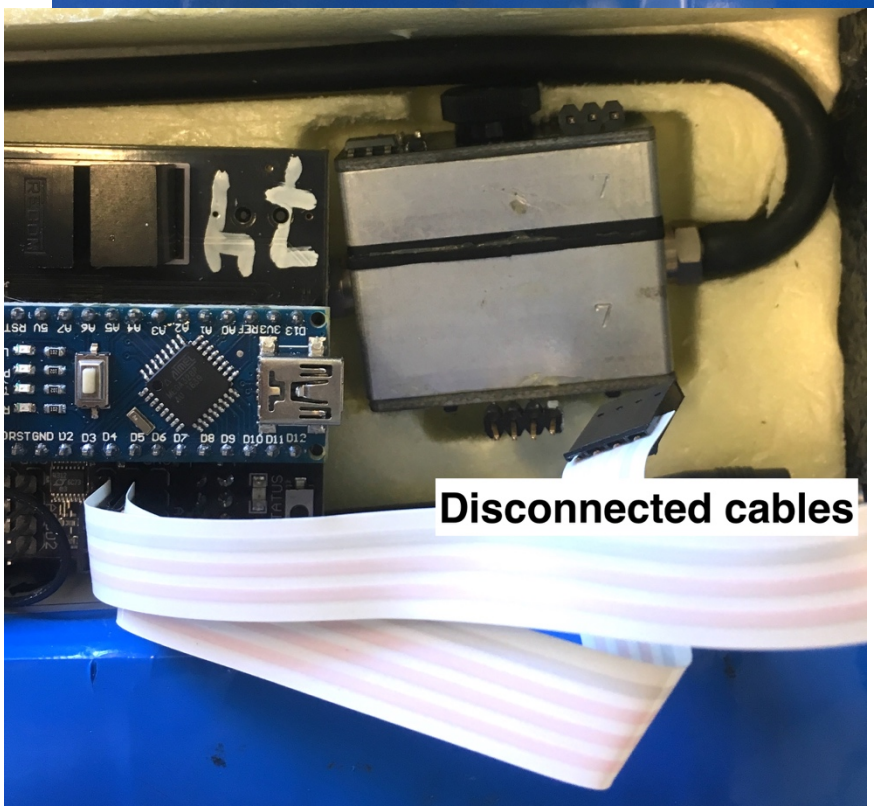
Cell Flip: Cell flipping is a method in which the cell can run in reverse and effectively double the amount of sampling before needing the change the filter. The flow path of the instrument directs air first through the signal cell and then through the reference cell. Since the BC-laden sample air gets filtered through the first signal filter, there reference cell will not be loaded. Cell flipping involves flipping the orientation of the cell such that the air flows in through the default outlet, and out through the default inlet so that the default signal cell (which is already BC loaded after use) now serves as the reference cell to the clean reference filter, which now serves as the signal cell.

To enable Cell Flipping, the **Flip_enable** setting must be set to 1 in the maincode firmware. Refer to the "Instrument Settings" section for more information.

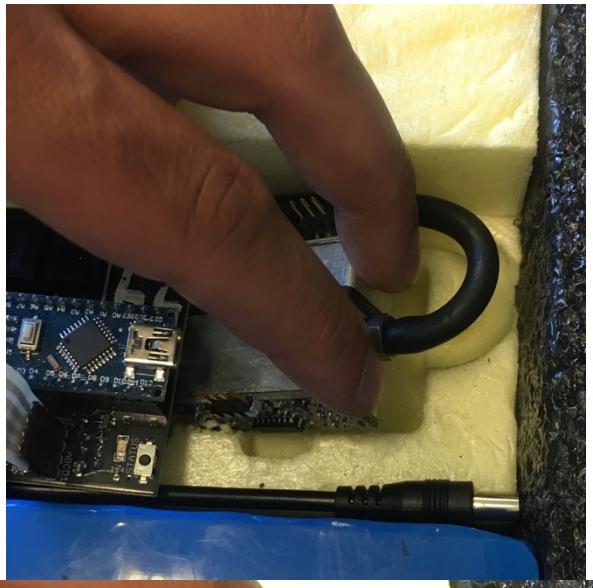
In practice one generally samples with the instrument until an ATN of 80-100 is reached before they decide to either change the filters or flip the cell. To flip the cell, the technician simply removes the cell, rotates it 180 degrees and reinserts it. The Arduino then needs to be instructed to operate in cell flipped mode, which is done by turning on the instrument while holding the status button. Full instructions for Cell flipping:

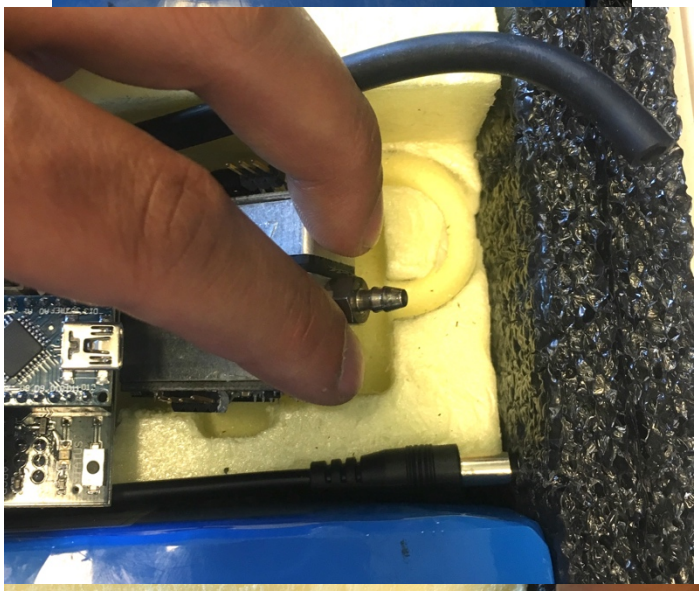
1) First power off the instrument by turning off the power switch on the battery.
2) Remove the cell and reinstall in a 180 degree orientation
    a. Disconnect LED wire and INST cable from Cell

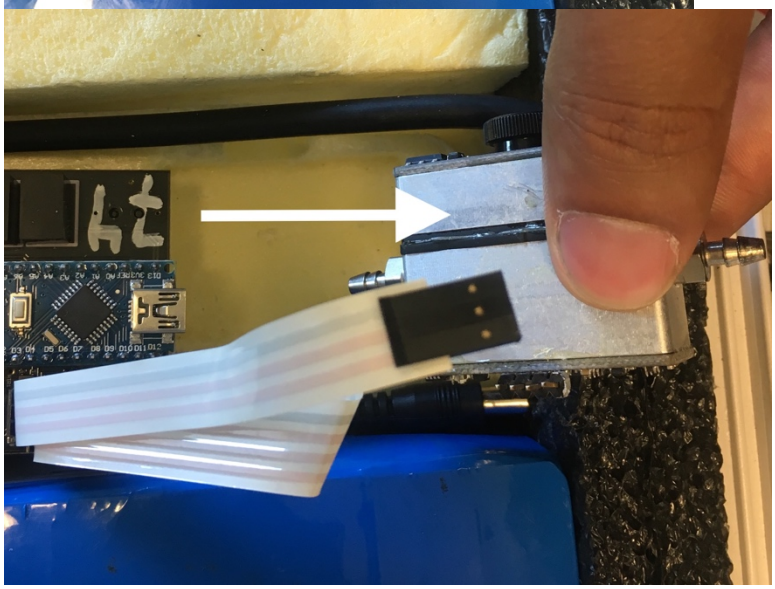




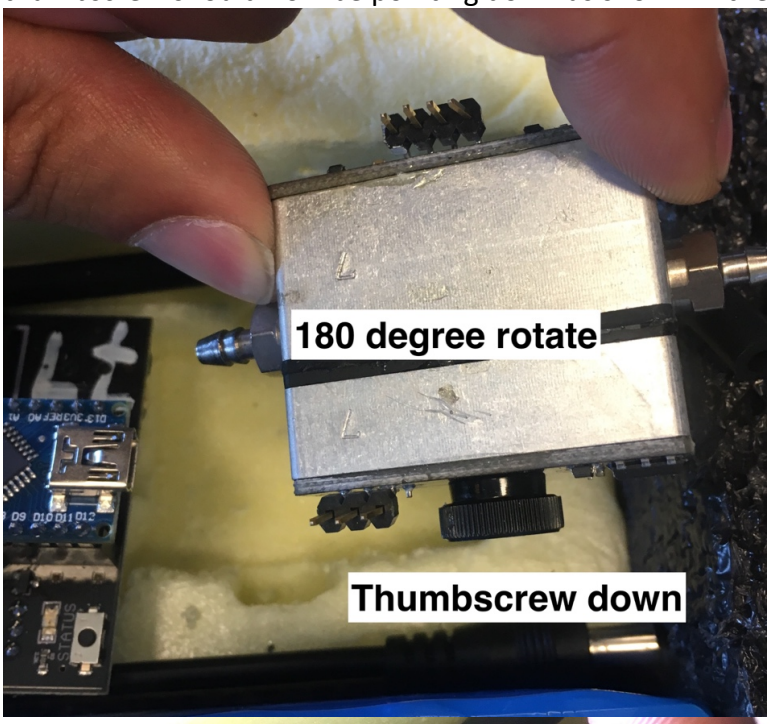
b. Use two fingers to pull cell inlet side out. Disconnect the inlet tubing. Then pull cell out away from outlet.

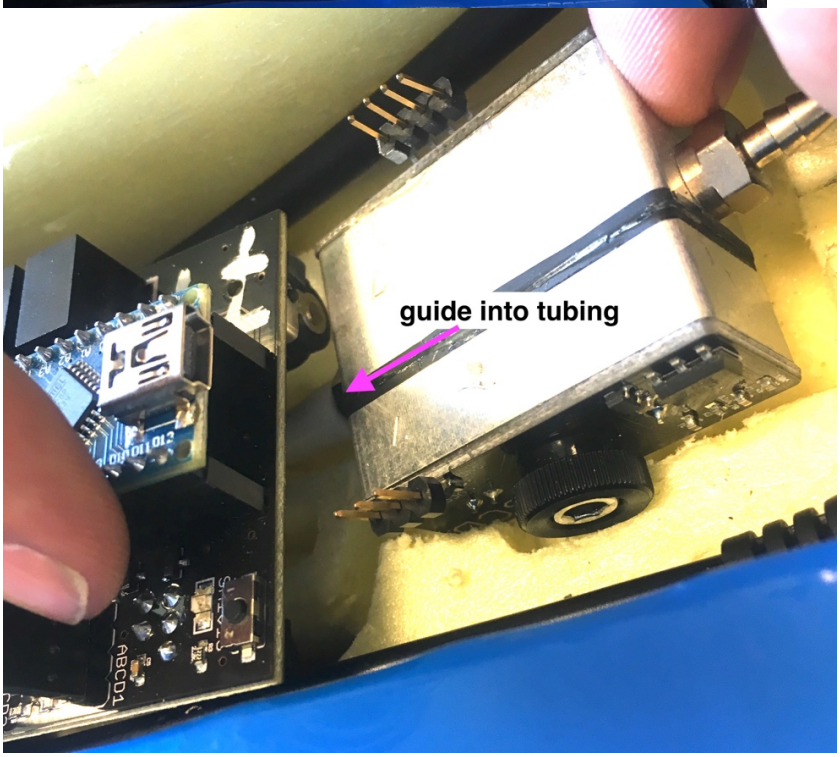


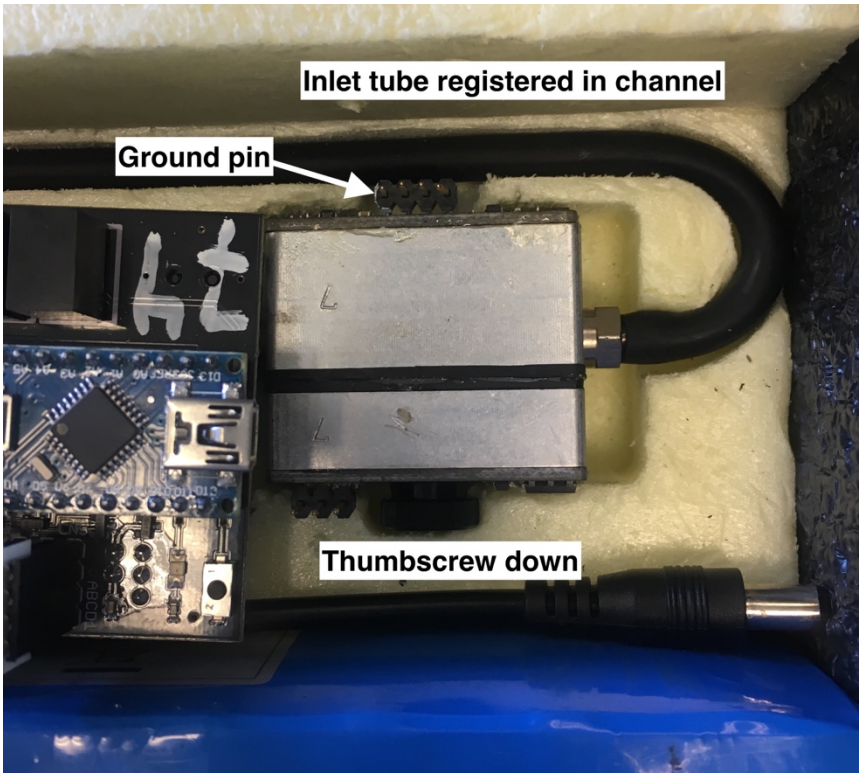




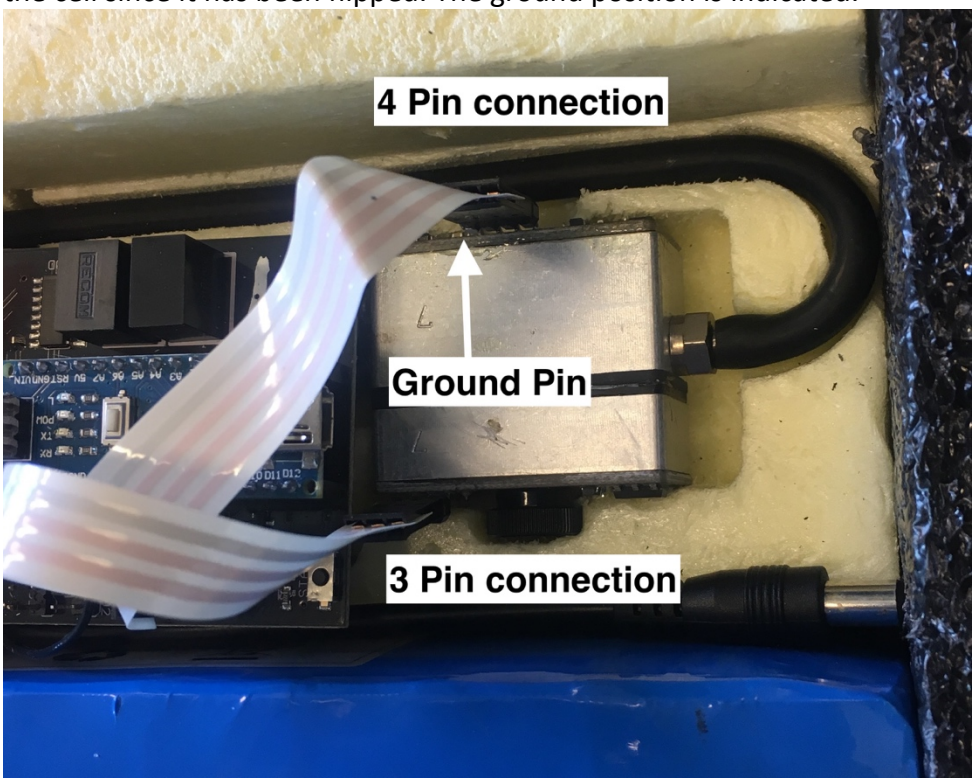
c. Turn the cell 180 degrees and reinstall by guiding the outlet barb into the outlet tubing which is slightly hidden beneath the AUX board. The thumbscrew should now be pointing down as shown in the figures.



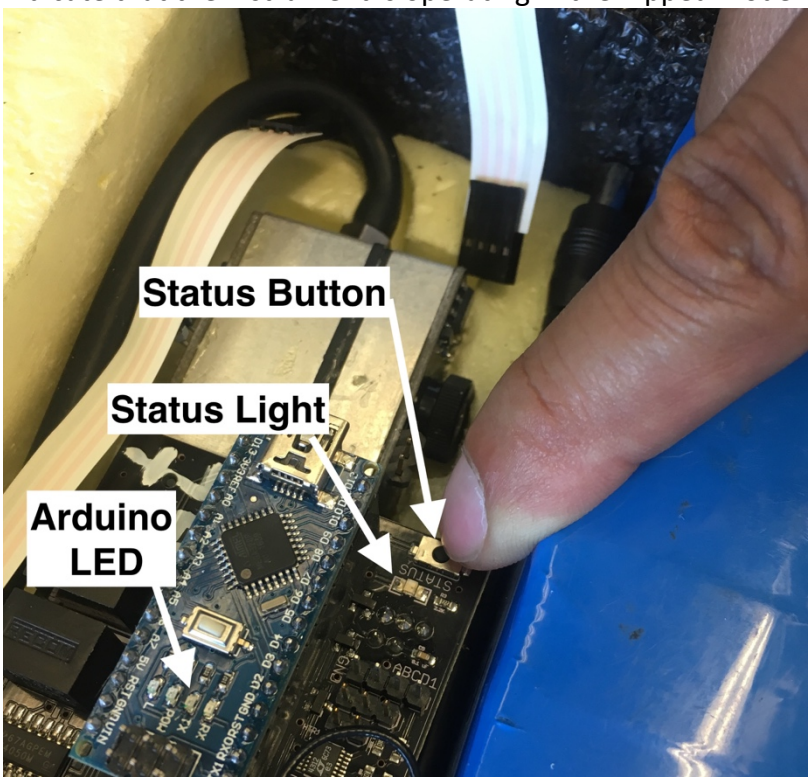




d. Reconnect the wires. Note that the 4 pin connector now goes to the top of the cell since it has been flipped. The ground position is indicated.



e. When starting up the instrument for the first time since being flipped, hold down the status button while powering on the instrument. The Status LED will blink rapidly 5 times immediately after the normal slow 3 blinks to indicate that the instrument is operating in the flipped mode.



When operating in the flipped mode, the instrument will automatically sample first through the new signal cell and second through the new reference cell. If the setting is enabled when the cell is oriented in the default position, it will result in wrong order sample collection, reporting of negative black carbon values, and continues decrease in ATN.

When the instrument is first instructed to operate in cell flip mode (button pressed while startup), the ATN will automatically be reset to 0.

In order to reset the instrument to operate In the normal configuration mode, perform a button 2 select: While the instrument is in normal logging mode (Arduino LED blinking at the sample rate specified), press and hold the Status button until the Status LED turns on, turns off, and then immediately release the button when the LED has turned off. The Status LED will blink slowly for 1 second.

Flowrate Calibration:

Each Flow Rate (FR) sensor inherently produces slightly different readings if using the generic factory calibration (±~10 cc/min). It is thus recommended to use a FR sensor specific calibration for each instrument.

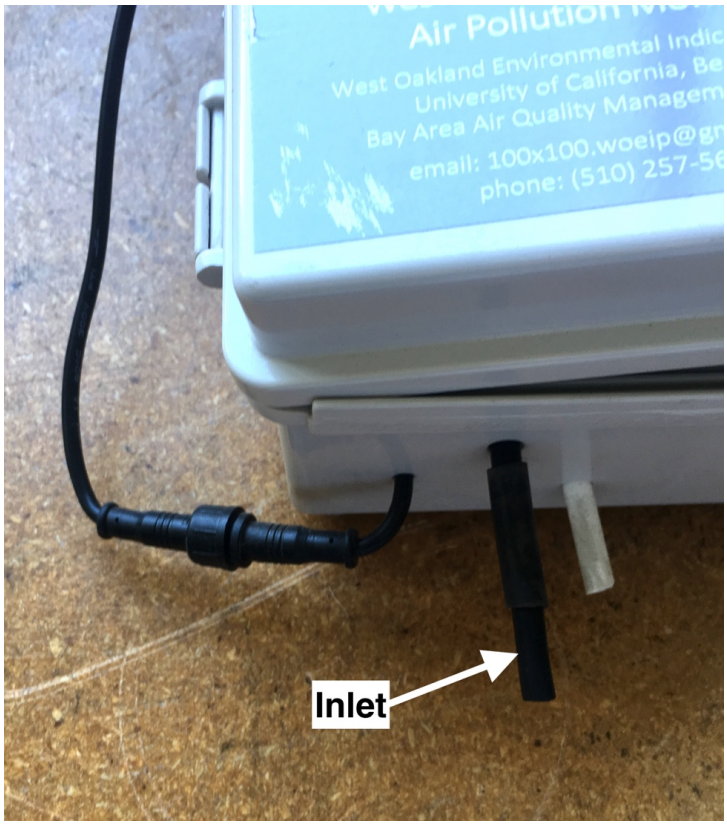
To use a FR Specific calibration, the instrument settings must be configured properly by setting **FR_Specific_Cal_enable = 1** and setting **FRcal** to the instrument specific calibration byte. To find instrument specific calibration byte you will need a flowrate measurement reference such as a Gillibrator.

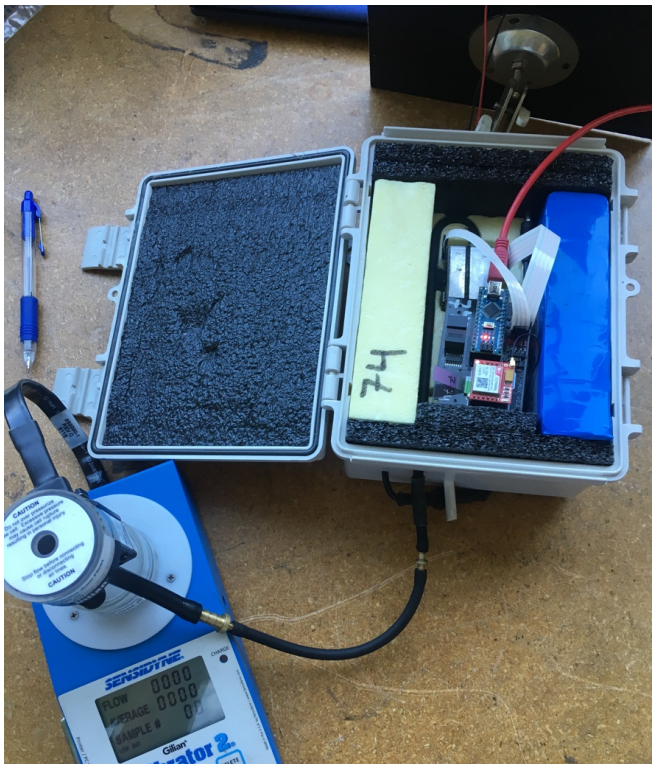
A Flowrate Calibration must be performed anytime you change the target flow rate **FRi** since the **FRcal** byte value corresponds to a single flowrate.

Procedure:

1) Turn on the Instrument from the Battery switch

2) Follow “Uploading Code” instructions to upload the **ABCD2.x.x_FRcal** code to the Arduino. Open the serial monitor.
3) Connect the FR measurement reference instrument to the inlet of the ABCD.


Air Pollution
West Oakland Environmental Indic
University of California, Be
Bay Area Air Quality Managem
email: 100x100.woeip@gr
phone: (510) 257-56
Inlet

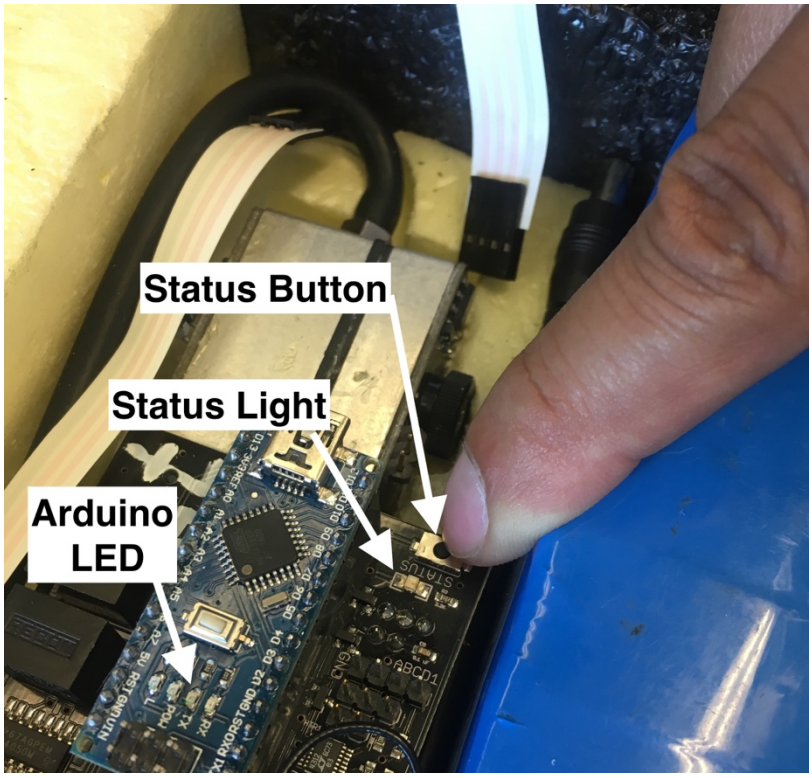

SENSIDYNE
FLOW 0000
AVERAGE 0000
SAMPLE # 00
Gilian

4) Allow the ABCD to warm up for at least 5 minutes so the pump and the FR sensor will be in a stable condition before calibrating.
5) Determine the FRcal value by enter a target value between 0-1024 into the serial monitor. The instrument will self-adjust the pump speed to match the and hold the value of the FR sensor output.
    a. The value corresponds to the output value of the flowrate sensor. We are trying to determine what value corresponds to our target flow rate.
    b. Keep adjusting the input target value until the flowrate is at your target flowrate as measured on your flowrate reference instrument.
        i. Ie. If your target flowrate is 110 cc/min, figure out what target input value corresponds to 110 cc/min.
        ii. When the flowrate is held stable at your target flowrate, enter "g" into the serial monitor to save the value to EEPROM. Also, record this value separately along with the flowrate.
            1. After it is saved in EEPROM, if **FR_EEPROM_cal** is set to 1 in the main code, the instrument will automatically pull this value. This is not however recommended since EEPROM often gets corrupted.
    c. The value that corresponds to your target flowrate is the **FRcal** byte. Record this value along with the target flowrate (write it on the insulation inside the instrument as well as in the instrument page of the database website)
6) Upload **ABCD2.x.x_MainCode** to the instrument and set **FR_Specific_Cal_enable** = 1, **FRcal** = the determined FRcal value, **FRi** = the flow rate associated with the FRcal value, and **FR_EEPROM_cal = 0** (if manually entering the FRcal value – recommended).

Setting Time:

The time is held in a digital chip called a Real Time Clock (RTC). The RTC is constantly powered by a coin cell battery on the bottom side of the AUX board (Figure 7). A dead coin cell battery or a short circuit of the instrument could lead to an RTC being resetting to the beginning of time (January 1970). If this happens the RTC time must be reset to the current time. Procedure:

1) Connect Arduino to computer.
2) Follow "Uploading Code" instructions to upload the **ABCD2.x.x_SetTime** code to the Arduino.
3) Hold down Status button while the code is uploading and while Arduino starts running the program (this will reset time to compiled time) AUX LED will turn on for 1 second as verification of reset action.



   a. It is important to do this step right as you upload the code, as it sets the time to the compiled time by the computer.
   b. LED will blink every second as a warning if time was not reset (you may have intended not to reset it if you just wanted to verify time)
   c. LED will stay on solid if the time is definitely not up to date (ie. The time is set to before Feb 9 2017).

4) Verify that the time is up to date. This can be done by either watching the serial monitor, or plugging in the Diagnostics reader to monitor the time output.
5) Verify proper functioning: turn off all power and turn it back on, the time should still be correct. You should verify this by disconnecting power, then reconnecting and watching serial monitor or a diagnostics device. For this step, **do not** hold down the button. The LED should be blinking which indicates that no changes have been made.
6) If the clock continues to be reset with power loss, the LED will be solid. Verify that a charged coin cell battery is properly installed.
7) Re-upload the **ABCD2.x.x_MainCode** with proper settings.

Battery:

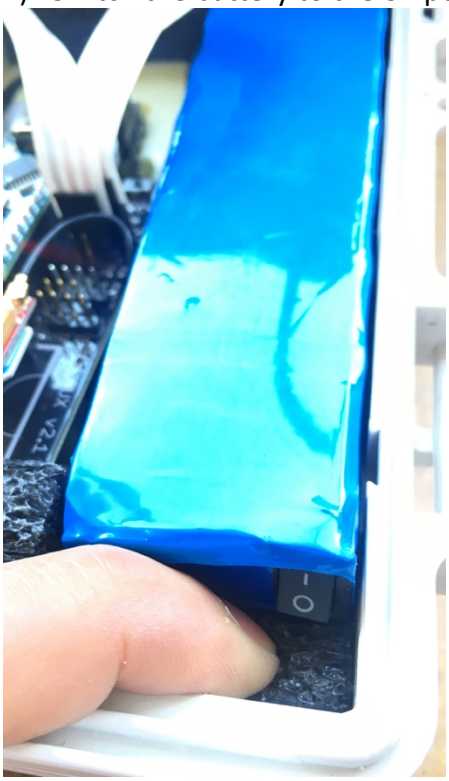
An instrument with a solar panel exposed to direct sunlight should run indefinitely. However if there is limited or no sunlight available and there is no AC power available, you may need to change or charge the battery.

To Charge the battery inside the instrument:

1) Remove the side insolation

2) Disconnect the DC jack connection

3) Plug in the AC power supply to exposed female jack.
4) Switch the battery to the on position.



To replace Battery:

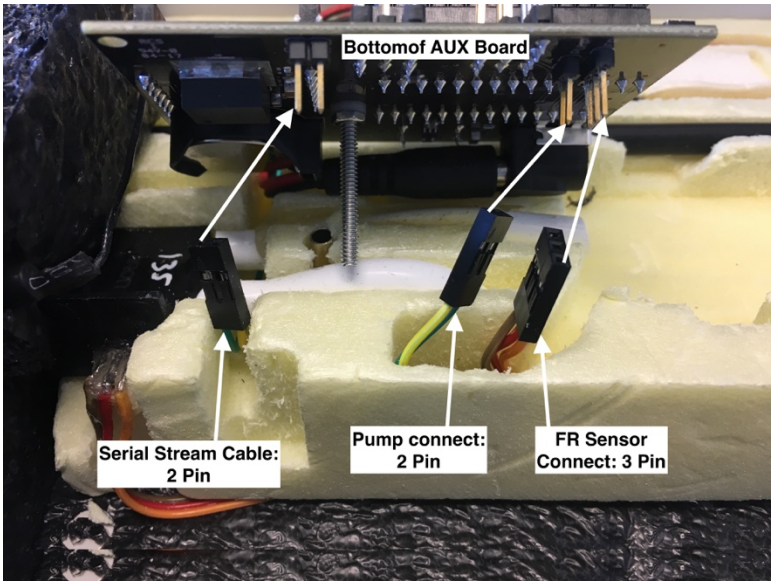
1) Remove side insulation, disconnect DC jack connection, remove top insulation:

2) Slightly lift AUX board:

3) Lift Battery up from the end:

4) Snake the battery cable out from under the AUX board:

5) Before replacing the battery, ensure that the pump, FR and Serial stream connections are still connected:



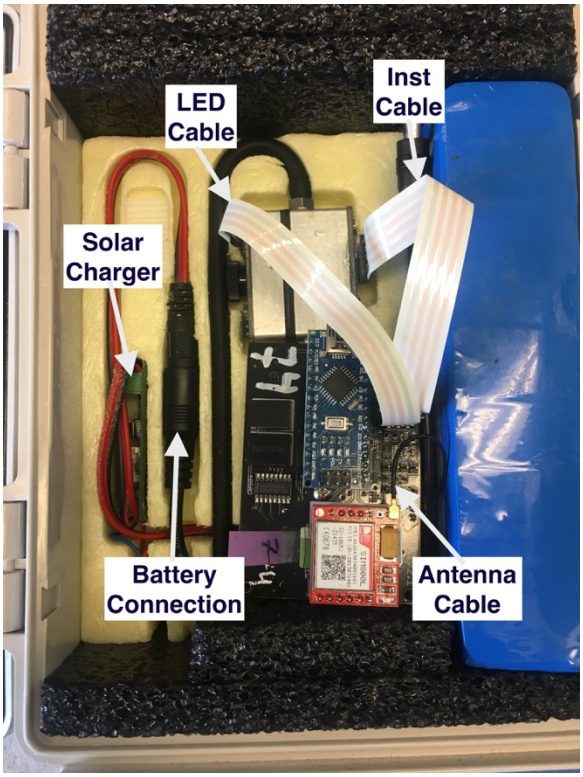
6) Work backwards from step 4.

Switching between Solar power to DC power sources:

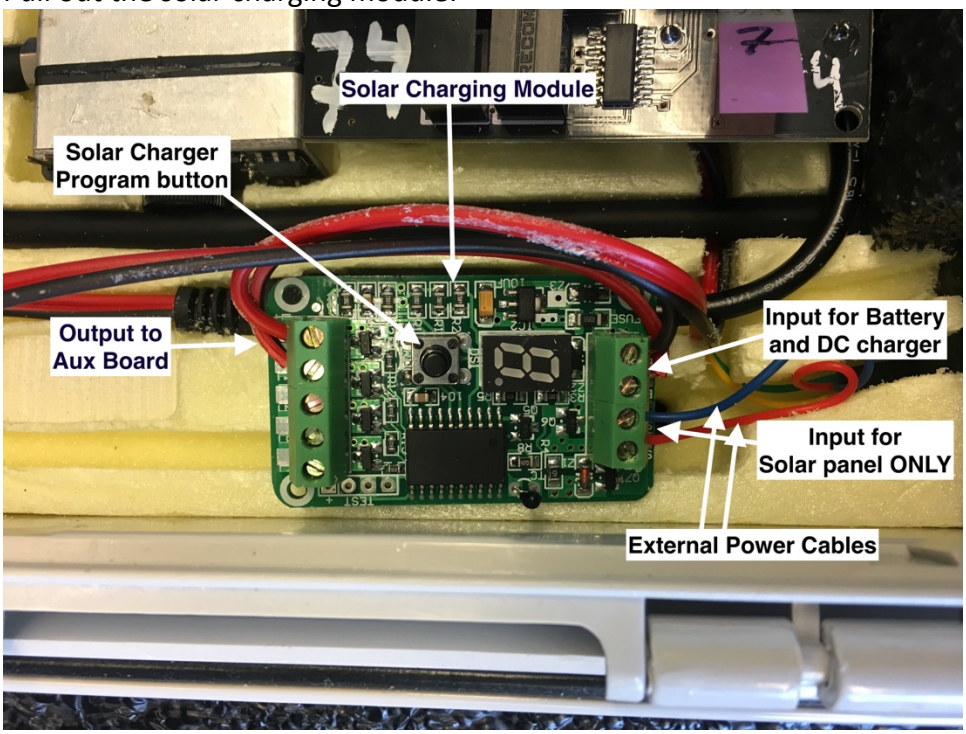
When a power outlet is available, it may be desirable to switch the instrument to work in AC mode. Switching between solar power and DC power sources involves changing the wiring of the solar charger module. To begin, first remove the side insulation to expose the solar module:

1)

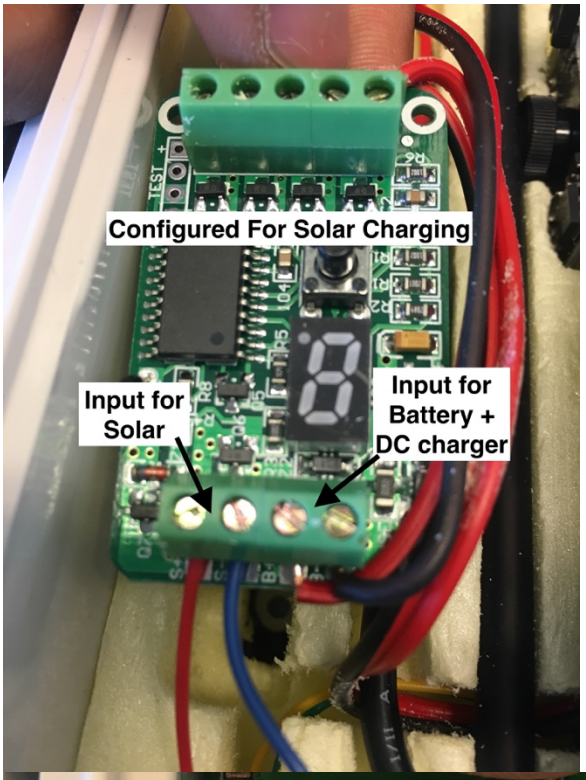

74

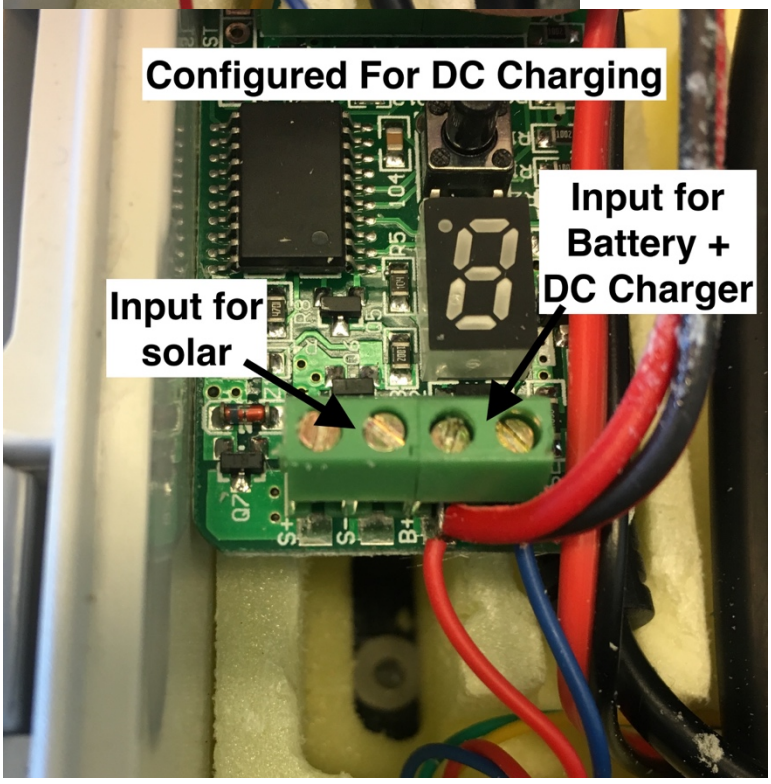


2) Pull out the solar charging module:



3) To switch between solar power and AC mode, switch the “external power cables between the input for Solar panel and Input for Battery and DC charger terminals:
4) Wire accordingly:

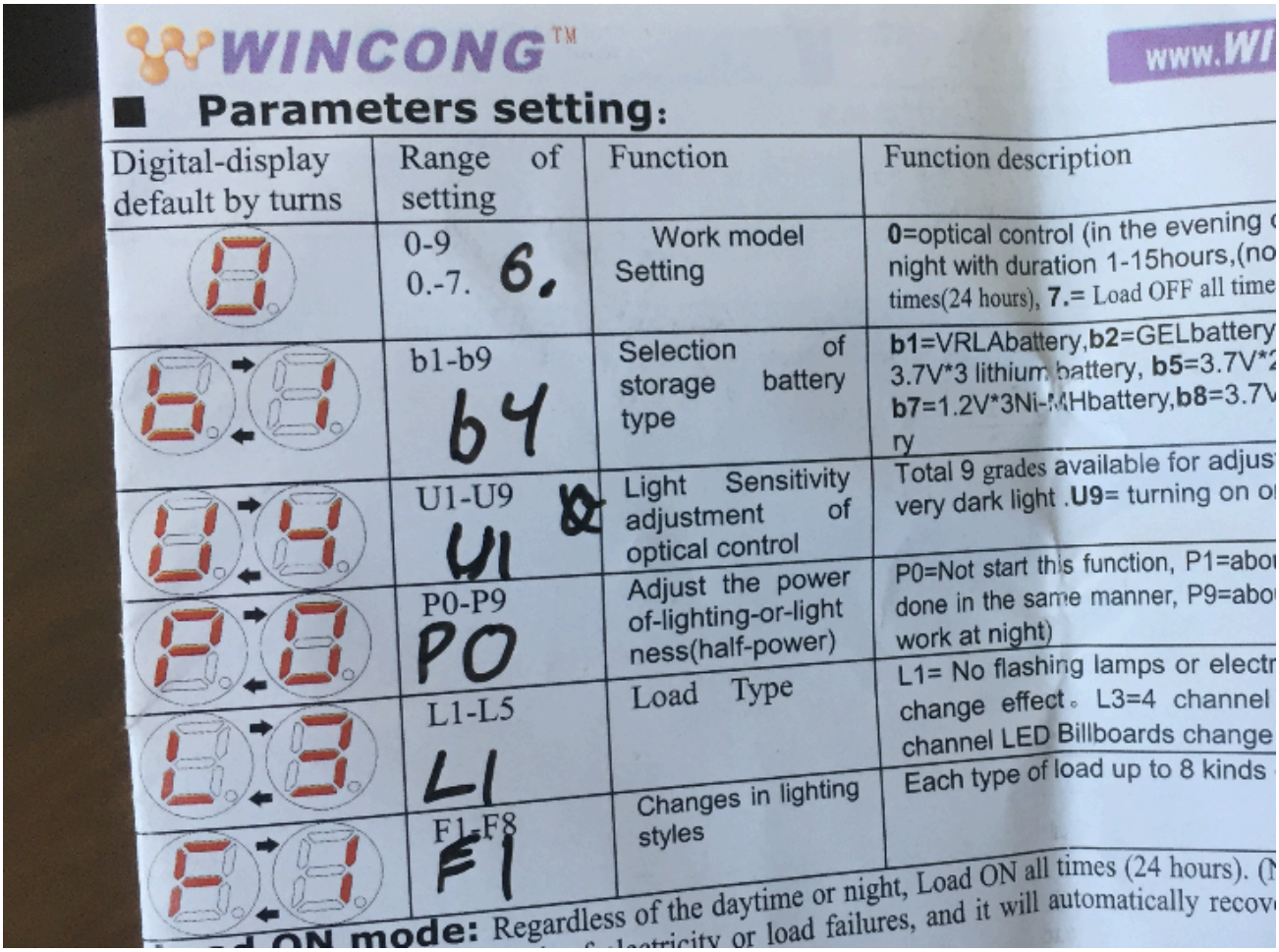




Programming Solar Charger:

It is an extremely rare sitation that the settings on the solar charging module would become reset. If it does become reset, it should be reprogrammed back to the necessary settings. To program the solar charging module, hold down the solar charger program button until the display starts to flash indicating that the settings are ready to be reset. The display will first show you what parameter is selected; pressing the button again will allow you to switch between settings. The correct settings are:
6. B4 U1 P0 L1 F1

For the first parameter is important to distinguish that “6.” Is different from “6”


WINCONG™

www.WI

■ Parameters setting:

Digital-display default by turns	Range of setting	Function	Function description
8.	0-9 0.-7. 6.	Work model Setting	**0**=optical control (in the evening night with duration 1-15hours,(no times(24 hours), **7.**= Load OFF all time
b1-b9 b4	Selection of storage battery type	**b1**=VRLAbattery,**b2**=GELbattery 3.7V*3 lithium battery, **b5**=3.7V* **b7**=1.2V*3Ni-MHbattery,**b8**=3.7V ry
U1-U9 U1	Light Sensitivity adjustment of optical control	Total 9 grades available for adjus very dark light .**U9**= turning on o
P0-P9 P0	Adjust the power of-lighting-or-light ness(half-power)	P0=Not start this function, P1=abo done in the same manner, P9=abo work at night)
L1-L5 L1	Load Type	L1= No flashing lamps or electr change effect。L3=4 channel channel LED Billboards change
F1-F8 F1	Changes in lighting styles	Each type of load up to 8 kinds

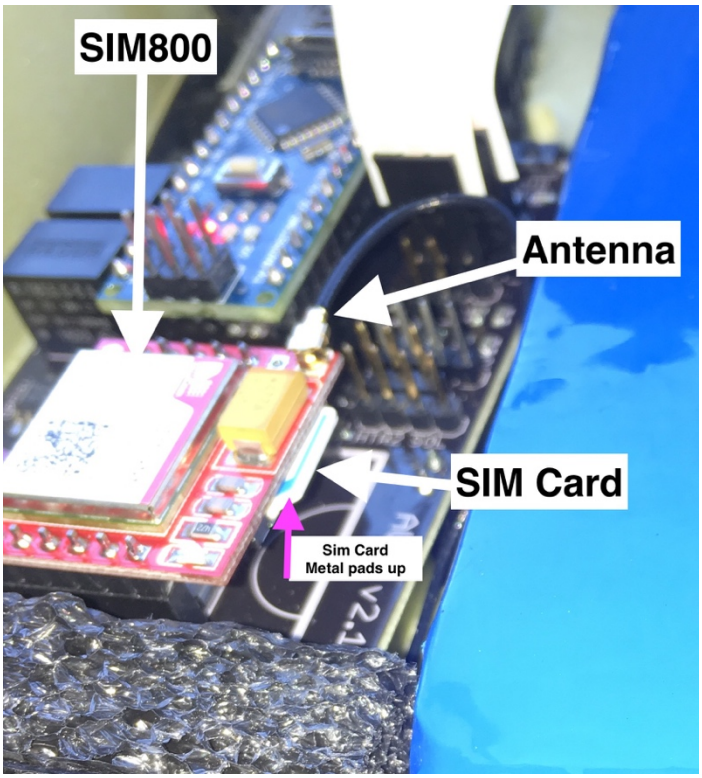
ON mode: Regardless of the daytime or night, Load ON all times (24 hours). (N
electricity or load failures, and it will automatically recov

Temperature Calibration Factors:

Cell ID and calibration factors should be preloaded on the SD cards in a file called info.txt
An example info.txt is saved in the google drive directory. The format of the file is:
cellID,refM,refB,sigM,sigB
eg.
1,1.232,0.0032,1,1.0234,.0022

This info.txt stays on the SD card, and the filename should be the cell ID. The SD card should always travel with the cell.

SIM800:



The SIM800 is a cellular modem capable of providing internet connectivity to the instrument. Network configuration settings and the URL endpoint in which to send the data is stored in the EEPROM of the SIM800 and must be programmed by the Arduino (if not programmed already). If there are ever issues with transmission that are not related to poor network connectivity, it may be helpful to reprogram the SIM800.

Programming SIM800:

1) Plug in SIM800
2) Make sure SIM card is present (notch side out, metal side up)
3) Turn on supply power to AUX board (12V)
4) Upload program **ABCD2.x_SIM800Set** or **ABCD2.x.x_GPRSdiagnose**(more advanced – skip to section)
5) Wait. While initializing, AUX LED is solid red while initializing.
    a. AUX LED turns off When finished successfully

b. AUX LED blinks if there is an issue.
    i. Verify SIM card is inserted
        1. Verify it is inserted properly (metal up, card notch outward)
    ii. Verify SIM800 is installed properly (proper orientation and pinning)
    iii. Verify 12 volts is supplied to AUX board (ie battery on)
    iv. Hard Restart. Ie. Disconnect from all power sources, and then turn on power. This will ensure that the Arduino and SIM800 are talking properly.
    v. If problem persists, try different SIM card and/or SIM800
        a. Contact support

If using **ABCD2.x.x_GPRSdiagnose:**

The SIM800 communicates via AT commands. Consult the SIM800 manual for more information and commands.
This code allows for several predefined routines as well as direct 2 way communication of AT commands to the SIM800 through the serial monitor.
The predefined routines are called by typing these hotwords into the serial monitor: “wake”, “sleep”, “rst”, “setup”, “prog”. “prog

wake : This will take the SIM800 out of sleep mode and turn on the cellular functions.
sleep : This will turn off cellular functions and put the module to sleep (preventing random power surges associated with routine connections to the cell towers)
rst : not working yet
setup : This configures the default startup procedure to the sim800. This is the same setup procedure called during the initialization of the main instrument code.
prog : This programs the SIM800 EEPROM with settings and URL endpoint. This is the same procedure as called in the code **ABCD2.x_SIM800Set.**

This code is meant for performing more advanced diagnosis on the SIM800 using AT commands.

Data collection:

Data is logged and transmitted with the following format:
TS(unix) , Vreference , Vsignal , Atenuation, BC (ug/m3) , RH(%) , T (ºC) , Flowrate (cc/min) , Vbatt

Data can be collected in 4 ways:

SD card – Data Is written to a file on the SD card with a filename corresponding to the instrument name. Data is logged at the sample rate.

USB Serial Streaming - With the Arduino driver installed on a computer, data is streamed through USB at the sample rate. This can be viewed through the serial monitor. Custom computer software can also be written to log this data from the COM port. Baud rate of the incoming data is 57600.

Networked data collection - If equipped with a SIM800 and SIM card, and the instrument is configured to log data to the database, the instrument will send out 1 minute (configurable) data in bursts every hour (also configurable).

Two-wire Serial Streaming – The external waterproof cable on the outside of the instrument houses 4 wires. Two cables are intended for charging and powering of the instrument (Blue(-) and Red (-). The other two cables are for transmitting serial data; yellow (Serial) and green (Ground). The baud rate of the incoming data is 57600. This serial data can be collected with an external logging device.

Button options:

There are **Two** button options which are accessed by holding down the Aux button and letting go during a light stage. To access the button, the instrument must be in logging mode (ie the Arduino light is flashing every two seconds).
To select an option, you hold down the button until the status light turns on and let go during one of the lighting stages.

1) Option 1 is accessed by letting go of button when light is lit initially. (0-2 seconds)
    a. **Attenuation reset** – Set first attenuation reading to power on attenuation (ie. Offset ATN to initially zero)
2) Option 2 is accessed by letting go of button when light is turned off after first being lit. (2-4 seconds)
    a. **Instrument reset** – turn off ATN offset, switch cell flip setting back to normal
3) Option 3 is accessed by letting go of button when light is turned back on again after being turned off. (>4 seconds)
    a. **Do nothing** - If holding for longer than 4 seconds, no action will be taken.

Troubleshooting:

Errors can be identified either through a solid red light during operation, verbose errors displayed on the database website, or verbosely streamed via serial. Errors are also saved to the SD card.

The database errors are the most explicit since the database runs algorithms to diagnose errors based on the quality of collected data in addition to the obvious errors that the instrument picks up.
If an error arises, it is best to connect the instrument to a computer and view the readout through the serial monitor to figure out what the issue is.

Errors displayed through Serial and network:

LED Saturated - The range of the photodetector is 2.5 volts. If the reading is at 2.5 V then all data collected is invalidated. This appears as a critical error.

Check to see that filters are properly installed in the cells. If problem persists, contact support

LED OFF – If the photodetector is not picking up and transmitted light, this will show up as a critical error.

Check LED cable connection. Ensure that that filters are not overly saturated

FR issue – Critical issue with the flow system.

A FR of ~ -121 indicates that the instrument was using a specific calibration based on an EEPROM calibration byte which got corrupted. Perform a flowrate calibration.
A FR of ~ -40 indicates that the flow is backwards. Check to makes sure the inlet and outlet of the pump is proper, and check that the wiring of the pump to the AUX board is also correct.
A FR >500 indicates an issue with the FR sensor. Ensure it is properly connected.
A FR ~ -6 indicates the pump is not running. Check to see if it is connected. Otherwise the pump may have a critical issue. Contact support

NAN ATN Reset Btn2 – This is a software bug which has been weeded out. It occurs due to the cumulative addition of ATN when a data reading becomes corrupt and NaN is added to the attenuation resulting in NaNs thereafter.

Reset the instrument using button option 2.
If the filters are freshly changed, proceed to reset the attenuation using button option 1. Otherwise replace the filters and reset the attenuation using button option 1

ADC error – This occurs when there is no power to the cell or a communication problem.

Verify the connection between the instrument and the AUX board.
Verify that there is power to the instrument (Battery power should be supplied to the instrument). Note that the instrument may appear to be

on when plugged in with a USB cable, but the entire instrument is not on unless the battery is on.

Network – This can be due to poor network reception, or due to corrupt settings on the SIM800.

Reprogram the SIM800 using the steps described in the SIM800 section.

Errors displayed on network only:

Noisy FR – if the STD of the FR is above 1, a warning will arise stating "Noisy FR". This is not necessarily always of concern because it can arise from two situations. If the flowrate really is noisy (maintenance then required), or if there was a sudden spike (a singular event in which a datapoint of FR spikes due to a momentary pump stall) which causes an elevated STD.

If it is a consistent issue, check to make sure that filters, springs, and washers are all properly installed in the cell.

Noisy FR bad pump - If the STD is above 10 however, a critical warning will arise stating "Noisy FR bad pump". This indicated that the pump or FR sensor is acting erratically.

This is most likely due to a faulty pump and thus the pump should be replaced. It can also be due to improper plumbing such as a cell not being plugged in correctly, or absence of washers in the cell.
Check plumbing to ensure there is no blockage
Check to ensure that there filters, washers, and springs are properly installed in the instrument cell.

Low Battery - If the instrument has a low battery (<10 V), this error arises as a warning.

Change or charge the battery

HIGH ATN – This warning arises when ATN > 80

Change or flip the cell.

Flip Error – If the database detects that the attenuation is continuously decreasing it suggests that either the instrument is improperly set to run in flipped mode, or the cell is running in the wrong orientation.

Check orientation of cell. If it is in the default orientation (thumbscrew towards door hinge), reset the instrument by selecting button option 2.
If running in flipped mode and the cell is oriented properly (thumbscrew away from door hinge), restart the instrument and set it to run in flipped mode by holding down the status button while turning on the instrument.