BS/RS Auxiliary Sensors

Mike and Stephanie
June 29, 2020

The following document is all the information I have on the HDF aux file created by Arts program during Mikes experimental data collection.

Ancillary General Information

Mike and I where talking about the names of things in the HDF aux file that Art created for the experimental data and for the in-water HDF file. So I thought I would write up a document to describe the data and other info. For these data the labjack channels are saved in a separate HDF file (experimental data). In the in-water data the same aux data will be save in the in-water HDF file with the images/tracks. This document will cover the experimental AUX file and the in-water HDF file.

There are 14 channels. Channel numbers start at 0.

These are the channel numbers I am assuming from the lab jack channel numbers.
AuxChannel: 0 1 2 3 4 5 6 7 8 9 10 11 12 13

I asked Art to get each measurement a code. They should move with the sensor if a sensor is moved to a different channel. This should reduce the need for verison numbers for the aux data.
AuxCode: 100 101 102 103 104 105 106 107 108 109 110 111 112 113

I asked Art to add a instrument serial number. So if we change out the sensor we can track that. These serial numbers are for a particular sensor on a BS or RS. If a new sensor is installed the serial number goes up by 1.
AuxSerialNum: 1 1 1 1 1 1 1 1 1 1 1 1 1 1

These are the names of each channels data and units provided by Art

			Powered by			
Description	Code	Channel	DC/DC	LabJack	SPEC	
Auxiliary mains voltage	100	0				
Auxiliary mains current	101	1	X		1	
Internal humidity temp	102	2				???
External heat sink temp	103	3		1		
CCD Base temp	104	4		1		
SBC heat sink temp	105	5		1		
Shutter block temp	106	6		1		
Case air temp	107	7	1			
Case air humidity	108	8	1			
Case moisture sensor 0%=dry	109	9	1			
Slit temperature	110	10		1		
Near-camera temperature	111	11		1		
Spectrograph internal humidity	112	12	1			
+5V supply and full scale ref.	113	13				

Aux board's DC/DC supply is the source of the 5 Vdc reference

See "Email from Mike on 5/18/2020, 11:09 PM" for more detail from Mike
The External Water Temperature thermistor was not wired up - for both BS & RS

The third internal Resonon thermistor s not connected. On the Resonon Hummer connector #1 is "Slit therm" and "Near Camera therm", which wire to Aux #10 & 11 (counting from zero)

The Resonon Hummer #2 has the "Internal Humidity sensor" at Aux#12, and the third Internal therm = "Under Cap" is not connected.

Mark thinks maybe Labjack has a multiplexor that can be added to get more aux channels (?). Labjack U6-PRO says it has a $150 multiplexer to add channels.

Question for Art:
Are the aux thermistors being read with full (24bit ?) resolution ? I.E. the ./readancin (in /home/user/src/labjack05) printed only 0.001 V resolution, for the stream of 6x thermistors @ Aux# 3,4,5,6,10,11 (counting from zero), which did not seem to be enough digits to see any converted temperature changes from a heated thermistor. Mark would like to see what the temperatures look like at two settings: 1.) gain 1x (10V range) at 21bit resolution, and 2.) gain 10x (1V range) at 18bit resolution. Just for the 6x thermistors in the string. "Ideal would be if #2 works because sample time is faster." Mark's last text: "I'm not sure if we can do this without the Art using the Labjack provided test program bit I doubt it."

I am now (May 2020) able to convert the raw aux data to converted which mostly match Art's math. After a Jun 2020 update the temperatures now ranges when heat gun is applied now look correct.

In an email from Mike on 6/24/2020, 9:05 PM he did an experiment to figure out 1) why the range of temperatures was so small even after a heat gun was use on the temp sensors and 2) to confirm which temp sensors (that he can see) are which. See the following section on this experiment.

Ancillary data description

Some of this is info from the in-water ancillary/auxilliary data but is valid for the experiment AUX_*.h5 data set as well.

First there are 14 text attributes which are the units of the scaled measurements from the ancillary labjack. These are just included once since they would be identical for each measurement.

```
Group '/raw/version0/ancillary'
      Attributes:
      'AuxChannel':  0 1 2 3 4 5 6 7 8 9 10 11 12 13
      'AuxCode':  100 101 102 103 104 105 106 107 108 109 110 111 112 113
      'AuxSerialNum':  1 1 1 1 1 1 1 1 1 1 1 1 1 1
      'AuxDescription000':  'Auxiliary mains voltage'
      'AuxScaledUnits000':  'V'
      'AuxDescription001':  'Auxiliary mains current'
      'AuxScaledUnits001':  'A'
      'AuxDescription002':  'Internal humidity temp'
      'AuxScaledUnits002':  'C'
      'AuxDescription003':  'External heat sink temp'
      'AuxScaledUnits003':  'C'
      'AuxDescription004':  'CCD Base temp'
      'AuxScaledUnits004':  'C'
      'AuxDescription005':  'SBC heat sink temp'
      'AuxScaledUnits005':  'C'
      'AuxDescription006':  'Shutter block temp'
      'AuxScaledUnits006':  'C'
      'AuxDescription007':  'Case air temp'
      'AuxScaledUnits007':  'C'
```

```
'AuxDescription008':  'Case air humidity'
 'AuxScaledUnits008':  '%'
 'AuxDescription009':  'Case moisture sensor 0%=dry'
 'AuxScaledUnits009':  '%'
 'AuxDescription010':  'Spectrograph internal temp 1'
 'AuxScaledUnits010':  'C'
 'AuxDescription011':  'Spectrograph internal temp 2'
 'AuxScaledUnits011':  'C'
 'AuxDescription012':  'Spectrograph internal humidity'
 'AuxScaledUnits012':  '%'
 'AuxDescription013':  '+5V supply and full scale ref.'
 'AuxScaledUnits013':  'V'
 'Nancillary':  8
```

AuxCode = Like the Analog Channel Code in the MO3 files. That way if they change a sensor from channel 0 to channel 5, or add another sensor to an empty channel etc my programs can still find them in an automated way. So for codes the -999 means there is no sensor connected. Sensors

```
                                                                           Codes
#define AMAINV    0  Auxiliary mains V */                                   100
#define AMAINI    1  Auxiliary mains current */                             101
#define xxxxxx    2  Internal humidity temp */                              102
#define AEXTT     3  External heat sink temp */                             103
#define ACCDT     4  CCD Base temp */                                       104
#define AFALCONT  5  SBC heat sink temp */                                  105
#define ASHUTT    6  Shutter block temp */                                  106
#define ACASET    7  Case air temp */                                       107
#define ACASERH   8  Case air humidity */                                   108
#define AMOISTSEN 9  Case moisture sensor */                                109
#define ASGT1    10  Slit temp (T1)  */                                     110
#define ASGT2    11  Near-Camera temp (T2) */                               111
#define ASGRH    12  Spectrograph internal humidity */                      112
#define AREFV    13  +5V supply and full scale reference */                 113
```

AuxSerialNum: 1 1 1 1 1 1 1 1 1 1 1 1 1 1.
Some kind of serial number for each sensor. All sensors start at 1. So if a sensor is replaced we can change serial number by one. This allows for changes in the conversion equation with new sensors. And help keep track of config changes so they can be matched with data changes.

AUX HDF file listing of groups and data for in-water hdf files
Listing of one of the data groups from AUX_20191119.h5. There is one group for each image Mike takes. (aux_cvn_data_test_.m)

```
D:\zflora\mldata2\mobyrefresh\characterization\moby-l285\RS04cal\day01
Group '/raw/version0/ancillary/ancillary001'
   Attributes:
      'MeasurementTime':  '20191119_061715_GMT'
      'AuxVrawNAvg':  10
      'AuxVrawMean': 2.706574 0.009836 2.326747 0.544089 0.443680 0.364854
      0.273633 2.384713 2.479065 5.023591 0.181783 0.090086 1.962680 5.024953
      'AuxVrawStdev':  0.000286 0.000044 0.013658 0.000118 0.012065 0.000134
      0.000121 0.015847 0.001047 0.032277 0.000075 0.000039 0.014250 0.032424
      'AuxScaled':  13.532870 -14.940984 25.122423 24.999354 25.037703
      25.014547 25.013459 25.118492 49.335084 0.027111 25.013724 25.016531
      61.226767 5.024953
      'GoesWithImage':  'd20191119001.fits'
```

```
'GoesWithImageData':  0 0
'CoolOnFlag':  1
'CoolTargTempC':  -60
'CoolCurrentTempC':  -59
'CoolAndorStatusCode':  20036
'CoolTempStableFlag':  1
```

All measurements are made after data collection. The following is a description of each Attribute in the ancillary group.

MeasurementTime: time stamp (yyyymmdd_HHMMSS_GMT)
AuxVrawNAvg: Number of times the labjack was read to generate average voltages (N)
AuxVrawMean: Mean of AuxVrawNAvg measurements on each labjack input channel
AuxVrawStdev: Standard Deviation of N measurements on each labjack input channel
AuxScaled: AuxVrawMean converted to engineering units (V, A, C, percent etc..)
GoesWithImage: The name of the image dataset corresponding to this ancillary measurement or 'N/A' if no image taken at the same time (e.g. during cool down)
CoolOnFlag: 1 = cooler is ON, 0 = cooler is OFF
CoolTargTempC: target temperature for cooler (C)
CoolCurrentTempC current temperature for cooler (C)
CoolAndorStatusCode: a code that the camera returns when you poll the temperature. 20037 = temp not stable, 20036 = temp is stable, I can send you the other values but basically any other number indicates an error. See Andor_Software_Development_Kit.pdf for more details (GetTemperature function (page 183)).
CoolTempStableFlag: 1 = cooler is stable (code 20036), 0 = cooler is not stable (Code 20037)

For the ancillary measurements that correspond to images, MeasurementTime will be equal to the end time of the image. So you could align based on /raw/version0/ancillary/ancillaryX/MeasurementTime and /raw/version0/images/*/GMT_End. The group /raw/version0/images only exists in the in-water files since the experimental files are fits files.

Converting the raw aux data (Jun 2020 Updated)

So on 6/24/2020, 9:05 PM Mike sent an email with an experiment to test the BS/RS temperature sensors (the ones he could see). He was concerned with the range of the converted data. When he put a heat gun on the sensor the voltages changed a lot but the converted temperatures where right around 25 C an changed less than a degree. After looking at the data with the team Ken found that the conversions where being done in Celsius not Kelvin and that fixed the conversions.

Using Art's ancillaryscaleV.c program to convert the raw aux dat to converted (May 2020, w/ Jun update)

C Var	Chan	chancode	Channel Name
AMAINV	0	100	Auxiliary mains V
AMAINI	1	101	Auxiliary mains current
AINTRHT	2	102	Internal humidity temp
AEXTT	3	103	External heat sink temp
ACCDT	4	104	CCD Base temp
AFALCONT	5	105	Falcon heat sink temp or SBC heat sink temp
ASHUTT	6	106	Shutter block temp
ACASET	7	107	Case air temp
ACASERH	8	108	Case air humidity
AMOISTSEN	9	109	Case moisture sensor
ASGT1	10	110	Spectrograph internal temp 1
ASGT2	11	111	Spectrograph internal temp 2
ASGRH	12	112	Spectrograph internal humidity
AREFV	13	113	+5V supply and full scale reference

T0 should be in Kelvin not in Celsius

```
T0 = 25+273;   %/* K, room temperature */
B  = 3950;
R0 = 10000;
m  = 30.0 ./ 5.0;
b  = -15.0;
I  = 10E-6; %/* 10 uA source */
scaleV = AREFV;
```

Convert Auxiliary mains V

```
AMAINVcnv = AMAINV*5
```

Convert Auxiliary mains current

```
m = 30.0 ./ 5.0;
b = -15.0;
AMAINIcnv = AMAINI.*m + b
```

Converting the temperatures from the resistor series

My vars	Arts vars	Long names	channel num
RAW(:,1)	AEXTT	External heat sink temp	3
RAW(:,2)	ACCDT	CCD Base temp	4
RAW(:,3)	AFALCONT	SBC heat sink temp	5
RAW(:,4)	ASHUTT	Shutter block temp	6
RAW(:,5)	ASGT1	Spectrograph internal temp 1	10
RAW(:,6)	ASGT2	Spectrograph internal temp 2	11

```
% AN3  volts = AN3-AN4
% AN4  VOLTS = AN4-AN5
% AN5  VOLTS = AN5-AN6
% AN6  VOLTS = AN6-AN10
% AN10 VOLTS = AN10-AN11
% AN11 VOLTSIs read directly
```

First subtract to get the correct readout

```
for k = 1:5 % We don’t subtract anything from RAW(:,6)
  Vnet(:,k) = RAW(:,k)-RAW(:,k+1);
end
```

Then convert each sensor

```
for k = 1:6
  R = Vnet(:,k) ./ I;
  Tinv = 1./T0 + 1./B .* log(R ./ R0);
  CNV(:,k) = (1./Tinv)-273;
end
```

```
Next Channel 2 - Internal humidity temp
R = 10000 ./ (scaleV ./ AINTRHT);
Tinv = 1./T0 + 1./B .* log(R ./ R0);
AINTRHTcnv = (1./Tinv);

Channel 7 - Case air temp
R = 10000 ./ (scaleV ./ACASET);
Tinv = 1./T0 + 1./B .* log(R ./ R0);
ACASETcnv = (1./Tinv)-273;

Channel 8 - case air humidity
ACASERHcnv = 100.*(ACASERH./scaleV);

Channel 9 - case moisture sensor HI=DRY=0% wet LO=WET=100% wet
AMOISTSENcnv = 100 - (AMOISTSEN./scaleV) .*100;

Channel 12 - spectrograph internal humidity
sensRH   = -0.1515 + ASGRH./ ( scaleV .* 0.00636);
specT    = nanmean(CNV(:,5:6)')';
ASGRHcnv = (sensRH ./ (1.0546 - 0.00216 .* specT))-273;
```

Mikes Jun 2020 heat gun test dir

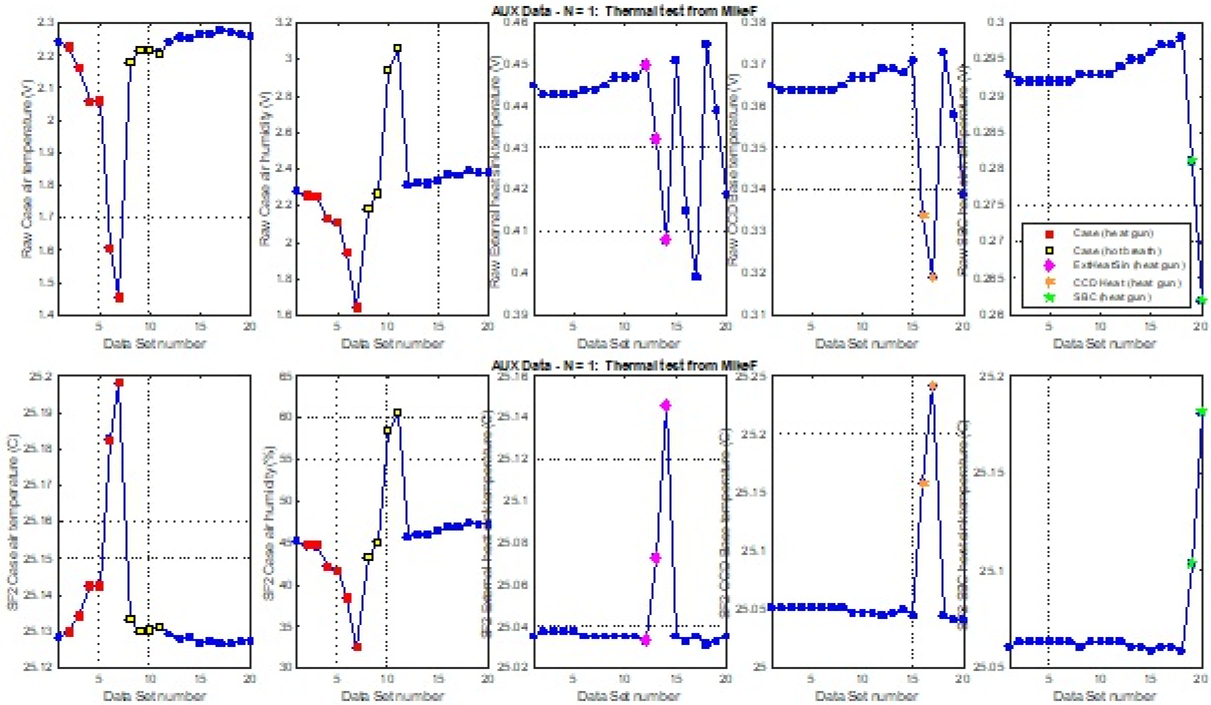
D:\zflora\mldata2\mobyrefresh\characterization\Hawaii-2020-03\RS04\20200619_aux_therm_chk1

Program: readtxtaux_

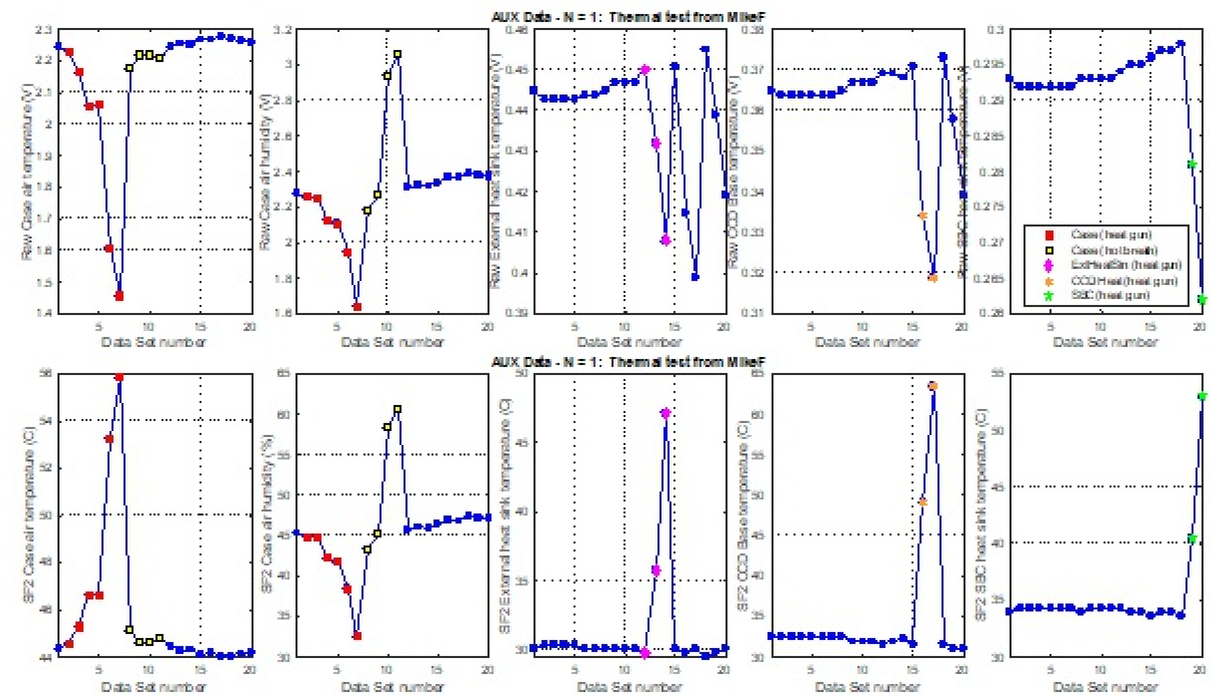
Mike did 5 different checks of 4 thermistors. The blew a heat gun on 4 thermistors. The hot breath was to increase the humidity and test the RH sensor. Test 3-5 might have a smaller response because they are connected to a heat sink.

1.) heat gun on %RH / degC pen
2.) hot breath on %RH/degC pen
3.) heat gun @ ext heat sink therm
4.) heat gun @ CCD heat sink therm
5.) heat gun @ SBC heat sink therm

This plot showing the data before it was fixed. You can see the change in voltage on the top graphs. The color dots tell which sensor had the heat gun or hot breath. The converted data changes where very small and do not make sense when I heat gun is pointed at the temperature sensor.



After the changes to the converting programs to make the conversion in Kelvin the graphs look much better. The temperatures look correct and the ranges make more sense.



Labjack info
From a doc Mike created to describe the file he created from the labjack data before Art wrote his program.Turns out very little of the "how to convert the data" is useful in the converting of the HDF file aux data.

Mikes doc = Aux_data_via_LabJack_for_MOBY-Refresh(rev2).pdf

MOBY-Refresh instruments will employ internal U6 DAQs for control and acquisition.

The U6 5 Vdc power source (supplied via USB cable connection) is used to supply voltage-divider input for 5x thermistors for temperature measurements, and 2x humidity sensors for %RH measurements. The Resonon instruments have 3x internal temperature sensors and 1x internal humidity sensor. A t l%RH+°An external %RH + °C sensor monitors ambient conditions, and another external thermistor is available to attach to another device – ex. Fiber splitter. There are 2x not-used channels wired and available in the U6 connector box for future use.

All thermistors are wired in resistive voltage divider circuits, following the equation:

Vout = Vin * (R2 / (R1 + R2)), where
Vout = measured output Voltage across the 10 kΩ NTC thermistor (i.e. temperature probe)
Vin = input supply voltage, measured as Chan 1
R1 = NTC resistance (Ohm), nominal 10 kΩ at 25°C
R2 = batch resistance (Ohm), 10 kΩ ± 0.1%, 5 ppm temp. coef. PTF5610K000BZE

So that the unknown NTC resistance can be solved from the above equation as:
R1 = R2 * ((Vin / Vout) – 1)

Labjack website for the U6 : https://labjack.com/products/u6

LabJack U6-PRO

Multifunction DAQ - USB

The U6 incorporates a wide range of features including our highest performance 24-bit analog inputs with on-board instrumentation amplifier.

$369

labjack.com/u6

I/O Features

- Analog input resolution as low as **1µV noise-free**
- Analog input ranges: ±10V, ±1V, ±0.1V and ±0.01V
- Expand to **84 analog inputs** with $150 add-on
- 16-bit high-speed ADC (up to 50kHz)
- **24-bit** low-speed ADC
- 14 analog inputs built-in
- 20 digital I/O
- Watchdog system
- Up to 2 counters
- 2 analog outputs (12-bit, 0-5V)
- Serial protocols: SPI, I2C, and more ...
- Up to 4 PWM, quadrature, pulse width, and more ...
- Thermocouples, load cells, bridges, and more ...
- Industrial temperature range (-40 to 85°C)

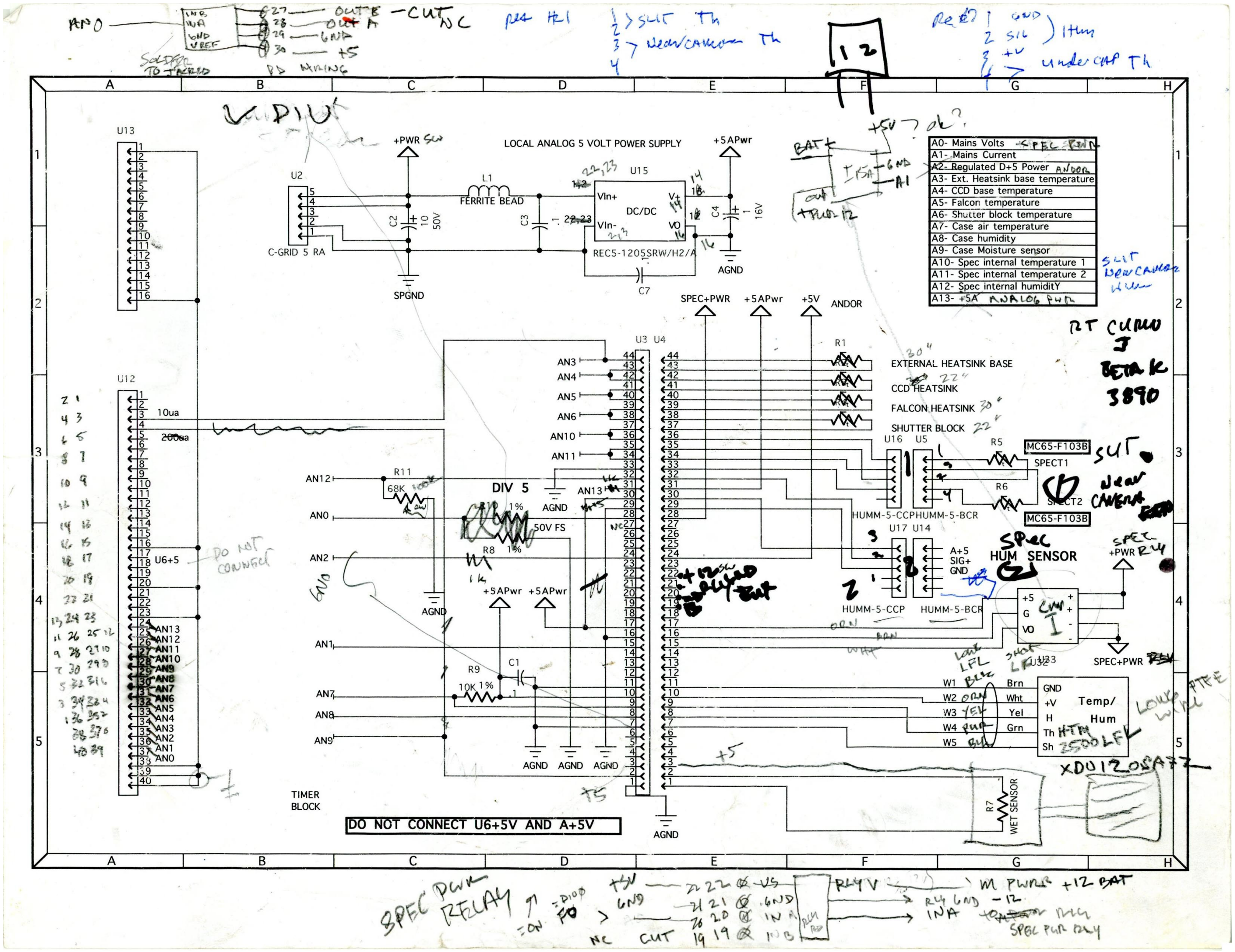
Other Highlights

- Each purchase includes **lifetime support**
- **Free applications** to configure, test, and log data to file
- Free examples: **C/C++**, **C#**, **Delphi**, **Java**, **LabVIEW**, **Matlab**, **Python**, **VB.NET** and more...
- Free UD Library – Wraps the low-level protocol and USB driver layer for ease-of-use
- Expansion boards – Add ±10V DACs, 4-20 mA inputs, terminal boards, relay boards and more...

"I really do love your products … They are first class for coach price, and your customer service is what every company should aspire to have."

-Brad
Neuro-Test Inc

Latest / updated version of the Aux wiring diagram



Email from Mike on 14 May 2020 @ 11:20 pm

Attached here is the latest / updated version of the Aux wiring diagram. All the hand-written notes are not yet updated on the digital version...

From this diagram "one can see that" the third internal Resonon thermistor is not connected. On the Resonon Hummer connector #1 is "Slit therm" and "Near Camera therm", which wire to Aux #10 & 11 (counting from zero) I.E. see diagram notes in blue-ink. The Resonon Hummer #2 has the "Internal Humidity sensor" at Aux#12, and the third Internal therm = "Under Cap" is not connected. Mark thinks maybe Labjack has a multiplexor that can be added to get more aux channels (?).

<u>Sensor Location information</u>

Channel 000: Auxiliary mains voltage (V)

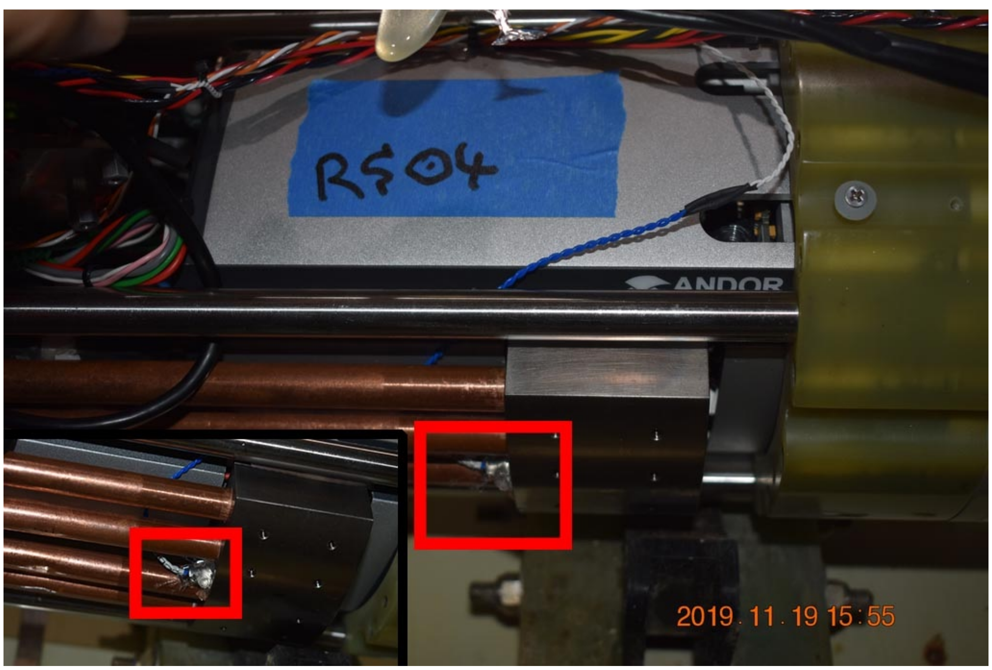
Channel 001: Auxiliary mains current (A)

Channel 002: Internal humidity temp ©

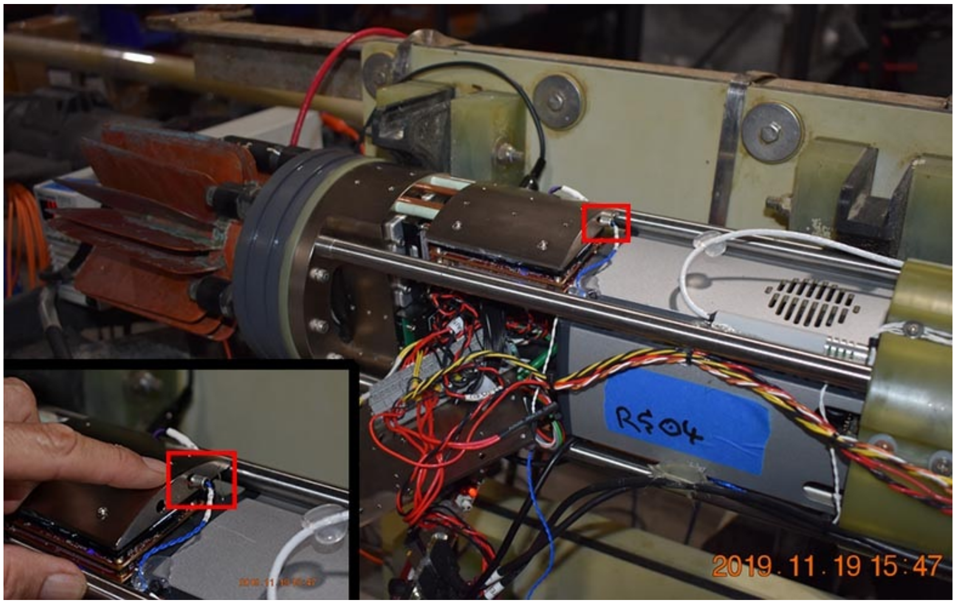
We are still not sure of this sensors location

<u>Channel 003:</u> External heat sink temperature: at the end of the housing where the external cooling fins attach is another thermistor. Photos = 2019111905, 06=External-heat-sink-thermistor.JPG

<u>Channel 004:</u> CCD Base temperature: at the side of the Andor camera, a portion of the housing is cut out to expose the heat sink at the base of the CCD. This heat sink mates with cooling "fingers" which move heat to the external cooling fins. Photos = 2019111903, 04=CCD-base-thermistor.JPG



<u>Channel 005:</u> SBC heat sink temperature: SBC = Single Board Computer, i.e. the Falcon computer. The SBC is really two boards, and there is a heat sink attached to the boards, and the heat-sink has heat-pipes to carry said heat to the cooling fins outside of the housing. On the computer's heat sink is a thermistor (duh). I took photos of this: 2019111901, 02=SBC-heat-sink-thermistor.JPG

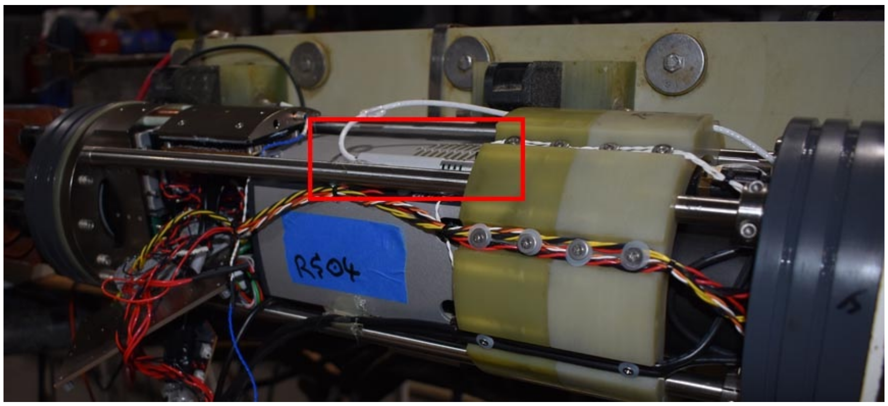


Thermistor in the heat sink (the silver curved-top piece) which is bolted to the SBC/Falcon boards.

<u>Channel 006:</u> Shutter block temperature: this thermistor is at the shutter block, which right now is covered by 1/2 of the water housing. But, Mark said he may need to remove this 1/2 of the water housing tomorrow, and if so, iShall try to photograph said thermistor!

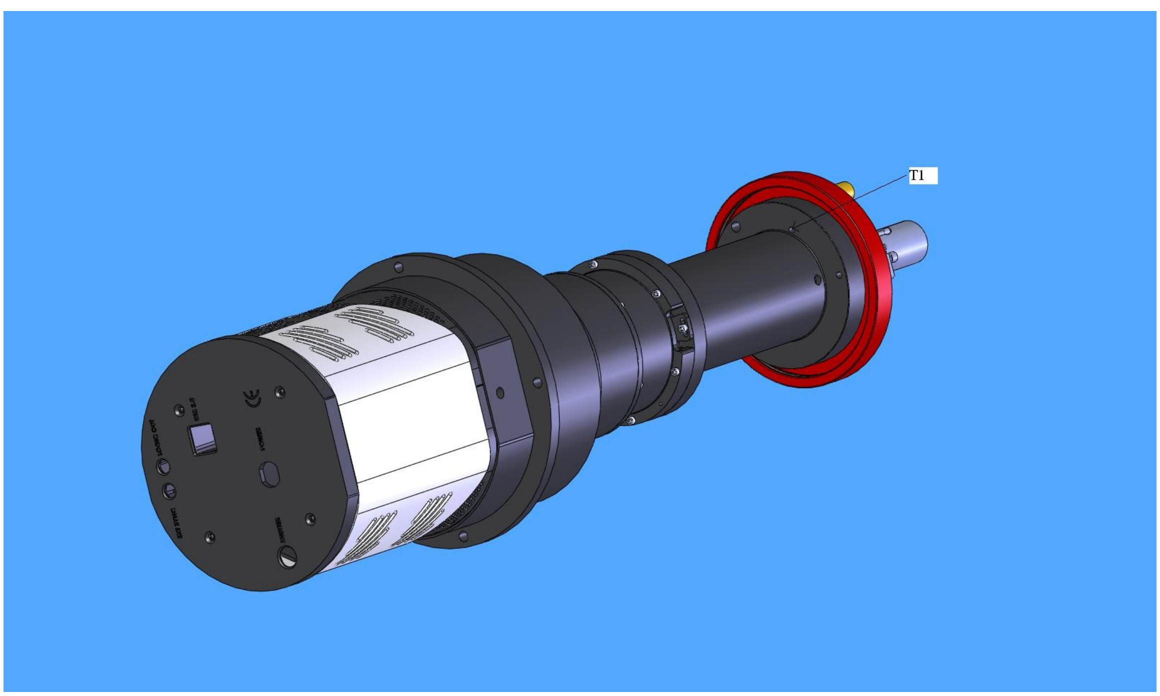
<u>Channel 007:</u> Case air temp (C) & <u>Channel 008:</u> Case air humidity (%)

The Chan007 "Case air temp (C)" and Chan008 "Case Air Humidity (%)" are in one unit, which is seen in the photo below = the gray pen-like gizmo with the silver tip at right-side and white cable at left-side, glued down just next to the air vent at the side of the Andor camera housing.

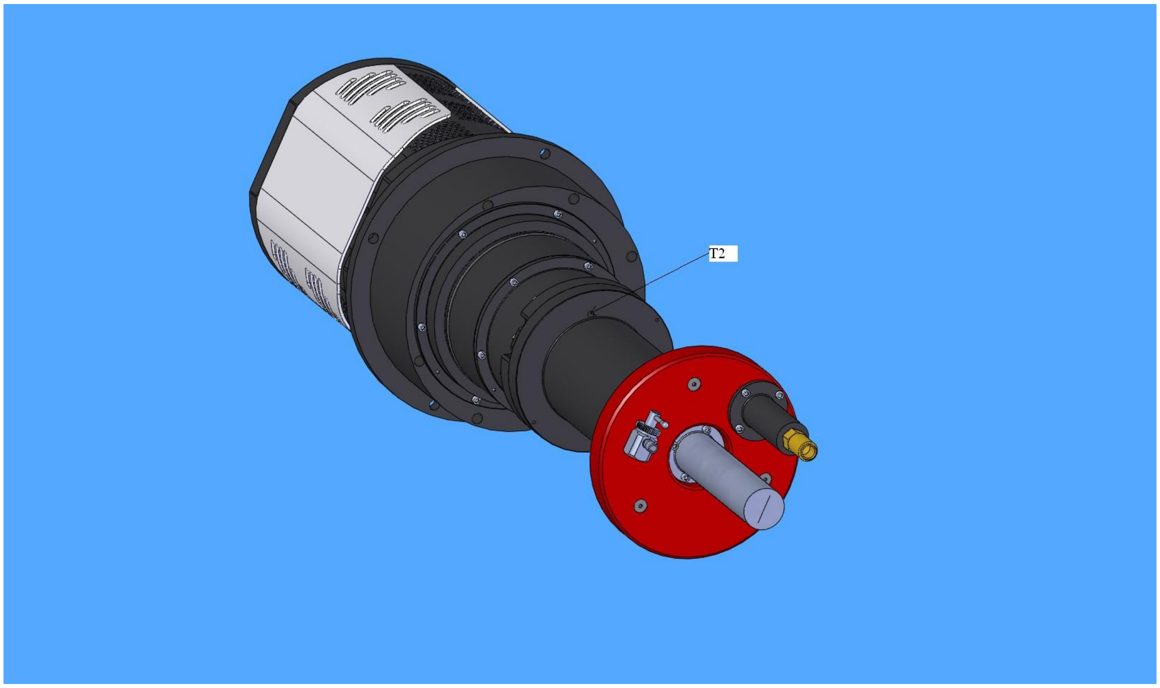


<u>Channel 009:</u> Case moisture sensor 0%=dry (%)

<u>Channel 010:</u> Spectrograph internal temp 1 (C) - Slit Thermistor - T1
The 3x Chan10,11,12 "Spectrograph..." sensors are inside the Resonon Spectrograph, so we will never have photos of these.



<u>Channel 011:</u> Spectrograph internal temp 2 (C) - Near-Camera Thermistor - T2
The 3x Chan10,11,12 "Spectrograph..." sensors are inside the Resonon Spectrograph, so we will never have photos of these.



<u>Channel 012:</u> Spectrograph internal humidity (%) - Humidity sensor

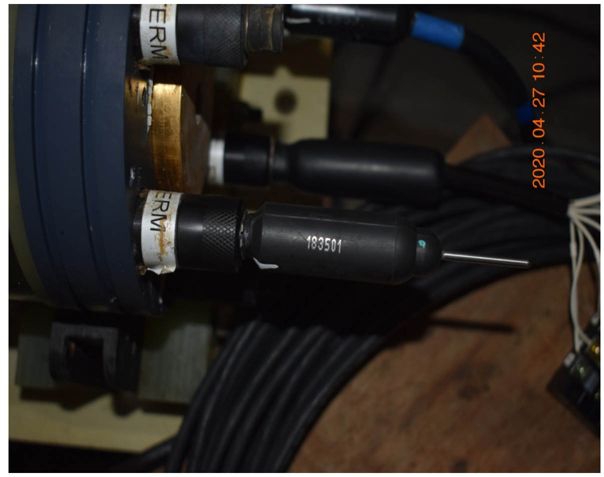
This and the 3x Chan10,11,12 "Spectrograph..." sensors are inside the Resonon Spectrograph, so we will never have photos of these.

Channel 013: +5V supply and full scale ref. (V)

Web location of photos
http://data.moby.mlml.calstate.edu/mobyrefresh/timeseries/photos/20191119_M267_post-cal/photos.html

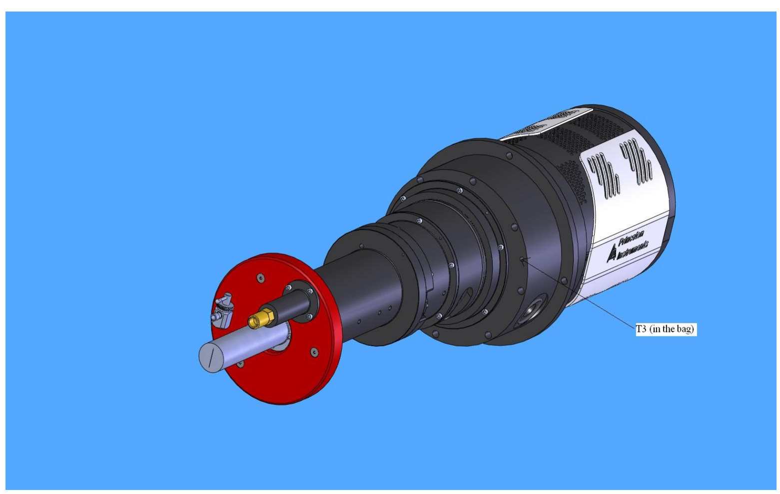
External (water) Temperature

According to email from Mike on 5/14/2020, 11:20 PM the external water temp is not hooked up because the labjack ran out of channels.

	
Photos from Mike in and email on 5/7/2020, 6:47 PM	Photos of External (water) Temperature

Under-Cap Thermistor

T3 ?? - Mike sent 3 images of the location of temperature sensors in the specs. This is of T3 (in the bag). The only sensor left is the Under-Cap Thermistor from the Wiring diagram. Now we just need to know if T3 is the Under-Cap Thermistor or not. And which sensor the Under-Cap Thermistor maps to by Channel number.



Emails from Mike say this is not hooked up to anything.

On 7 May 2020 Art emailed me the c program he uses to convert the data

In June we found that all the conversions should be done in Kelvin so T0 should have 273 added and all the converted data should have 273 subtracted.

Name of file is ancillaryscaleV.c
Pwd: D:\zflora\mldata\htmldir\mobyrefresh\timeseries\configs\components\aux_sensors

dir with the files and test m-files in it
D:\zflora\mldata2\mobyrefresh\characterization\Hawaii-2020-02\20200507_Art_email
Program: test_aux_graphs.m

```
#ifndef _LJ_H_
#include "lj.h"
#endif

int ancillaryscaleV(struct ljancillary* A)
/*==============================================================================
=
/ Scale ain voltages to whatever their real values are.
/ call ancillarygetain before calling this function
/
/ Art Gleason 1 Aug 2016
/ - Original version
/ Art Gleason 20 Feb 2019
/ - Modified to use new struct ljancillary
/==============================================================================*
/
{
    int errcode = LJOK;
    int ix;
    /*---5V reference scale = AREFVCODE---*/
    double scale5v;
    /*---general linear coefficients---*/
    double m, b; /* slope, intercept */
    /*---spectrograph RH---*/
    double sensRH, specT;
    /*---thermistor series---*/
    int Nseries=6;  /* make sure seriesix is length Nseries */
    int seriesix[6];/* make sure seriesix is length Nseries */
    double Vnet;
    double I, R;
    double Tinv;
    double T0 = 25; /* C, room temperature */
    double B  = 3950;
    double R0 = 10000;
    /*---what channels are which sensors?---*/
    /* These used to be defined in lj.h but are now defined in a config file
    /  and loaded by fillljcfg */
    int AMAINV      = LJNULL;
    int AINTRHT     = LJNULL;
    int AMAINI      = LJNULL;
    int AEXTT       = LJNULL;
    int ACCDT       = LJNULL;
    int AFALCONT    = LJNULL;
    int ASHUTT      = LJNULL;
    int ACASET      = LJNULL;
```

```
int ACASERH     = LJNULL;
int AMOISTSEN   = LJNULL;
int ASGT1       = LJNULL;
int ASGT2       = LJNULL;
int ASGRH       = LJNULL;
int AREFV       = LJNULL;

/*---Initialization----------------------------------------------*/
for(ix=0; ix<A->nancain; ix++){
  /*---initialize all scaled values to 0---*/
  A->scaleV[ix] = LJNULL;
  strcpy( A->units[ix], "unconverted" );
  strcpy( A->description[ix], "N/A" );
  /*---initialize which channels are which--*/
  switch( A->anccode[ix] ){ /* anccode was set in fillljcfg() */
      case AMAINVCODE:
        AMAINV = ix;
        strcpy( A->units[AMAINV], "V" );
        strcpy( A->description[AMAINV], "Auxiliary mains voltage" );
        break;
      case AINTRHTCODE:
        AINTRHT = ix;
        strcpy( A->units[AINTRHT], "C" );
        strcpy( A->description[AINTRHT], "Internal humidity temp" );
        break;
      case AMAINICODE:
        AMAINI = ix;
        strcpy( A->units[AMAINI], "A" );
        strcpy( A->description[AMAINI], "Auxiliary mains current" );
        break;
      case AEXTTCODE:
        AEXTT = ix;
        strcpy( A->units[AEXTT], "C" );
        strcpy( A->description[AEXTT], "External heat sink temp" );
        break;
      case ACCDTCODE:
        ACCDT = ix;
        strcpy( A->units[ACCDT], "C" );
        strcpy( A->description[ACCDT], "CCD Base temp" );
        break;
      case AFALCONTCODE:
        AFALCONT = ix;
        strcpy( A->units[AFALCONT], "C" );
        strcpy( A->description[AFALCONT], "SBC heat sink temp" );
        break;
      case ASHUTTCODE:
        ASHUTT = ix;
        strcpy( A->units[ASHUTT], "C" );
        strcpy( A->description[ASHUTT], "Shutter block temp" );
        break;
      case ACASETCODE:
        ACASET = ix;
        strcpy( A->units[ACASET], "C" );
        strcpy( A->description[ACASET], "Case air temp" );
        break;
      case ACASERHCODE:
        ACASERH = ix;
        strcpy( A->units[ACASERH], "%" );
        strcpy( A->description[ACASERH], "Case air humidity" );
        break;
      case AMOISTSENCODE:
```

```
            AMOISTSEN = ix;
            strcpy( A->units[AMOISTSEN], "%" );
            strcpy( A->description[AMOISTSEN], "Case moisture sensor 0%=dry"
);
            break;
          case ASGT1CODE:
            ASGT1 = ix;
            strcpy( A->units[ASGT1], "C" );
            strcpy( A->description[ASGT1], "Spectrograph internal temp 1" );
            break;
          case ASGT2CODE:
            ASGT2 = ix;
            strcpy( A->units[ASGT2], "C" );
            strcpy( A->description[ASGT2], "Spectrograph internal temp 2" );
            break;
          case ASGRHCODE:
            ASGRH = ix;
            strcpy( A->units[ASGRH], "%" );
            strcpy( A->description[ASGRH], "Spectrograph internal humidity" );
            break;
          case AREFVCODE:
            AREFV = ix;
            strcpy( A->units[AREFV], "V" );
            strcpy( A->description[AREFV], "+5V supply and full scale ref." );
            break;
      }
    }
    /*---other init---*/
    scale5v = A->ain[AREFV];
    A->scaleV[AREFV] = scale5v;

    /*---mains voltage---*/
    A->scaleV[AMAINV] = A->ain[AMAINV] * 5;

    /*---mains current---*/
    //m = 30.0 / scale5v;
    m = 30.0 / 5.0;
    b = -15.0;
    A->scaleV[AMAINI] = A->ain[AMAINI] * m + b;

    /*---temps from resistor series---*/
    seriesix[0] = AEXTT;
    seriesix[1] = ACCDT;
    seriesix[2] = AFALCONT;
    seriesix[3] = ASHUTT;
    seriesix[4] = ASGT1;
    seriesix[5] = ASGT2;
    for(ix=0; ix<Nseries; ix++){
      if( ix<Nseries-1 ){
        Vnet = A->ain[seriesix[ix]] - A->ain[seriesix[ix+1]];
      }
      else{
        Vnet = A->ain[seriesix[ix]];
      }
      I = 10E-6; /* 10 uA source */
      R = Vnet / I;
      Tinv = 1/T0 + 1/B * log(R / R0);
      A->scaleV[seriesix[ix]] = (1/Tinv);
    }

    /*---internal humidity temp---*/
```

```
    R = 10000 / (scale5v / A->ain[AINTRHT]);
    Tinv = 1/T0 + 1/B * log(R / R0);
    A->scaleV[AINTRHT] = (1/Tinv);

    /*---case air temp---*/
    R = 10000 / (scale5v / A->ain[ACASET]);
    Tinv = 1/T0 + 1/B * log(R / R0);
    A->scaleV[ACASET] = (1/Tinv);

    /*---case air humidity---*/
    A->scaleV[ACASERH] = (A->ain[ACASERH] / scale5v) * 100.0;

    /*---case moisture sensor HI=DRY=0% wet LO=WET=100% wet---*/
    A->scaleV[AMOISTSEN] = 100.0 - (A->ain[AMOISTSEN] / scale5v) * 100.0;

    /*---spectrograph internal humidity---*/
    /* From instrument datasheet :
    /  VOUT=(VSUPPLY)(0.00636(sensor RH) + 0.1515), typical at 25 C
    /  Temperature compensation
    /  True RH = (Sensor RH)/(1.0546 - 0.00216T), T in C
    / */
    sensRH = -0.1515 + A->ain[ASGRH] / ( scale5v * 0.00636);
    specT  = (A->scaleV[ASGT1] + A->scaleV[ASGT2]) / 2;
    A->scaleV[ASGRH] = sensRH / (1.0546 - 0.00216 * specT);

    /*---print if debugging---*/
    if( A->debug == 1 ){
      printf("---Mean and Scaled values---\n");
      for(ix=0; ix<A->nancain; ix++){
        printf("AIN%d = %6.3f V = %6.3f %s (%s)\n", ix, A->ain[ix],
A->scaleV[ix],
                 A->units[ix], A->description[ix]);
      }
    }

    return errcode;
}
```

AUX setup in the spec config files

Each BS or RS spectrograph has its own specification file and in the file each cfg change has its own section. The one I am working on currently is RS04 and cfg007 and cfg008 for the Hawaii-2020-03 data.

Dir: D:\zflora\mldata2\mobyrefresh\characterization\Hawaii-2020-03\RS04
Program: exp_savehdf5aux_ and RS04_ cfg 007

This is an example of the Aux data config info in the RS04_.m config file. I know Arts hdf file already has the channel, code and serial number but these will be used to make sure my and Arts info are correct. And if one or the other is not correct to make sure the file is corrected.

```
  % Auxilliary Data ------------------------------------------------
  FNAME = 'AUX';
  SET = 1;
  PARAMS.(FNAME).indexs            = [1];
  PARAMS.(FNAME)(SET).dates        = [datenum(2020,05,16) now];
 % Cal Date - Matlab date number
```

```
  PARAMS.(FNAME)(SET).cfgchecked   = 1;
 % I've checked the cfg again Arts and we agree
% LabJack Channel number
  PARAMS.(FNAME)(SET).channel      = [  0   1   2   3   4   5   6   7   8   9  10  11  12  13];
 % Sensor Code
  PARAMS.(FNAME)(SET).code         = [100 101 102 103 104 105 106 107 108 109 110 111 112 113];
 % Sensor S/N
  PARAMS.(FNAME)(SET).sn           = [  1   1   1   1   1   1   1   1   1   1   1   1   1   1];
 % Bad sensor
  PARAMS.(FNAME)(SET).badsenor     = [  0   1   0   0   0   0   0   0   0   0   0   0   0   0];
 % Printing groups
  PARAMS.(FNAME)(SET).prntgrp      = [  1   2   3   4   6   4   4   4   3   5   6   6   3   1];
 % What powers the sensor
  PARAMS.(FNAME)(SET).powerby      = [  0   3   0   2   2   2   2   1   1   1   2   2   1   0];
  %0 = unknown or NaN, 1 = DC/DC, 2 = LabJack, 3 = Spec
  PARAMS.(FNAME)(SET).cnv2use      = [  2   2   2   2   2   2   2   2   2   2   2   2   2   2];
 % Which converted data to use 1= Art, 2 = Sflora
 % I added the cnv2use so in the case where Arts conversions are wrong I can tell the program to
  use mine not Arts.

  % The data must be converted in this order-
  C = 1;
  PARAMS.(FNAME)(SET).cnv(C).global   = 'global Vaux Xdat;indx = find(Vaux(34,:) == 113 &
Vaux(35,:) == 200); scaleV = Xdat(:,indx);T0 = 25+273; B  = 3950; R0 = 10000;  I = 10E-6;';
  C = C + 1;

  PARAMS.(FNAME)(SET).cnv(C).dscs     = 'Auxiliary mains voltage';
  PARAMS.(FNAME)(SET).cnv(C).units    = 'V';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow  = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode  = [100 200];
  PARAMS.(FNAME)(SET).cnv(C).estr     = 'CNV = AUX.*5;';
  PARAMS.(FNAME)(SET).cnv(C).cnvcode  = [100 202];
  C = C + 1;

  PARAMS.(FNAME)(SET).cnv(C).dscs     = 'Auxiliary mains current';
  PARAMS.(FNAME)(SET).cnv(C).units    = 'A';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow  = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode  = [101 200];
  PARAMS.(FNAME)(SET).cnv(C).estr     = 'CNV = AUX.*(30.0./5.0) + -15;';
  PARAMS.(FNAME)(SET).cnv(C).cnvcode  = [101 202];
  C = C + 1;

  PARAMS.(FNAME)(SET).cnv(C).dscs     = 'Internal humidity temp';
  PARAMS.(FNAME)(SET).cnv(C).units    = 'C';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow  = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode  = [102 200];
  PARAMS.(FNAME)(SET).cnv(C).estr     = 'R = 10000 ./ (scaleV ./AUX);Tinv = 1./T0 + 1./B .* log(R
./ R0);CNV = (1./Tinv)-273;';
  PARAMS.(FNAME)(SET).cnv(C).cnvcode  = [102 202];

  C = C + 1;
  PARAMS.(FNAME)(SET).cnv(C).dscs      = 'External heat sink temp';
  PARAMS.(FNAME)(SET).cnv(C).units     = 'C';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow   = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode   = [103 200];
  PARAMS.(FNAME)(SET).cnv(C).auxcode2  = [104 200];
  PARAMS.(FNAME)(SET).cnv(C).estr      = 'Vnet = AUX - AUX2;  R = Vnet./I; Tinv = 1./T0 + 1./B .*
log(R ./ R0);CNV = (1./Tinv)-273;';
  PARAMS.(FNAME)(SET).cnv(C).series    = 1;
  PARAMS.(FNAME)(SET).cnv(C).cnvcode   = [103 202];
  C = C + 1;
  PARAMS.(FNAME)(SET).cnv(C).dscs      = 'CCD Base temp';
  PARAMS.(FNAME)(SET).cnv(C).units     = 'C';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow   = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode   = [104 200];
  PARAMS.(FNAME)(SET).cnv(C).auxcode2  = [105 200];
  PARAMS.(FNAME)(SET).cnv(C).estr      = 'Vnet = AUX - AUX2;  R = Vnet./I; Tinv = 1./T0 + 1./B .*
log(R ./ R0);CNV = (1./Tinv)-273;';
  PARAMS.(FNAME)(SET).cnv(C).series    = 1;
  PARAMS.(FNAME)(SET).cnv(C).cnvcode   = [104 202];
  C = C + 1;
  PARAMS.(FNAME)(SET).cnv(C).dscs      = 'SBC heat sink temp';
  PARAMS.(FNAME)(SET).cnv(C).units     = 'C';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow   = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode   = [105 200];
  PARAMS.(FNAME)(SET).cnv(C).auxcode2  = [106 200];
```

```
  PARAMS.(FNAME)(SET).cnv(C).estr      = 'Vnet = AUX - AUX2;  R = Vnet./I; Tinv = 1./T0 + 1./B .*
log(R ./ R0);CNV = (1./Tinv)-273;';
  PARAMS.(FNAME)(SET).cnv(C).cnvcode   = [105 202];
  PARAMS.(FNAME)(SET).cnv(C).series    = 1;
  C = C + 1;
  PARAMS.(FNAME)(SET).cnv(C).dscs      = 'Shutter block temp';
  PARAMS.(FNAME)(SET).cnv(C).units     = 'C';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow   = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode   = [106 200];
  PARAMS.(FNAME)(SET).cnv(C).auxcode2  = [110 200];
  PARAMS.(FNAME)(SET).cnv(C).estr      = 'Vnet = AUX - AUX2;  R = Vnet./I; Tinv = 1./T0 + 1./B .*
log(R ./ R0);CNV = (1./Tinv)-273;';
  PARAMS.(FNAME)(SET).cnv(C).cnvcode   = [106 202];
  PARAMS.(FNAME)(SET).cnv(C).series    = 1;
  C = C + 1;

  PARAMS.(FNAME)(SET).cnv(C).dscs      = 'Case air temp';
  PARAMS.(FNAME)(SET).cnv(C).units     =  'C';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow   = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode   = [107 200];
  PARAMS.(FNAME)(SET).cnv(C).estr      = 'R = 10000 ./ (scaleV ./AUX);Tinv = 1./T0 + 1./B .*
log(R ./ R0);CNV = (1./Tinv)-273;';
  PARAMS.(FNAME)(SET).cnv(C).cnvcode   = [107 202];

  C = C + 1;
  PARAMS.(FNAME)(SET).cnv(C).dscs      = 'Case air humidity';
  PARAMS.(FNAME)(SET).cnv(C).units     =  '%';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow   = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode   = [108 200];
  PARAMS.(FNAME)(SET).cnv(C).estr      = 'CNV = 100.*(AUX./scaleV);';
  PARAMS.(FNAME)(SET).cnv(C).cnvcode   = [108 202];

  C = C + 1;
  PARAMS.(FNAME)(SET).cnv(C).dscs      = 'Case moisture sensor 0%=dry';
  PARAMS.(FNAME)(SET).cnv(C).units     = '%';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow   = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode   = [109 200];
  PARAMS.(FNAME)(SET).cnv(C).estr      = 'CNV = 100 - (AUX./scaleV) .*100;';
  PARAMS.(FNAME)(SET).cnv(C).cnvcode   = [109 202];
  C = C + 1;

  PARAMS.(FNAME)(SET).cnv(C).dscs      = 'Spectrograph internal temp 1';
  PARAMS.(FNAME)(SET).cnv(C).units     = 'C';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow   = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode   = [110 200];
  PARAMS.(FNAME)(SET).cnv(C).auxcode2  = [111 200];
  PARAMS.(FNAME)(SET).cnv(C).estr      = 'Vnet = AUX - AUX2;  R = Vnet./I; Tinv = 1./T0 + 1./B .*
log(R ./ R0);CNV = (1./Tinv)-273;INTTMP1 = CNV; ';
  PARAMS.(FNAME)(SET).cnv(C).series    = 1;
  PARAMS.(FNAME)(SET).cnv(C).cnvcode   = [110 202];
  C = C + 1;

  PARAMS.(FNAME)(SET).cnv(C).dscs      = 'Spectrograph internal temp 2';
  PARAMS.(FNAME)(SET).cnv(C).units     = 'C';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow   = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode   = [111 200];
  PARAMS.(FNAME)(SET).cnv(C).estr      = 'Vnet = AUX;  R = Vnet./I; Tinv = 1./T0 + 1./B .* log(R
./ R0);CNV = (1./Tinv)-273;INTTMP2 = CNV;';
  PARAMS.(FNAME)(SET).cnv(C).cnvcode   = [111 202];
  PARAMS.(FNAME)(SET).cnv(C).series    = 1;

  C = C + 1;
  PARAMS.(FNAME)(SET).cnv(C).dscs      = 'Spectrograph internal humidity';
  PARAMS.(FNAME)(SET).cnv(C).units     = '%';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow   = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode   = [112 200];
  PARAMS.(FNAME)(SET).cnv(C).estr      = 'sensRH = -0.1515 + AUX ./ ( scaleV .* 0.00636); specT =
transpose(nanmean(transpose([INTTMP1 INTTMP2]))); CNV = sensRH ./ (1.0546 - 0.00216 .* specT);';
  PARAMS.(FNAME)(SET).cnv(C).cnvcode   = [112 202];

  C = C + 1;
  PARAMS.(FNAME)(SET).cnv(C).dscs      = '+5V supply and full scale ref.';
  PARAMS.(FNAME)(SET).cnv(C).units     = 'V';
  PARAMS.(FNAME)(SET).cnv(C).vauxrow   = [34 35];
  PARAMS.(FNAME)(SET).cnv(C).auxcode   = [113 200];
  PARAMS.(FNAME)(SET).cnv(C).estr      = 'CNV = AUX;';
```

```
PARAMS.(FNAME)(SET).cnv(C).cnvcode    = [113 202];
C = C + 1;
```

The cnv sections will be used to convert the aux data. Note 2 things. One the global has to be run first. And the rest of the cnv's need to be run in order, so that variables created exist for the next sensor that needs it.

I should probably add a range to each sensor. I would have to know what that range is. This range would be for marking data bad. But there is another possibility is an alarm value. If this sensor goes above or below the set value an alarm is emailed or something.

Aux data Vaux set up

As of 18 May 2020 in mr_vauxtyp_.m has the aux data set up as

```
% Aux Data ---------------------------------------------------------------------------
  Vaux(400,VR) = Aux data Start Time                    Matlab Date                 |
  Vaux(401,VR) = Aux data Stop Time                     Matlab Date                 |
  Vaux(402,VR) = AuxVrawNAvg                            Number of samples averaged  |
  Vaux(403,VR) = PrcFlag                                0 = raw, 1 = converted      |
  Vaux(404,VR) = CoolTargTempC                          Celcius                     |
  Vaux(405,VR) = CoolCurrentTempC                       Celcius                     |
  Vaux(406,VR) = CoolOnFlag                             Flag                        |
  Vaux(407,VR) = CoolAndorStatusCode                    Flag                        |
  Vaux(408,VR) = CoolTempStableFlag                     Flag                        |
  Vaux(409,VR) = Number of Channels                     N                           |
  Vaux(410:419,VR) = Not assigned .............................................
  Vaux(420,VR) = C100: Auxiliary mains                   Voltage                     |
  Vaux(421,VR) = C101: Auxiliary mains current           Amps                        |
  Vaux(422,VR) = C102: nothing                           Celcius                     |
  Vaux(423,VR) = C103: External heat sink temp           Celcius                     |
  Vaux(424,VR) = C104: CCD Base temp                     Celcius                     |
  Vaux(425,VR) = C105: Falcon heat sink temp             Celcius                     |
  Vaux(426,VR) = C106: Shutter block temp                Celcius                     |
  Vaux(427,VR) = C107: Case air temp                     Celcius                     |
  Vaux(428,VR) = C108: Case air humidity                 Percent                     |
  Vaux(429,VR) = C109: Case moisture sensor                                          |
  Vaux(430,VR) = C110: Spectrograph internal temp 1      Celcius                     |
  Vaux(431,VR) = C111: Spectrograph internal temp 2      Celcius                     |
  Vaux(432,VR) = C112: Spectrograph internal humidity    Percent                     |
  Vaux(433,VR) = C113: +5V supply/full scale reference   Voltage                     |
  Vaux(434:599,VR) = Not assigned .............................................
```

New sensors are added below 433, so code 100 will always be row 420 etc. Data flagged as bad sensors in the specs cfg file are NaN in the files Vaux. The std of the data that Art saves in the hdf file will be saved in a collated aux data mat file and not in this processed file.

2020 May 07 aux data in Arts HDF file

These are the "global" aux data info that is the same for an entire hdf file

```
Group '/raw/version0/ancillary'
 Attributes:
 'AuxChannel':  0 1 2 3 4 5 6 7 8 9 10 11 12 13
 'AuxCode':  100 101 102 103 104 105 106 107 108 109 110 111 112 113
 'AuxSerialNum':  1 1 1 1 1 1 1 1 1 1 1 1 1 1
 'AuxDescription000':  'Auxiliary mains voltage'
 'AuxScaledUnits000':  'V'
 'AuxDescription001':  'Auxiliary mains current'
 'AuxScaledUnits001':  'A'
 'AuxDescription002':  'Internal humidity temp'
 'AuxScaledUnits002':  'C'
 'AuxDescription003':  'External heat sink temp'
 'AuxScaledUnits003':  'C'
 'AuxDescription004':  'CCD Base temp'
 'AuxScaledUnits004':  'C'
 'AuxDescription005':  'SBC heat sink temp'
 'AuxScaledUnits005':  'C'
 'AuxDescription006':  'Shutter block temp'
 'AuxScaledUnits006':  'C'
 'AuxDescription007':  'Case air temp'
 'AuxScaledUnits007':  'C'
 'AuxDescription008':  'Case air humidity'
 'AuxScaledUnits008':  '%'
 'AuxDescription009':  'Case moisture sensor 0%=dry'
```

```
'AuxScaledUnits009':  '%'
'AuxDescription010':  'Spectrograph internal temp 1'
'AuxScaledUnits010':  'C'
'AuxDescription011':  'Spectrograph internal temp 2'
'AuxScaledUnits011':  'C'
'AuxDescription012':  'Spectrograph internal humidity'
'AuxScaledUnits012':  '%'
'AuxDescription013':  '+5V supply and full scale ref.'
'AuxScaledUnits013':  'V'
'Nancillary':  25
```

These are the data that are saved in the file. Art starts saving the aux data imediatly (I.E. before the camera's temperature is stable. This allows us to see how the temps are changing before data are collected. You can see this below because the `'CoolTempStableFlag'` is 0 which means the camera's temperature is not stable.

```
Group '/raw/version0/ancillary/ancillary001'
 Attributes:
 'MeasurementTime':  '20200507_033632_GMT'
 'AuxVrawNAvg':  10
 'AuxVrawMean':  2.592800 0.009600 2.605900 0.625200 0.510700 0.396100 0.286100
      2.691500 2.675300 5.025100 0.198300 0.100100 2.711500 5.025700
 'AuxVrawStdev':  0.001300 0.000000 0.013400 0.000600 0.000800 0.000500 0.000400
      0.014400 0.001500 0.029800 0.000200 0.000100 0.016400 0.032300
 'AuxScaled':  12.964200 -14.941800 25.104300 24.978500 24.978400 24.984900 25.020600
      25.099100 53.231700 0.012700 25.002800 24.999800 84.628500 5.025700
 'GoesWithImage':  'N/A'
 'GoesWithImageData':  0 0
 'CoolOnFlag':  1
 'CoolTargTempC':  -60
 'CoolCurrentTempC':  20
 'CoolAndorStatusCode':  20037
 'CoolTempStableFlag':  0

Group '/raw/version0/ancillary/ancillary013'
 Attributes:
 'MeasurementTime':  '20200507_034243_GMT'
 'AuxVrawNAvg':  10
 'AuxVrawMean':  2.615200 0.009700 2.605000 0.625500 0.509900 0.404900 0.294100
      2.668900 2.721100 5.020500 0.204500 0.103000 2.715300 5.027100
 'AuxVrawStdev':  0.000200 0.000000 0.017200 0.000600 0.000800 0.000500 0.000500
      0.016900 0.002000 0.029300 0.000300 0.000100 0.016900 0.032100
 'AuxScaled':  13.076300 -14.941700 25.104400 24.977000 24.992300 24.983600 25.017400
      25.100500 54.129500 0.130800 24.997600 24.995200 84.723000 5.027100
 'GoesWithImage':  'N/A'
 'GoesWithImageData':  0 0
 'CoolOnFlag':  1
 'CoolTargTempC':  -60
 'CoolCurrentTempC':  -60
 'CoolAndorStatusCode':  20036
 'CoolTempStableFlag':  1
```

Once the temperature is stable (flag = 1) Art will start taking data. If there is an image/track associated with this aux data you will see the `'GoesWithImageData'` have two numbers none zero numbers in it. The first number is the type if image (dark = 1, background = 4, light = 2) and the second number is the dataset number. So if the dataset is dark001 then the second number would be 001.

```
'GoesWithImage':  '/raw/version0/images/dark001'
'GoesWithImageData':  1 1
'CoolTempStableFlag':  1
```

- Each /raw/version0/ancillary/ancillaryX group has the following attributes:
MeasurementTime: time stamp
AuxVrawNAvg: Number of times the labjack was read to generate average voltages
AuxVrawMean: Mean of AuxVrawNAvg measurements on each labjack input channel
AuxVrawStdev: Standard Deviation of AuxVrawNAvg measurements on each labjack input channel
AuxScaled: AuxVrawMean converted to engineering units (V, A, C, percent etc..)
GoesWithImage: The name of the image dataset corresponding to this ancillary measurement
or 'N/A' if no image taken at the same time (e.g. during cool down)
CoolOnFlag: 1 = cooler is ON, 0 = cooler is OFF
CoolTargTempC: target temperature for cooler (C)
CoolCurrentTempC current temperature for cooler (C)
CoolAndorStatusCode: a code that the camera returns when you poll the temperature.
20037 = temp not stable, 20036 = temp is stable, I can send you
the other values but basically any other number indicates an error.
CoolTempStableFlag: 1 = cooler is stable (code 20036), 0 = cooler is not stable (Code 20037)

- For the ancillary measurements that correspond to images, MeasurementTime will be equal to the end time of the image. So you could align based on

/raw/version0/ancillary/ancillaryX/MeasurementTime
and
/raw/version0/images/*/GMT_End

or align based on /raw/version0/ancillary/ancillaryX/GoesWithImageData
whichever works better for you.

History

Email from Mark to Art on converting the labjack measurements

You will need the following info to make sense of the second labjack readings

A0- Mains volts, 12 volts nominal, Use 5v scale
The input voltage is divided by 5 to account for the 10v max input

A1-Mains current
± 15 amps FS, Use 5v scale, (FS proportional to A13 A+5v) mid range is zero current.

A2- unused

A3- Ext heat sink Temperature
See note below, use 1v scale

A4 - CCD Base Temperature
See note below, use 1v scale

A5 - Falcon heat sink Temperature
See note below, use 1v scale

A6 - Shutter block Temperature
See note below, use 1v scale

A7 - Case Air Temperature
Scaled the same as the Controller T/H probe, use 5v scale (FS proportional to A13 A+5v)

A8 - Case Humidity
Scaled the same as the Controller T/H probe, use 5v scale (FS proportional to A13 A+5v)

A9 - Moisture Sensor
Resistance reading using the 200uA excitation.
When dry it will read a maximum voltage based on the 200uA excitation compliance voltage
When wet the resistance will approach 100k ohms.

A10 - Spectrograph internal Temperature #1
See note below, use 1v scale

A11 - Spectrograph internal Temperature #2
See note below, use 1v scale

A12 - Spectrograph internal Humidity
See data sheet attached, use 5v scale, reading FS is proportional to A+5v

A13 - A+5 v supply and FS reference for channels as noted above, use 5v scale

Note on reading temperatures:
Temperature channels AN3, AN4, AN5, AN6, AN10, AN11 have the thermistors wired in series and excited by the 10uA source.
So the voltage readings for each thermistor must be obtained by subtraction.
AN3 volts = AN3-AN4
AN4 VOLTS = AN4-AN5
AN5 VOLTS = AN5-AN6
AN6 VOLTS = AN6-AN10
AN10 VOLTS = AN10-AN11
AN11 VOLTS is read directly

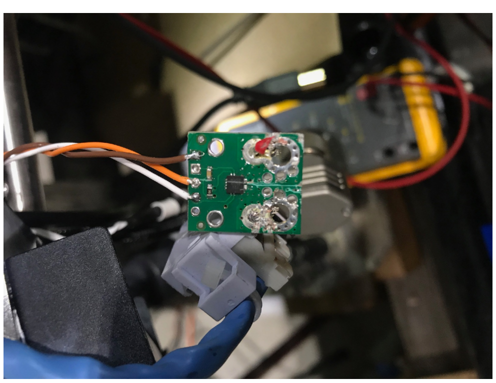
The 1 volt range should work well enough for all these channels but An10 and 11 may benefit by the higher resolution of a higher gain scale.

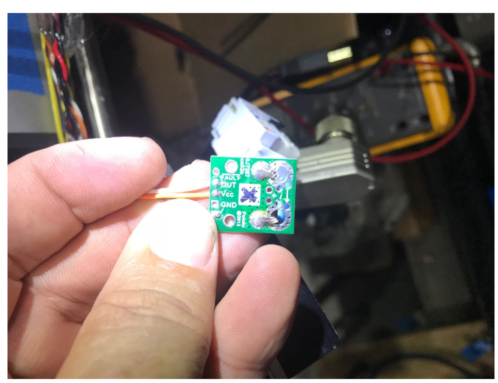
For these channels. resistance is determined by VOLTS / AMPS = RESISTANCE (originally this was VOLTS x AMPS which was wrong)
Temperature is calculated using this resistance as with all our other thermistors.

There is more info on the series resistance measurement in the online Labjack app notes.

U6-PRO Datasheet PDF
https://labjack.com/sites/default/files/LabJack-U6-Datasheet-Export-20161024.pdf

Email from Mike on 5/12/2020, 8:00 PM
The small current sensor board, which fired out on both the BS and RS during M267 deployment - presumably during the JPJ flood / power-short anomaly - is probably "Auxiliary mains current", and hence is bogus for both BS & RS now. Here are photos of said board after the M267 deployment:





If you add these photos to your aux doc, iWill try to update them when the sensor is replaced.

Email from Mike on 5/14/2020, 11:20 PM

Repair work on the RS04 began today.
1.) The bad main 12 Volt supply was replaced.
2.) 2x shutter control boards were replaced - old/bad= SN 003, new/OK= SN 001.
3.) Bad DC power supply on the Auxiliary board was replaced.

The External Water Temperature thermistor was not wired up - for both BS & RS. This was because: 1.) the 14x Aux channels were all used, 2.) Mark worried aboot possibly breaking a connector if the thermistor got hung up - i.e. it stuck outside of the clam shell housing. Maybe we can build a cage to protect the cooling fins and the H2O thermistor?

However, there is an External Water Temperature thermistor on the OM, but maybe Ken needs to program it's data acquisition ?

Also, there could be an External Water Temperature thermistor installed on the Splitter housing - even though this data, via the Upper Controller acquisition probably will not coincide with BS,RS data acq. (?)

Oh yes, the official name is now UC = Upper Controller. UC01 was on MOBY268 and will be on MOBY270.

Attached here is the latest / updated version of the Aux wiring diagram <attached: 20200515_BS,RS_Aux-wiring-diagram_1, 2.jpg>
...2 versions to get the top & bottom edges scanned. All the hand-written notes are not yet updated on the digital version...

From this diagram "one can see that" the third internal Resonon thermistor is not connected. On the Resonon Hummer connector #1 is "Slit therm" and "Near Camera therm", which wire to Aux #10 & 11 (counting from zero) I.E. see diagram notes in blue-ink. The Resonon Hummer #2 has the "Internal Humidity sensor" at Aux#12, and the third Internal therm = "Under Cap" is not connected.

Mark thinks maybe Labjack has a multiplexor that can be added to get
more aux channels (?).

Question for Art:
Are the aux thermistors being read with full (24bit ?) resolution ?
I.E. the ./readancin (in /home/user/src/labjack05) printed only 0.001 V resolution,
for the stream of 6x thermistors @ Aux# 3,4,5,6,10,11 (counting from zero),
which did not seem to be enough digits to see any converted temperature changes
from a heated thermistor. Mark would like to see what the temperatures look like
at two settings: 1.) gain 1x (10V range) at 21bit resolution, and 2.) gain 10x (1V range)

at 18bit resolution. Just for the 6x thermistors in the string. "Ideal would be if #2 works because sample time is faster." Mark's last text: "I'm not sure if we can do this without the Art using the Labjack provided test program bit I doubt it."

Data for Steph / Art,
After we checked that all shutters open/closed OK, we took a 1Dk/2Lt/1Dk test scan set with shutter #14 (counting from one) open, via asg_interactive, ambient tent light on bare shutters. The .fits and .h5 are at the NAS /ftp1/Mike/HI-2020-03/RS04/day01/
Also there are putty .log files - #4 shows the ./readancin output after the
DC supply on the Aux board was replaced.

One last item for Art's to-do list:
Right now there is a 10sec delay before the shutter open/close pulse is sent -
this can be shortened to 2sec wait time (and possibly less than 2sec if 2sec works OK).

MF

Email from Mike on 5/15/2020, 11:05 PM

Hi Steph,
I confirmed with Mark & the mysterious Aux wiring diagram, that the Aux board's DC/DC supply is the source of the 5 Vdc reference, and it powers the (missing) mains current sensor, and the 2x humidity sensors - one inside the Resonon spec, and one inside the housing - and the whet sensor. So, all those would have had bogus readings yesterday during the day01 test data set. However, the 6x thermistors are powered via the Labjack board, so that is why they were OK.

There will be no "Mains Current" sensor installed, until Mark understands why they went boom.

Today, Mark re-assembled the RS innards. I took another test scan set, which is at the NAS under HI-2020-03/RS04/day02, along with the putty log. The log shows the OK results of Aux scans via ./readancin. The CCD cooler was OFF, and all 14x shutters were open for the 1Dk/2Lt/1Dk test.

I think that on Monday it is OK for Sean to re-assemble the RS with the Splitter. Which I believe means installing at least 1/2 of the plastic housing (?). Mark did remind that the O-rings at the shutters do not need O-ring grease - they only seal light leaks.

Then we can re-setup for more Art testing in the tent.

I think the Ti housings will be installed before H2O bath work - this requires lengthening the internal rods that all the innards assembled onto.

The 9-pin Enet conn on the RS04 is bad / stripped & needs replaced.

That should be aboot all for now...
Happy Friday! MF

Email from Mike on 5/16/2020, 8:48 PM

Thanks Steph!
It looks like the mains & 5V-ref Volts are good, and the internal humidity & case moisture are good - which is all good. The mains current remains bad which is expected.

I am wondering aboot the conversion eqn you're using for the "Spectrograph Internal Humidity" - is it the same eqn as for the "Internal Humidity Sensor" ? If same, then I think that's incorrect, and we need to dig up the correct conversion for the Spec %RH. I.E. the Spec %RH looks too high @ 81.

Also... it is interesting to me to see the scan timing, via the vertical dashed lines in the Aux plots! I.E. the longer delay between Dk#1 and Lt#1 when the shutters are opened, and the shorter delay between Lt#2 and Dk#2 when the shutters are closed, and the shortest delay between Lt#1 and Lt#2 with no shutter change. I am thinking aboot the telecon discussion r.e. spec vs laser timing...

And, light on all tracks is also a good sign. With the CCD cooler off I guess there are noisy pixels in the dark scans...

I think it is OK to go ahead with RS re-assembly on Monday.
MF

Email from Mike on 5/18/2020, 11:09 PM

Hi Steph,
iThink this is close!

iThink yes, the 6x thermistors powered by the LabJack are the AN3, AN4, AN5, AN6, AN10, AN11 in the diagram, which can be seen just to the left of the U3 connector near center, and their power is the from the long U12 connector at diagram left (looks like pin#3 labeled "10ua") where the U12 is the Labjack interface.

On wiring diagram, the box at upper right says
A3=Ext. Heatsink base temperature, which agrees with your below code#103 chan#3
A4=CCD base temperature == code#104, chan#4
A5=Falcon temperature == code#105, chan#5 == your "SBC heat sink temp"
A6=Shutter block temp == code#106, chan#6
A10=Spec internal temp 1 == code#110, chan#10
... iThink this is "Slit thermistor" in blue ink on the wiring diagram at top
A11=Spec internal temp 2 == code#111, chan#11
... iThink this is "Near camera thermistor" in blue ink on the wiring diagram at top

A12=Spec Internal Humidity, powered by DC/DC (not Labjack)

A13=hand-written "Analog Pwr", which monitors the DC/DC power supply.
iSuppose you could say is "Powered By" the DC/DC.

On the diagram it looks like A7 & A8 are the "Case air temp" & "Case humidity",
and those are powered via the DC/DC (not Labjack).
And A9 = "Case moisture", also DC/DC supply.

On the diagram AN0 goes to the scribbled-out "DIV 5" with is a 5x-Voltage-divider - maybe Mark thought to make it a 2x-divider? (see hand-scribble near upper left) but iThink it is still a 5x-divider. You can see this in the photo iSent
"20200514-12_RS4-repair-Aux-DCDC-supply-sm.jpg"
the little board to the left of the silver DC/DC supply, withone red wire coming out, screwed @ pins #27:30 - it says "LJTick-Divider-5" on it !

On the diagram AN1=Mains current, but iAm confused aboot it's power source. The sensor board is at the right side of the diagram, in the yaxis section labeled 4, the square box hand labeled "Cmp I". It looks like it's powered by "SPEC PWR" which is probably the main power supply that blew up.

So the only remaining mystery is AN2 !
Wiring diagram upper-right box calls it "Regulated D+5 Power" -
but it is crossed out in that upper-right box with "ANDOR" hand-written !
And you call it "Internal humidity temp".

The wire on the diagram @ U3 U4 connector pin#24 goes to "+5V ANDOR",
which is different than the "SPEC +PWR" for the AN1 current sensor.
I am confused aboot the diff between ANDOR and SPEC power?
It doesn't look like any kine temperature though...

MF

Email from Mike on 5/21/2020, 7:09 PM

Ooooh, that's an EASY one - yay!
SBC, Single Board Computer == Falcon, in the BS & RS (and SBC == Rabbit in the OM).
But, SBC is an un-truth, because there is more than one board. Just the serial and USB interfaces take up one board,and iThink there is a separate board for the disc drive (?).

Looking at Mark's wiring diagram, the upper-right name-box says "A5- Falcon temperature", but below this box, the third one down is "FALCON HEATSINK", which connects to the U3,U4 connector at pins 40,40, called AN5 to the left of U3.

And, in your .pdf aux doc, pg.16 top figure for "Channel 005" shows the thermistor in the heat sink (the silver curved-top piece) which is bolted to the SBC/Falcon boards.

(See following email this turns out to be wrong which Mike confirmed in next email)
However...
iAlso have the attached photo from RS04 repairs on 14-May-2020
<attached: 20200514-01_RS4-repair.JPG>
which shows the the SBC/Flacon thermistor connected at the
bottom of the serial/USB board - where the therm is at the end of
the two twisted blue wires with the white shrink tube at the end,
screwed down just below that 9pin male "D" serial port -
...so, maybe Mark changed the placement of this therm?
I.E. moved it from the heat sink to the computer board(s)?

I'll go have a look to see if iCan confirm/deny this...
And iWill have a look at your new pages.
MF

Email from Mike on 5/21/2020, 8:20 PM

Good. iShall look at your .pdf!

First,
Cancel what iSaid below r.e. the 14-May photo.
iTook that photo bcs Mark tole me it was the Falcon therm,
but iMust have mis-understood. Looking at the RS04 just now,
the 14-May photo is of the External Heat Sink themistor -

it just looks like it's at the serial/USB board from this angle.
So that 14-May photo for the RS04 is the same therm as in
your (old) aux .pdf doc pg.12 top photo = Channel 003, for the BS03.
On the RS04, this therm is on the other-side of the Ext heat sink -
so RS position is off by 180 degrees from the BS position.

There is still a therm in the SBC heat sink on the RS04,
same as your pg.13 Channel 005.

Sorry, MF

<u>Email from Mike on 5/21/2020, 9:43 PM</u>

iLike photos too. iThink they are great documentators.
Speaking of...
iStarted digging around for M270 FO head parts and came across a Falcon SBC.
<attached: 20200521_Flacon-SBC_XX.JPG>

It looks there are three boards involved.
The second board is not a disc drive -
near the middle of photo #04 is a memory chip, which has a hand-written
serial number, so iBet this is where Art's operating system resides.
In photos #1:5 the silver plate @ bottom is the heat sink,
and the two copper tubes sticking out of the heat sink are the heat pipes
that connect said heat sink to the external cooling fins.

The #06 photo shows serial/USB board at top, and
there is a Enet RJ45 connector attached to the SBC, and
the big black connector at bottom connect the SBC & USB boards,
and the connector at bottom right is for power.

In the #07 photo the Ligitech chip in one of the 4x USB ports
is a wireless keyboard/mouse receiver.
MF

OLD Converting the raw aux data (May 2020)

Using Art's ancillaryscaleV.c program to convert the raw aux dat to converted (May 2020)

C Var	Chan	chancode	Channel Name
AMAINV	0	100	Auxiliary mains V
AMAINI	1	101	Auxiliary mains current
AINTRHT	2	102	Internal humidity temp
AEXTT	3	103	External heat sink temp
ACCDT	4	104	CCD Base temp
AFALCONT	5	105	Falcon heat sink temp or SBC heat sink temp
ASHUTT	6	106	Shutter block temp
ACASET	7	107	Case air temp
ACASERH	8	108	Case air humidity
AMOISTSEN	9	109	Case moisture sensor
ASGT1	10	110	Spectrograph internal temp 1
ASGT2	11	111	Spectrograph internal temp 2
ASGRH	12	112	Spectrograph internal humidity
AREFV	13	113	+5V supply and full scale reference

```
T0 = 25;   %/* C, room temperature */
B  = 3950;
R0 = 10000;
m  = 30.0 ./ 5.0;
b  = -15.0;
I  = 10E-6; %/* 10 uA source */
scaleV = AREFV;
```

Convert Auxiliary mains V

```
AMAINVcnv = AMAINV*5
```

Convert Auxiliary mains current

```
m = 30.0 ./ 5.0;
b = -15.0;
AMAINIcnv = AMAINI.*m + b
```

Converting the temperatures from the resistor series

My vars	Arts vars	Long names	channel num
RAW(:,1)	AEXTT	External heat sink temp	3
RAW(:,2)	ACCDT	CCD Base temp	4
RAW(:,3)	AFALCONT	SBC heat sink temp	5
RAW(:,4)	ASHUTT	Shutter block temp	6
RAW(:,5)	ASGT1	Spectrograph internal temp 1	10
RAW(:,6)	ASGT2	Spectrograph internal temp 2	11

```
% AN3  volts = AN3-AN4
% AN4  VOLTS = AN4-AN5
% AN5  VOLTS = AN5-AN6
% AN6  VOLTS = AN6-AN10
% AN10 VOLTS = AN10-AN11
% AN11 VOLTSIs read directly
```

First subtract to get the correct readout

```
for k = 1:5 % We don't subtract anything from RAW(:,6)
  Vnet(:,k) = RAW(:,k)-RAW(:,k+1);
end
```

Then convert each sensor

```
for k = 1:6
  R = Vnet(:,k) ./ I;
```

```
  Tinv = 1./T0 + 1./B .* log(R ./ R0);
  CNV(:,k) = (1./Tinv);
end
```

Next Channel 2 - Internal humidity temp

```
R = 10000 ./ (scaleV ./ AINTRHT);
Tinv = 1./T0 + 1./B .* log(R ./ R0);
AINTRHTcnv = (1./Tinv);
```

Channel 7 - Case air temp

```
R = 10000 ./ (scaleV ./ACASET);
Tinv = 1./T0 + 1./B .* log(R ./ R0);
ACASETcnv = (1./Tinv);
```

Channel 8 - case air humidity

```
ACASERHcnv = 100.*(ACASERH./scaleV);
```

Channel 9 - case moisture sensor HI=DRY=0% wet LO=WET=100% wet

```
AMOISTSENcnv = 100 - (AMOISTSEN./scaleV) .*100;
```

Channel 12 - spectrograph internal humidity

```
sensRH   = -0.1515 + ASGRH./ ( scaleV .* 0.00636);
specT    = nanmean(CNV(:,5:6)')';
ASGRHcnv = sensRH ./ (1.0546 - 0.00216 .* specT);;
```

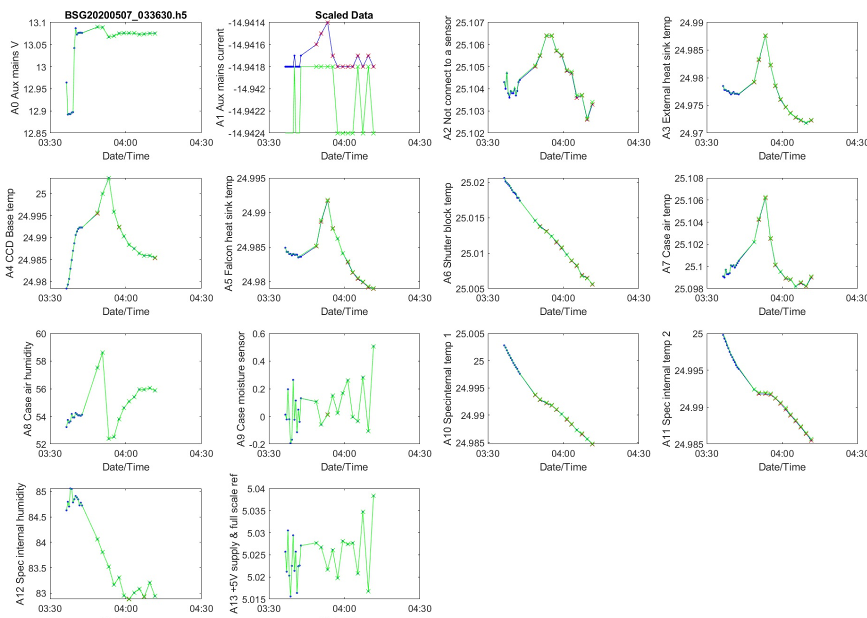


Figure 16. Converted Aux data from Raw data using Art conversion program

You can see I am still having issues with Mains Current. Not sure why??

Another unrelated issue is why are the ranges of many of the temperatures so tiny. Mark said they would check to labjack and make sure the volts are coming out right. - fixed Jun 2020

The above graph is from the “in-water” hdf file Art has been practicing on. When I convert using the AUX*.h5 files the differences are 1e-14.

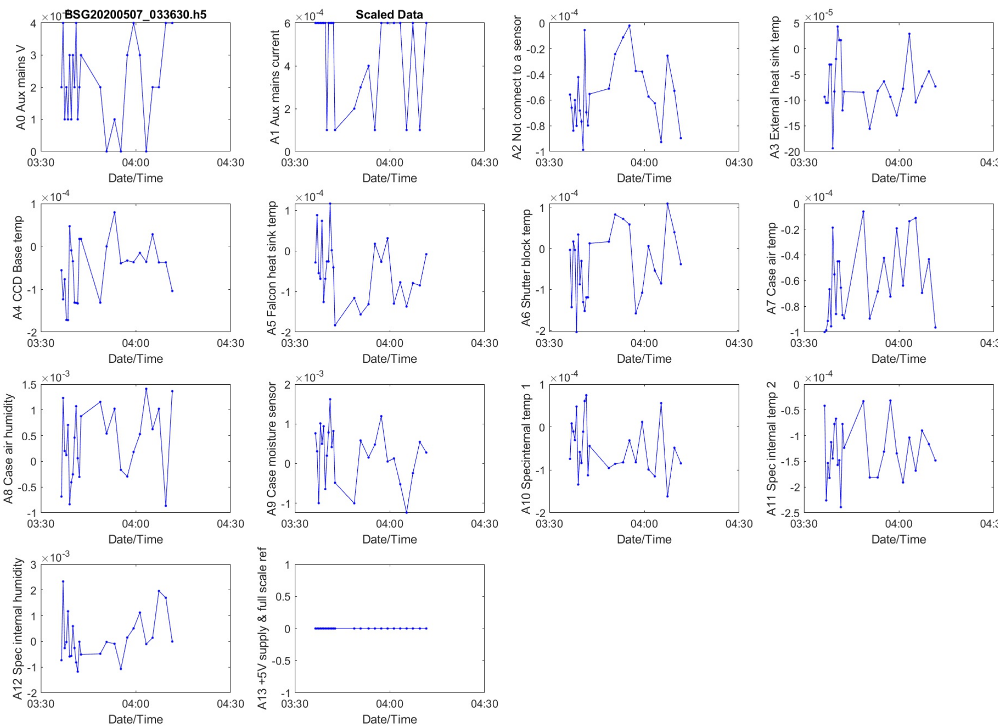
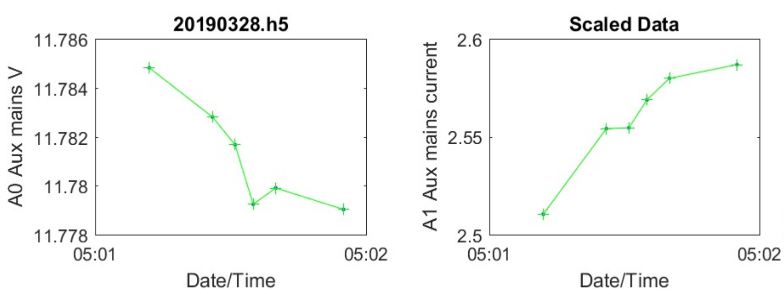


Figure 17. The difference between my converted aux data and Arts converted data

Dir: D:\zflora\mldata2\mobyrefresh\testing\hdf\20190328_auxdata\
file: 20190328_003_01_cropped.png



For a test aux file art sent in Mar 2019 the mains current comparison of my and Art converted data are exactly the same. Not sure what the changes is.

When I so the same on a h5 file with image data and aux data the conversion works perfectly. So this seems like it is really just the experimental AUX data mike takes using Art programs.

<u>Wiring Diagram from Mike Feinholz</u> (2015)

Original Resonon documentation, ca. Mar-2015 (below) which shows: "Slit Thermistor", "Near-camera Thermistor", and "Under-cap Thermistor",and notes, the "Under-cap Thermistor" is next to the humidity sensor (Case Air Temperature???).

Moby Blue wiring chart

Connector 1	
Pin 1	Slit Thermistor
Pin 2	Slit Thermistor
Pin 3	Near-camera Thermistor
Pin 4	Near-camera Thermistor
Pin 5	Unused

Connector 2	
Pin 1	Humidity Sensor –ve (0 V)
Pin 2	Humidity Sensor Signal Out*
Pin 3	Humidity Sensor +ve (+3.3 V)
Pin 4	Under-cap Thermistor
Pin 5	Under-cap Thermistor

*No load has been bridged to ground as suggested in the Honeywell Typical Application Circuit.

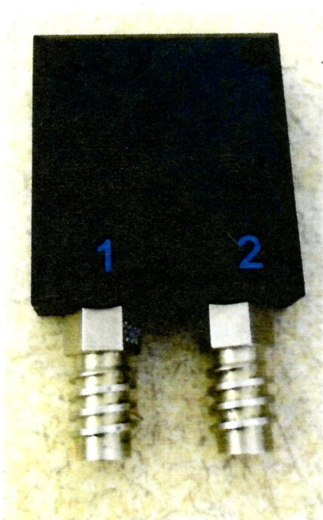


Figure 1: Connectors 1 and 2

26-Mar-2015, MF

email from Casey Dodge:
thermistors = US Sensors PT103J2
± 0.2°C (0-70°C), 10kΩ @ 25°C,
R-T curve = J, Beta (K) 0-50°C = 3890

email from Casey Smith:
Humidity sensor is: PN HIH-5030
sensing.honeywell.com
Voltage output, 2.7 to 5.5 Vdc input, ± 3 %RH
Response time: 5 sec, 1/e in slow moving air
Stability: 0.25 %RH / year, 1.2 % RH / 5 yr
Operating range: −40 to 85°C, 0 to 100 % RH
Current draw: 200 μA

31-Mar-2015: "Under-cap Thermistor" is directly under the cap next to the humidity sensor. The 5-pin conn/cable is from SeaCon (hummer series) HUMG-5-BCR bulkhead conn
CCP cable conn

The thermistors, for temperature sensing, are all two-wire sensors. A thermistor is a resistor, whose resistance, Ohm, changes with temperature. But the Aux board can only measure Voltage, so the thermistor Ohms need converted to Voltage, by way of V = I*R. There is a known current I (Amps) run thru the thermistor resistance R (Ohm) to get output Voltage V (Vdc). The "known current" itself comes from an input V thru a know R, I = V/R, where the input V is from the Labjack Aux board itself (or maybe from Mark's DC power supply?).

On connector 1, pin 1&2 are for one thermistor, pin 3&4 are for another.
On connector 2, pin 4&5 are for the other thermistor.

The Humidity sensor on connector 2 is a different beast. It needs a Voltage input and provides a Voltage output.

Trying to match up wiring diagram to channel numbers

I'm not 100 percent sure which connector and pin combinations correspond with which channel. Also not sure about how channel names are mapped to the wiring names below. Mike has some guesses. These I think are the sensors installed at Resonon and are for the most part internal to the camera and there are no photos of them??

Connector/pin	Name	Ch#	HDF sensor name guess
1/1	Slit thermistor	10	Spectrograph internal temp 1
1/2	Slit thermistor	10	Spectrograph internal temp 1
1/3	Near-Camera Thermistor	11	Spectrograph internal temp 2
1/4	Near-Camera Thermistor	11	Spectrograph internal temp 2
1/5	Unused		
2/1	Humidity sensor ~ve(0 V)	12	Spectrograph internal humidity
2/2	Humidity sensor Signal Out*	12	Spectrograph internal humidity
2/3	Humidity sensor +ve (+3.3 V)	12	Spectrograph internal humidity
2/4	Under-Cap Thermistor		Maybe Case air temp ?
2/5	Under-Cap Thermistor		Maybe Case air temp ?

<u>Sensor specs and conversion equations</u>

Channel 000: Auxiliary mains voltage (V)

Multiply the value by 5 to get original voltage values - this matches Arts scaled data

Channel 001: Auxiliary mains current (A)

Use a linear regression (P = [6 -15]) to convert to amps - this matches Arts scaled data

---

I have spend about a week or so trying to convert some of the below sensors to degrees C to no avail. I have not even gotten to trying the humidity sensors. I just do not have enough details to figure it out. I have looked at Mikes docs that describe the labjack data collected using his Labview program. Which I was able to figure out how to read in and convert. But none of those equations work on this data. I have tried using the VOLTS X AMPS = RESISTANCE equations to no avail. I have look through the Labkack U6-Pro documents. And read everything I can find. I have made my best guess at what the equations should be and then made like 50 variations of them and none name any sense.

Channel 002: Internal humidity temp (C)
May be from the same sensor as Channel 12??
Channel 003: External heat sink temp (C)
Channel 004: CCD Base temp (C)
Channel 005: SBC heat sink temp (C)
Channel 006: Shutter block temp (C)
Channel 007: Case air temp (C)
Channel 008: Case air humidity (%)
Channel 009: Case moisture sensor 0%=dry (%)
Channel 010: Spectrograph internal temp 1 (C) -
US Sensors PT103J2 <attached: PT103J2.pdf, PT103J2 REV NONE (R-T Table).xls>
Channel 011: Spectrograph internal temp 2 ©
US Sensors PT103J2 <attached: PT103J2.pdf, PT103J2 REV NONE (R-T Table).xls>
Channel 012: Spectrograph internal humidity (%)
Internal-spectrograph humidity sensor is Honeywell PN HIH-5030
<attached: HIH-5030_5031_Install_50038351-2-EN_Final_23Feb12.pdf>
<attached: honeywell-sensing-hih5030-5031 series-product-sheet-009050-2-EN.pdf>

Channel 013: +5V supply and full scale ref. (V)

No converting needed already in volts - this matches Arts scaled data

Sensor information using wiring diagram names

Slit Thermistor, Near-camera Thermistor, and Under-cap Thermistor. Under-cap Thermistor is next to the humidity sensor.

Thermistors are from US Sensors, type PT103J2:
10 kOhm @ 25 degC, +/- 0.2 degC (0 to 70 degC)

U.S. Sensor 10k Ohm NTC thermistor, PT103J2, Resistance vs Temperature
http://www.ussensor.com/standard-precision-interchangeable-thermistors-pt103j2-table
PT103J2 REV NONE (RESISTANCE VS. TEMPERATURE TABLE)
Column #1: Temp (°C), Column #2: Resistance (O) 10,000 O @ +25°C
<attached: PT103J2_RT.txt>

Steinhart-Hart equation: 1/T = C1 + C2*(lnR) + C3*(lnR)^3
Temperature (T) is in Kelvin, and resistance (R) is in ohms (0 degC == 273.15 K)
C1=1.12485e-3; C2=2.34793e-4; C3=0.85453e-7;

3rd order regress coef's (...for Labview)
X=natural_logarithm(Ohm)
T=A + B*X + C*X^2 + D*X^3 (degC)
A=508.26; B=-99.7397; C=7.0545; D=-0.20863;

Humidity Sensor is from Honeywell, PN: HIH-5030,
+/- 3%RH, -40 to 85 degC, 0 to 100 %RH, non-condensing environment,
input voltage 2.7 to 5.5 Vdc, current draw 200 uAmp
Typical Application Circuit: 65 kOhm minimum load between Vout and -Ve(0V)

Specifications for Honeywell HIH-5030, Low Voltage Humidity Sensors
http://sensing.honeywell.com/index.php?ci_id=49692
V_out = (V_supply)(0.00636(sensor RH) + 0.1515), typical at 25 degC
True RH = (Sensor RH)/(1.0546 - 0.00216T), T in degC

or, for MATLAB,

sensor_RH = (V_out/V_supply - 0.1515)/0.00636
True_RH = Sensor_RH/(1.0546 - 0.00216*sensor_T)
sensor_T = degC (Chan 3 = Under-cap Thermistor (Ohm) [US Sensors PT103J2])
