BOLO Production User Guide

High Performance Simultaneous Data Acquisition

Revision History

Revision	Date	Author(s)	Description
1.0	29/11/2022	SR	Created
1.1	01/12/2022	SR	Fold in ”quickstart” courtesy of SA
1.2	06/12/2022	SR	Update some images. Expand Production Test logs
1.3	15/05/2023	SR	More details on Current Sense. Adding Data Format section
1.4	25/05/2023	SR,SB,BK	Streamlined calibration script courtesy of SB. LED Test
1.5	11/07/2023	SR	Including DSP block diagram
1.6	06/11/2023	SR	Refreshing Section 4
1.7	10/11/2023	SR	Re-arrange sections so customer facing info appears first

Contents

1 Introduction

For a basic theory of operation and further background information see the BOLODSP Software User's Guide

The Product Specification for the BOLO8 Mezzanine is available here

1.1 Circuit

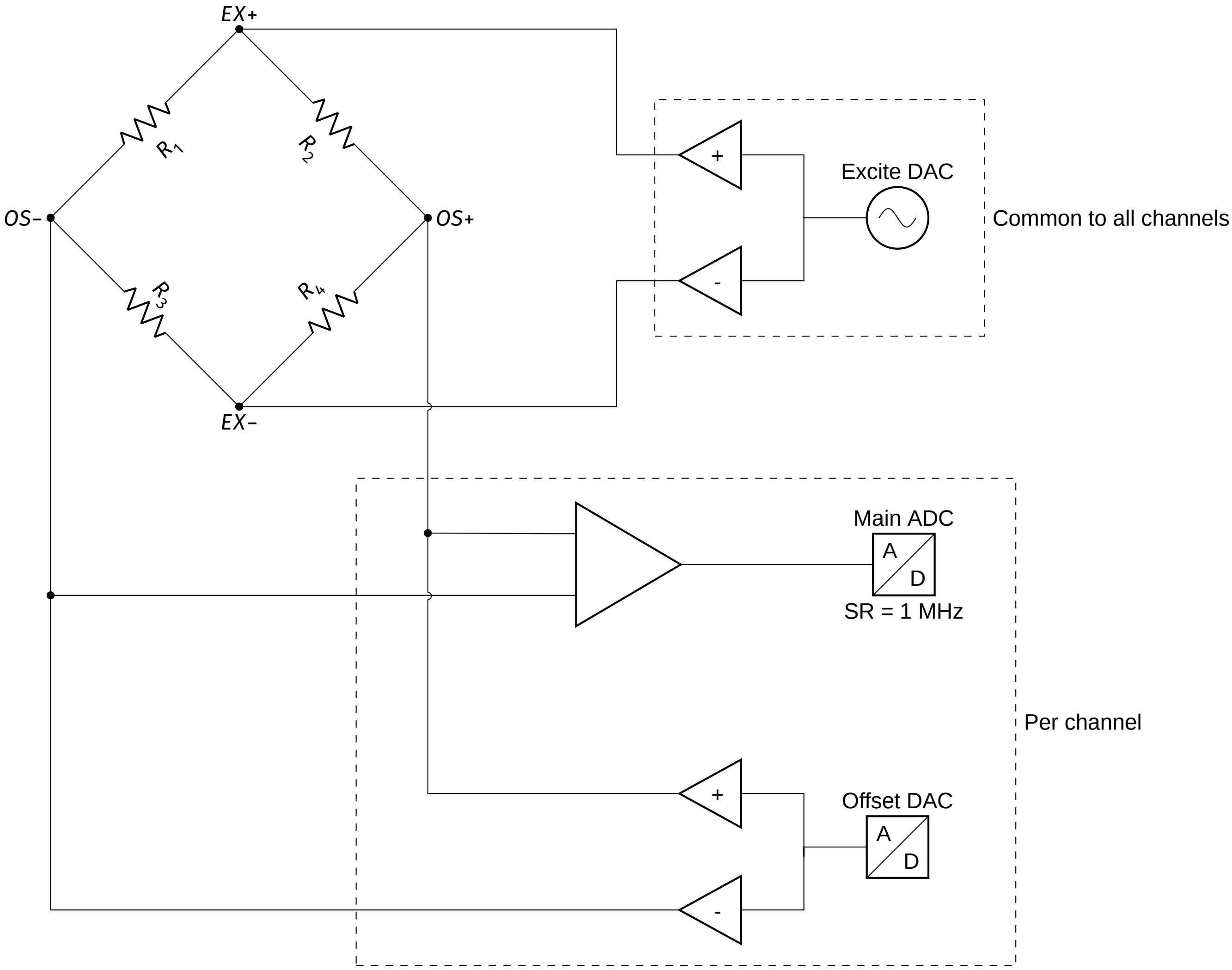


Figure 1: Simplified Bolometer Mezzanine Circuit

1.2 Main ADC Sample Rate and Ranges

Nominal Sample Rate (SR) = 1 MHz
DSP Sample Rate (Decimate by 100) = 10 kHz

- 10 V **This is the default when the bolodsp package is not enabled!**
- 5 V
- 2.5 V
- 1.2 V

1.3 Excite DAC - Frequency and Amplitude

- 18 kHz from wave table in DSP
- Selectable amplitudes. Default for Production Test is 1 V. Default for Calibration is 18 V.
  - "00" = 1 V
  - "01" = 9 V
  - "10" = 15 V
  - "11" = 18 V

1.4 Current Sense Amp and ADC - Sample Rate and Ranges

- Sample rate set in logic to 200 kHz
- The ADC output is then averaged (x20) in the FPGA logic to produce an update rate of 10 kHz
- More averaging (x100) is applied in the DSP Module.
  - These averages are inexact, in that we accumulate over a decimal number, then right-shift by 2^x.
  - We apply scaling factors of $\frac{32}{20}$ and $\frac{128}{100}$ to account for this.

Originally this was a 16-bit ADC but due to component shortages a 12-bit ADC from the same family was subbed in. It was determined that due to the amount of averaging being performed, 12-bit performance could be accepted.

1.5 Offset DAC - Amplitude

Offset DAC maximum amplitude output is ±7.5 V. Anything above 5V (0x5555) will saturate when looped back to the Main ADC.

1.6 Configuration File

The `bolo.sh` file controls some of the basic parameters for the BOLO8 calibration procedure.
Refer to the BOLODSP Software User's Guide for more information.

```
[dt100@eigg-fs acq2106_123]$ cat local/sysconfig/bolo.sh
export BOLO_ACTIVE_CHAN="1 2 3 4"
export BOLO_VERBOSE=1
set.site 14 DIODE_DROP_V 0.5
set.site 14 THEAT 1.0
set.site 14 TCOOL 1.0
set.site 14 VBIAS 1.2
export COPY_CALIB_DATA=1
export CALIBFITARGS="-H 0.6"
export CALIBFIT_EXCITEV=18.0
export LOAD_POWER_FILTERS=1
```

2 Data Format

2.1 Raw Data (Production Test Mode)

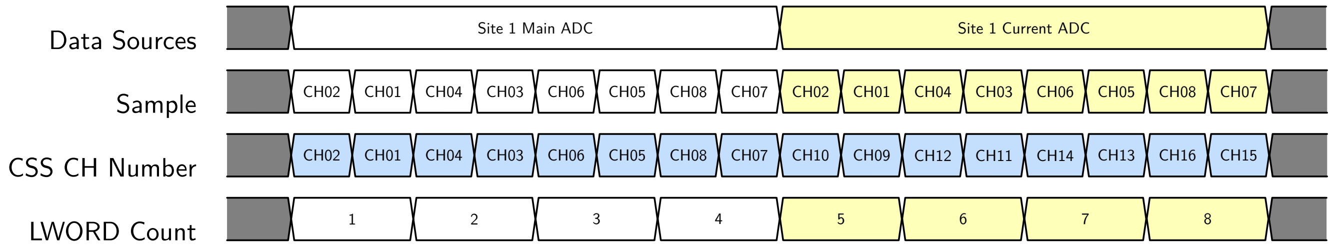


Figure 2: Raw Data Format

Max Data Rate = 8 LWords(4B) × 6 sites × 1 MHz = 192 MB/*s*

2.2 DSP Data

In the following Figures :

A = Voltage amplitude from the quadrature synchronous detection
ϕ = Phase from the quadrature synchronous detection
P = Real-time power
V_{OH} = Ohmic heating voltage, I_{OH} = Ohmic heating current

2.2.1 Normal Operation

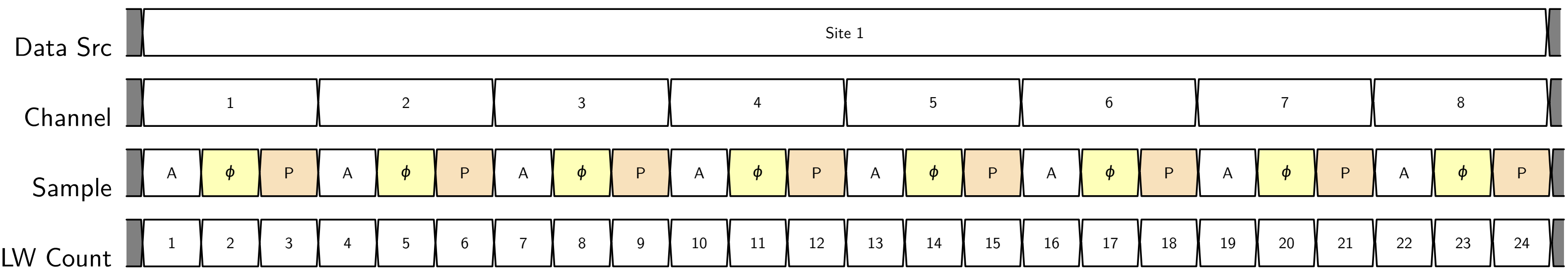


Figure 3: DSP Data Format during Calibration

2.2.2 During Calibration

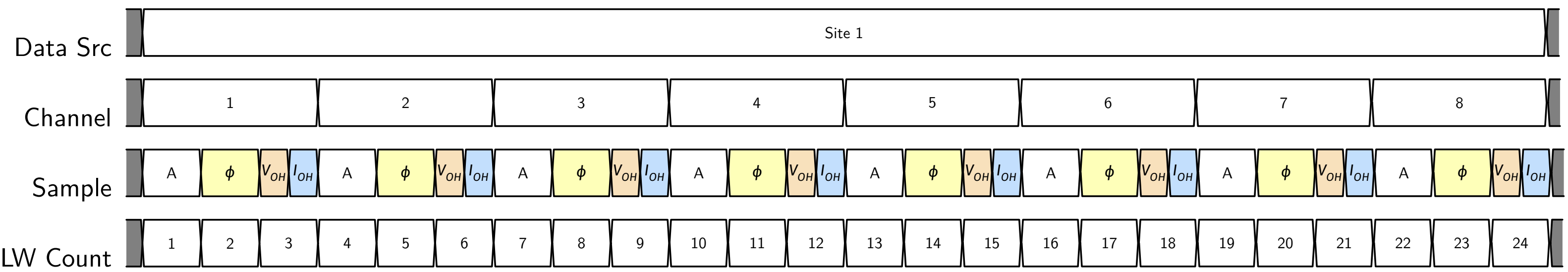


Figure 4: DSP Data Format

Max Data Rate = 24 LWords(4B) × 6 sites × 10 kHz = 5.76 MB/*s*

3 HAPI - Host App Python Interface

See repo on Github for install instructions.

In Section 4 we demonstrate the new streamlined flow driven by the `bolo8_cal_cap_loop.py` script included in the acq400_hapi repo.

3.1 bolo8_cal_cap_loop.py

```
[dt100@eigg-fs acq400_hapi]$ ./user_apps/special/bolo8_cal_cap_loop.py -h
usage: bolo8_cal_cap_loop.py [-h] [--cap CAP] [--cal CAL] [--cc CC]
                             [--single_calibration_only SINGLE_CALIBRATION_ONLY]
                             [--post POST] [--clk CLK] [--trg TRG]
                             [--shots SHOTS] [--active_chan ACTIVE_CHAN]
                             [--fpgpio_strobe FPGPIO_STROBE]
                             uuts [uuts ...]

bolo8_cal_cap_loop

positional arguments:
  uuts                  uut list

optional arguments:
  -h, --help            show this help message and exit
  --cap CAP             capture
  --cal CAL             calibrate
  --cc CC               --cc=1 sets cap=1,cal=0; --cc=2 => cap=0,cal=1; --cc=3
                        => cal=1,cap=1
  --single_calibration_only SINGLE_CALIBRATION_ONLY
                        run one calibration shot only
  --post POST           post trigger length
  --clk CLK             clk "int|ext SR [CR]"
  --trg TRG             trg "int|ext rising|falling"
  --shots SHOTS         set number of shots [1]
  --active_chan ACTIVE_CHAN
                        comma separated list of active channels, ; to split
                        between uuts (because not all channels have foils)
  --fpgpio_strobe FPGPIO_STROBE
                        custom lamp control: 0: OFF, 1:ON >1: flash at N Hz
```

4 Simulating Operation - Lamp Test

4.1 Setup

The D-TACQ BOLO8-BLF test panel uses a ZIF (zero insertion force) socket to hold the bolometer and allows the user to measure various test points on the bolometer when in use.

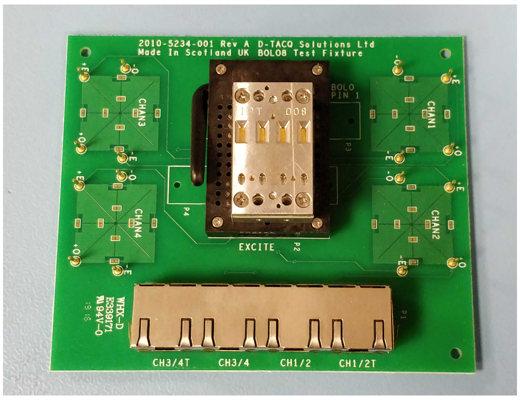


Figure 5: Bolometer fitted to the test board ZIF socket

Figure 6: Bolometer connected to a BOLO8 mezzanine in an acq2106 system

Some form of strobing light source is required as an input to the bolometer foils to witness the response detailed below.

The Python scripts used in this section are covered in more detail in Section 3 of this document.

4.2 CSS

Throughout this section the relevant OPI buttons to launch the current view are highlighted in red.

CS Studio can be used to view data from the acq2106. This is extremely useful for initial validation and testing as it allows the user to verify that the system is working very quickly.

Open the BOLO8 Launcher from the Navigator pane.
If present, close the ACQ400 Launcher (this will prevent duplicate plots spawning).

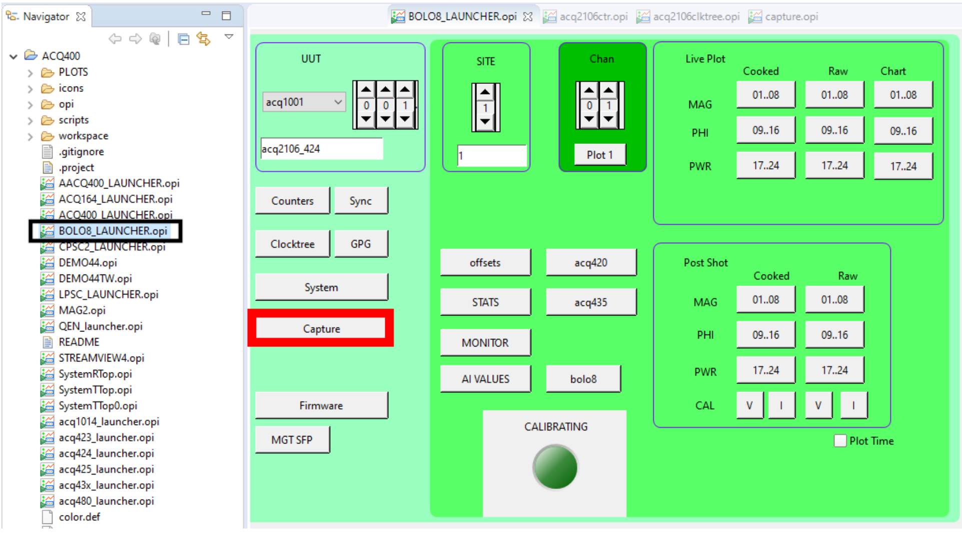


Figure 7: CS Studio screen showing where to access the BOLO8 Launcher, Capture OPI highlighted in red

4.2.1 Streaming Data

Open the Capture OPI (Figure 7) and select the “Stream” tab.

START a Stream and open the plot window highlighted in Figure 8. Apply a strobing light source to the bolometer foils and you should see a familiar heating/cooling curve appearing on the plot. Note that you may have to increase the strobe rate to see more cycles within this window.

Figure 8: CS Studio screen showing live MAG data for the selected bolo foils

Because it presents a “live view” of the data and quick feedback, streaming is useful when experimenting with the response of bolometer foils to a light source.

4.2.2 Pre-Cal Data Capture

Capture some data (before performing a calibration) :

```
[dt100@eigg-fs acq400_hapi]$ ./user_apps/special/bolo8_cal_cap_loop.py --cal=0 --cap=1 --shots=1 acq2106_123
Cycle 1
trigger
```

The shark fin shapes are produced as the foils heat up and cool down when the LEDs run through the strobe pattern.

Note that the channels have not been calibrated and so the offsets have not been accounted for yet. This means that some channels likely will be visually offset from one another.

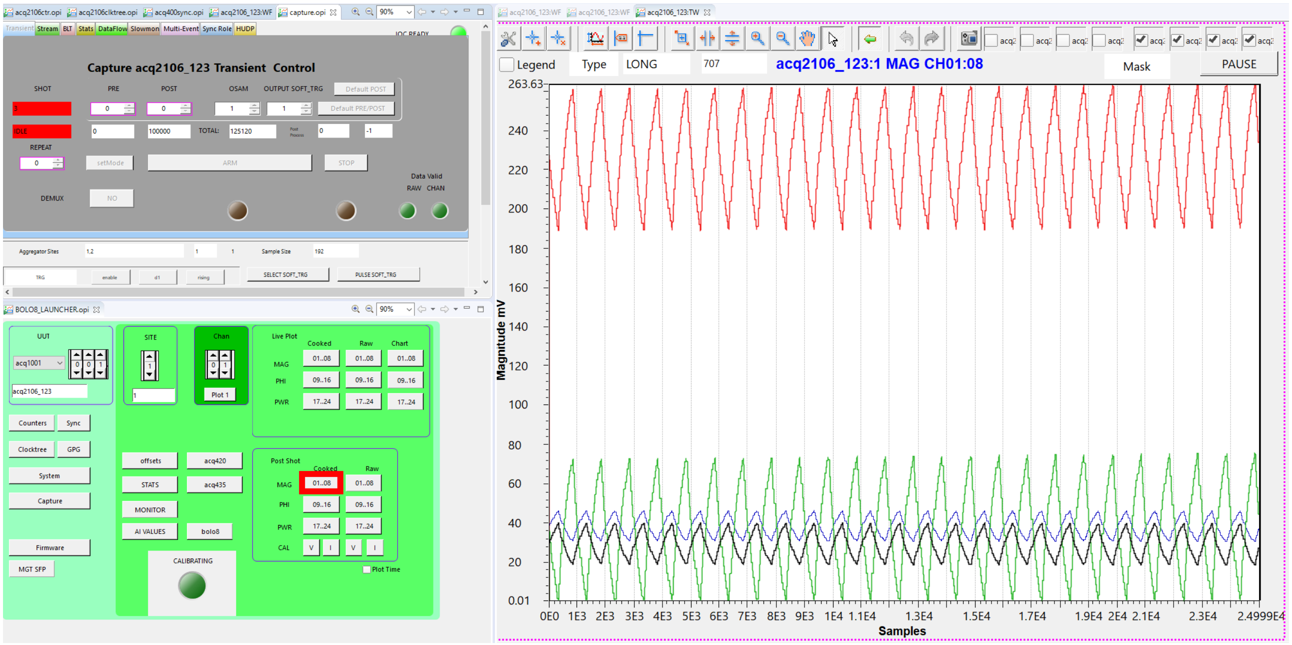


Figure 9: CS Studio screen showing 4 channels without calibration. Note the offsets between channels

4.3 Calibration

Beware: In a room with constantly changing light/temperature levels your calibration from 30 mins ago may not still be valid. Customers are recommended to calibrate at the beginning of every shot. Remember to remove the light source from the bolometer before carrying out a calibration.

Here, we only have bolometers connected to channels 1 through 4, hence the use of the `active_chan` argument. If all channels are in use the user can omit this argument.

```
[dt100@eigg-fs acq400_hapi]$ ./user_apps/special/bolo8_cal_cap_loop.py --cal=1 --cap=0 --shots=1 --active_chan=1,2,3,4 acq2106_123
Cycle 1
>
>Old : acq2106_123:B8:CALIBRATING     0
>New : acq2106_123:B8:CALIBRATING     1
>filter_status 33 vr 1 pw 1
>
>custom CALIBFITARGS -H 0.6
>Setting GAINS to 1V2 and reset OSDACs before running.
>calibration starts for channels "1 2 3 4"
>filter_status 33 vr 1 pw 1
>
>timeout 2
>timeout 6
>
>Fitting data
>calibfit took 0m 4.12s
>1          5.0997        0.048074       -0.011584407  0.018125918
>2          5.2063        0.045411       0.080704018   -0.080880105
>3          5.5033        0.044986       0.014283978   -0.013639961
>4          5.6799        0.046298       -0.0086123683 0.0091954544
>power_filter_design.tcl 1 5.0997 0.048074
>power_filter_design.tcl 2 5.2063 0.045411
>power_filter_design.tcl 3 5.5033 0.044986
>power_filter_design.tcl 4 5.6799 0.046298
>
>load_offset_channel.tcl 1 -0.011584407 0.018125918 5.0997
>load_offset_channel.tcl 2 0.080704018 -0.080880105 5.2063
>load_offset_channel.tcl 3 0.014283978 -0.013639961 5.5033
>load_offset_channel.tcl 4 -0.0086123683 0.0091954544 5.6799
>filter_status 33 vr 1 pw 1
>
>Old : acq2106_123:B8:CALIBRATING     1
>New : acq2106_123:B8:CALIBRATING     0
>END
logging to ./cal_1.log
```

The result of the calibration shows the heating/cooling pulse in both voltage and current terms. These plot OPIs are available in the BOLO8 Launcher (Post Shot Cooked CAL V & I).

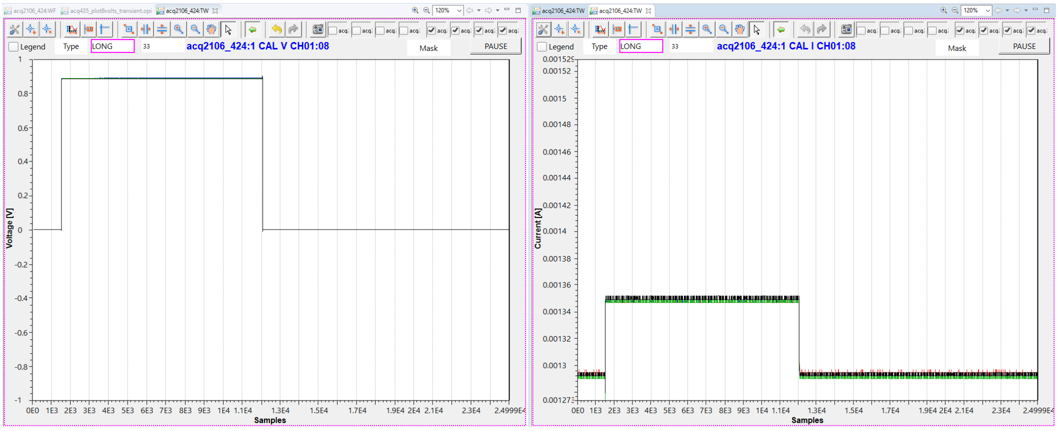


Figure 10: Offset voltage and current for 4 channels during calibration

4.4 Post-Calibration Data Capture

The same data capture can be performed after calibrating the system. Re-apply the light source to the foils to see the calibrated response.

```
[dt100@eigg-fs acq400_hapi]$ ./user_apps/special/bolo8_cal_cap_loop.py --cal=0 --cap=1 --shots=1 acq2106_123
Cycle 1
trigger
```

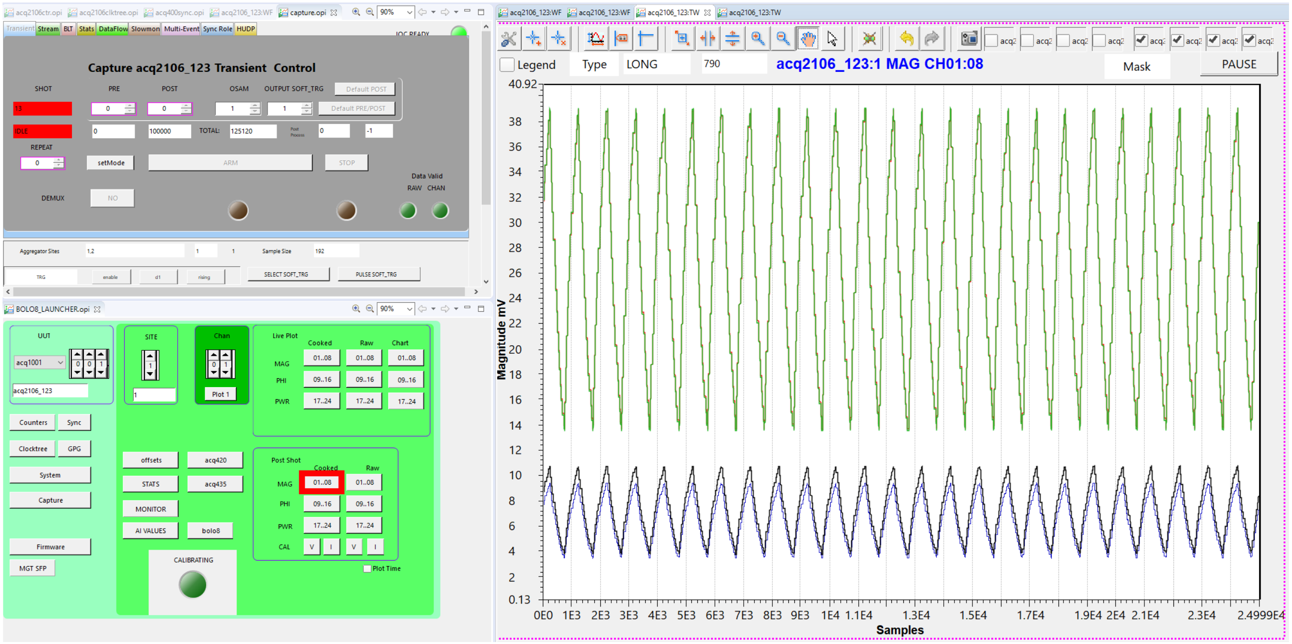


Figure 11: CS Studio screen showing 4 channels post calibration. Offsets calibrated out

Note that channel offsets have been largely calibrated out. We assume that the varying fin amplitude may be attributed to imperfect positioning of the LED module above the foils and/or differing foil characteristics.

4.5 Real-Time Power

Once we have run a calibration and characterised the attached foils, we can use some of the figures of merit from the calibration to produce a real-time power plot. This takes our shark fin magnitude trace and converts it to a real-time incident power.

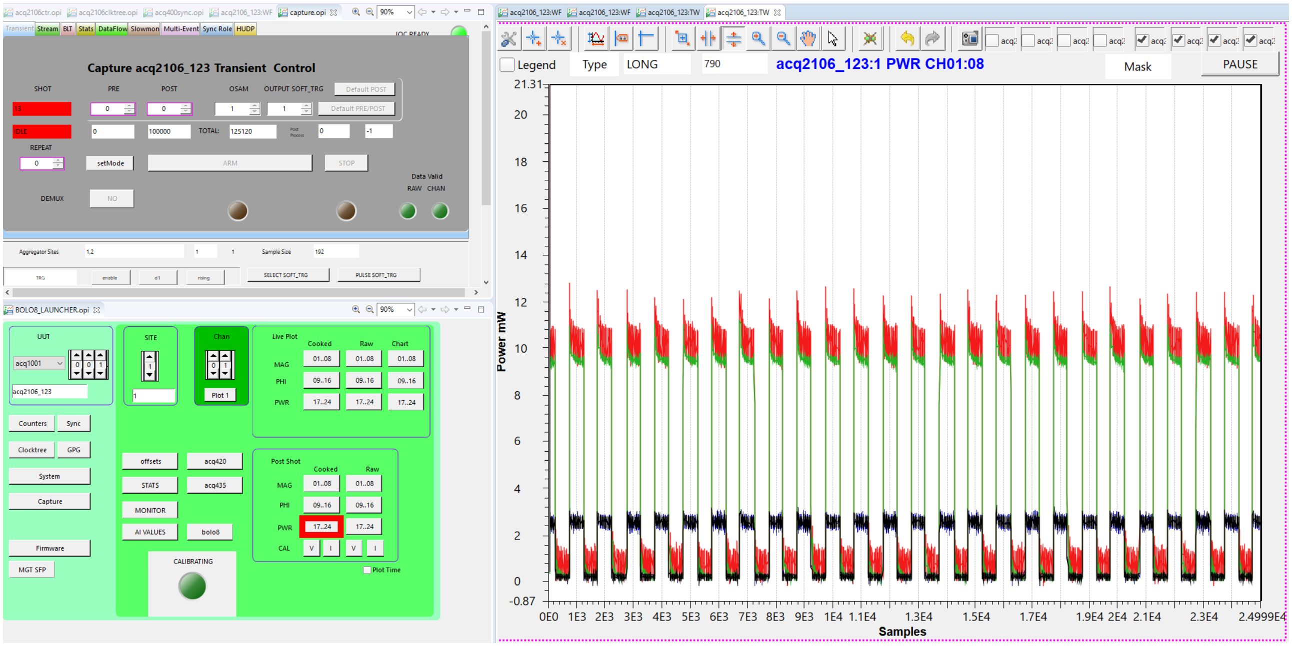


Figure 12: Real-time power CSS plot

A Validated Releases

Date	Release	bolodsp	Description
08/12/2022	REL-432		Shipment to customer
30/03/2023	REL-579		Internal upgrades and testing
25/05/2023	REL-601	2305191548	First shipment to large US customer
XX/07/2023	REL-XXX	2307181414	Updates to include automated lamp tests
08/11/2023	REL-620	2307181414	DOC-228070-01

Table 1: Table of FW Releases specifically validated with BOLO

B Snag list

Date	Release	User	Snag Description
30/11/2022	REL-542	SR	The calibration process broke. It would not complete a shot and spammed "timeout" on to the terminal. This was due to an incompatibility in /usr/local/bin/wait_until_state
30/11/2022	REL-542	SR	On this particular release, EPICS crashes when we "Set ARM" on a transient. Only with bolodsp package instantiated. See Bug 180
07/12/2022	REL-546	SR,BK	With bolodsp package instantiated the run0 calcualtion is incorrect. It is treating the BOLO channels as 16b rather than 32b. Check site 0 NCHAN and ssb variables
07/12/2022	REL-432	SR,BK,AL	Without bolodsp package (Production test mode) the Site 1 clock comes up at 10 kHz rather than the desired 1 MHz

Table 2: Ongoing BOLO Snaglist

C Background

There is now **ONE** version of the BOLO FPGA. It includes the BOLO DSP module but we can bypass the DSP to see raw ADC input on CSS.

By default, if the bolodsp package is not enabled then the DSP boots in BYPASS=1.

C.1 Analogue Supplies

In `/usr/local/CARE/choose_vap` we set the analogue supplies for all BOLO boxes to ±12 V in order to minimise the heat dissipation in the driver op-amp.

WARNING : Pots for the analogue supplies will not be present on acq2106s manufactured later than circa 2022!
To set rails manually :

Add Resistor of value 5K49 to R292 for the +ve analog rail

Add Resistor of value 5K9 to R299 for the -ve analog rail

C.2 Single RJ45 Pinout

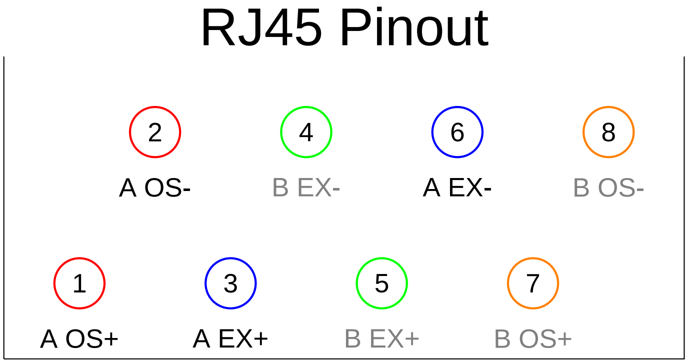


Figure 13: Single RJ45 Pinout for Probing on a Mezzanine

Nowadays it is probably easier to breakout to one of the accessory boards and probe on the test points but the RJ45 pinout is provided here for reference.

D Current Sense Circuit Detail

- Sample rate set in logic to 200 kHz
- The ADC output is then averaged (x20) in the FPGA logic to produce an update rate of 10 kHz
- More averaging (x100) is applied in the DSP Module.
  - These averages are inexact, in that we accumulate over a decimal number, then right-shift by 2^x.
  - We apply scaling factors of $\frac{32}{20}$ and $\frac{128}{100}$ to account for this.

Originally this was a 16-bit ADC but due to component shortages a 12-bit ADC from the same family was subbed in. It was determined that due to the amount of averaging being performed, 12-bit performance could be accepted.

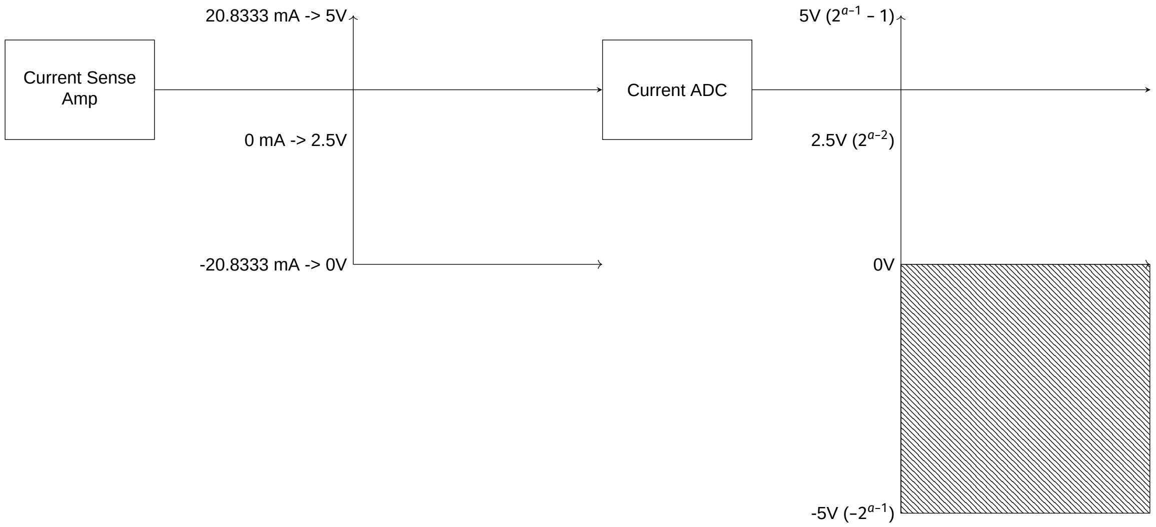


Figure 14: Current Sense Circuit and Ranges

The output of the current sense amp is centred around 2.5 V, so we have to account for this offset once we receive our result from the ADC.

$$\text{I (mA)} = \frac{(ADC_{Vin} - 2.5)}{0.12} \tag{1}$$

Thus, to convert ADC_{codes} to a value for current we should use one of the following expressions.

$$\text{I (mA)} = (ADC_{codes} - 2^{a-2}) \times \frac{10}{2^a} \times \frac{25}{3} \tag{2}$$

OR

$$\text{I (mA)} = \left(ADC_{codes} \times \frac{10}{2^a} - 2.5\right) \times \frac{25}{3} \tag{3}$$

where, a = ADC resolution

$\frac{25}{3}$ = Scaling factor to convert from volts to mA

D.1 Accounting for scaling issue in averaging

$$\text{I (mA)} = \left(ADC_{codes} \times \frac{10}{2^a} \times \frac{32}{20} \times \frac{128}{100} - 2.5\right) \times \frac{25}{3} \tag{4}$$

E Production Test

This section details the D-TACQ Production Test to verify board operation. It is included here for completeness but we do not anticipate customers undertaking this procedure.

In the Production Test we use the BOLO FPGA image with the DSP in BYPASS. This simplifies things massively, and we return to a board which contributes 8 channels of ADC to the Aggregator.

See Appendix F for a streamlined look at how to run a Production Test and complete a test sheet.

E.1 Production Test Script

```
/usr/local/CARE/bolo-pro-test
```

The bolo-pro-test script is used to set offsets to verify the operation of the Offset DACs during production test.

Running it once after boot will also start the Excite waveform output.

It takes a single argument which corresponds to the Offset DAC value in codes.

```
acq2106_123> /usr/local/CARE/bolo-pro-test 0
acq2106_123> /usr/local/CARE/bolo-pro-test 1fff # 1.875 V
acq2106_123> /usr/local/CARE/bolo-pro-test 3fff # 3.75 V
acq2106_123> /usr/local/CARE/bolo-pro-test 5fff # 5.625 V
acq2106_123> /usr/local/CARE/bolo-pro-test 7fff # 7.5 V
```

See Figure 19 for expected results at each offset level.

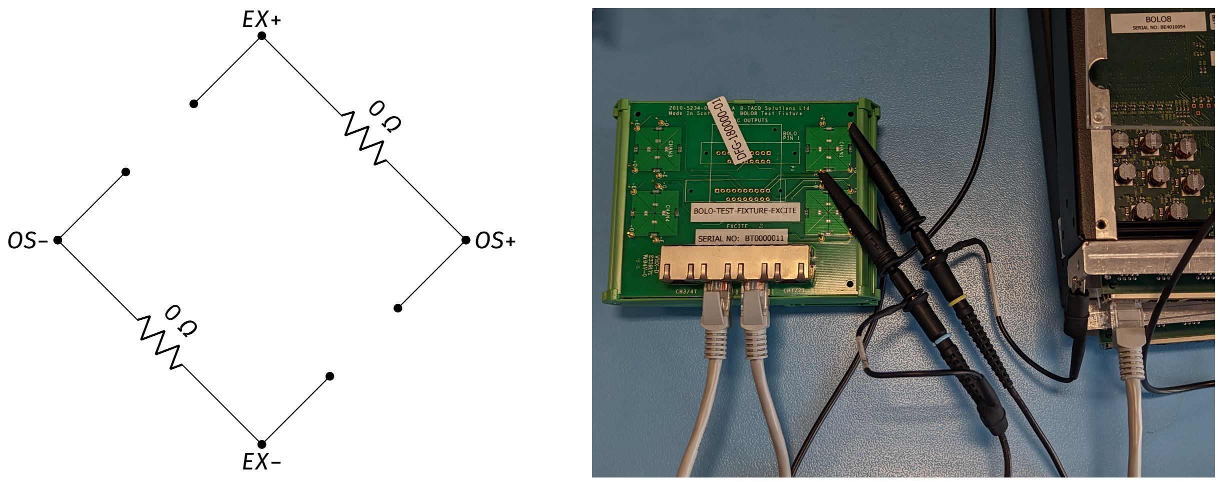
There is also a `loop` argument which cycles through `{ 0 1fff 3fff 5fff 7fff 7fff 0 }`. You may wish to redirect the output of this to `/dev/null` to keep your console cleaner.

```
acq2106_123> /usr/local/CARE/bolo-pro-test loop > /dev/null
```

E.2 Test Accessories

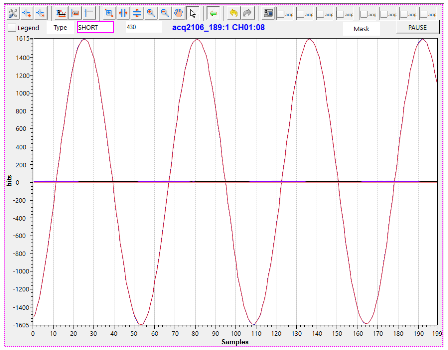
E.2.1 Excite Tester

This test accessory connects the Excite DAC outputs directly to the Main ADC in the bolometer mezzanine. This means we can directly observe the excite sine waves in CSS.

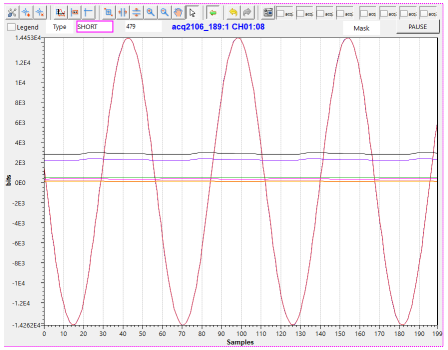


(a) Excite Loopback (b) Probes on EX+ and EX- measuring output from Excite DAC

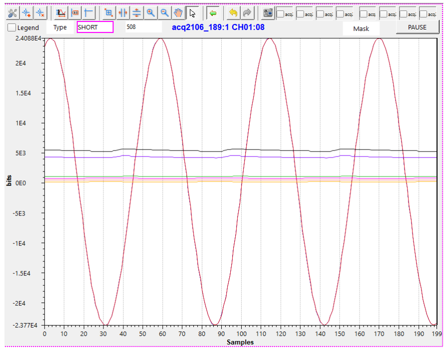
Figure 15: Excite Accessory Circuit and Probe Points



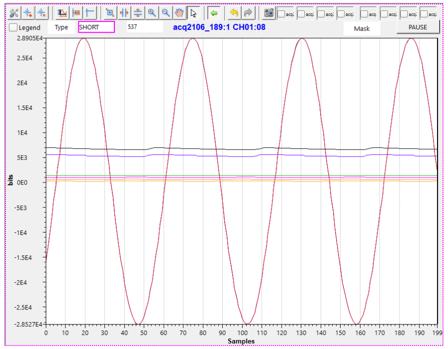
(a) 1 V Excite



(b) 9 V Excite



(c) 15 V Excite



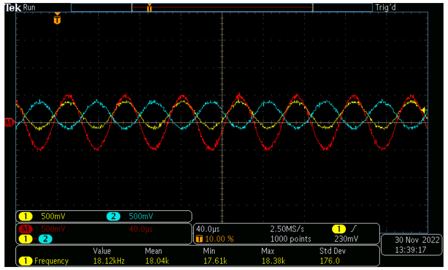
(d) 18 V Excite

Figure 16: All 4 Excite amplitudes displayed in a CSS Raw Plot

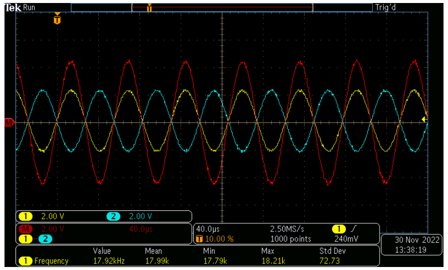
$$+1600\ \mathit{codes} = 1600 \times \frac{10}{2^{15}} \approx 500\ \mathrm{mV}, \times 2 \approx 1\ \mathrm{V}_{\text{p-p}}$$

```
acq2106_189> map /usr/local/acq2106.map # Use these commands to change the amplitude
mm $DSP1+18 0 # 1 V; mm $DSP1+18 1 # 9 V; mm $DSP1+18 2 # 15 V; mm $DSP1+18 3 # 18 V
```

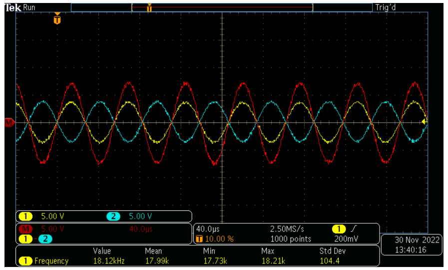
Code Amplitude	Volts Amplitude	Volts Peak-Peak
1600	≈ 500 mV	1 V
14500	≈ 4.5 V	9 V
24100	≈ 7.5 V	15 V
28900	≈ 9 V	18 V



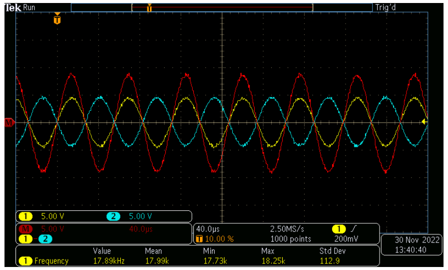
(a) 1 V Excite



(b) 9 V Excite



(c) 15 V Excite



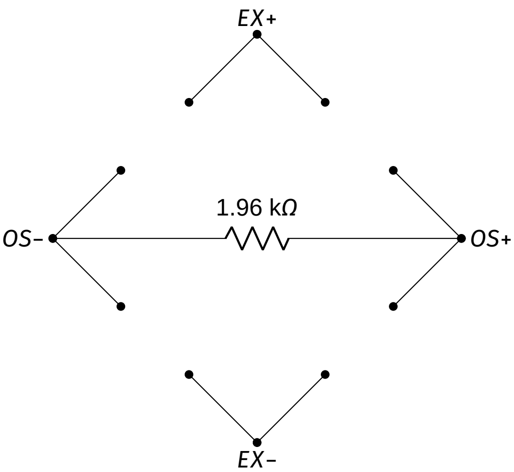
(d) 18 V Excite

Figure 17: All 4 Excite amplitudes measured on an oscilloscope

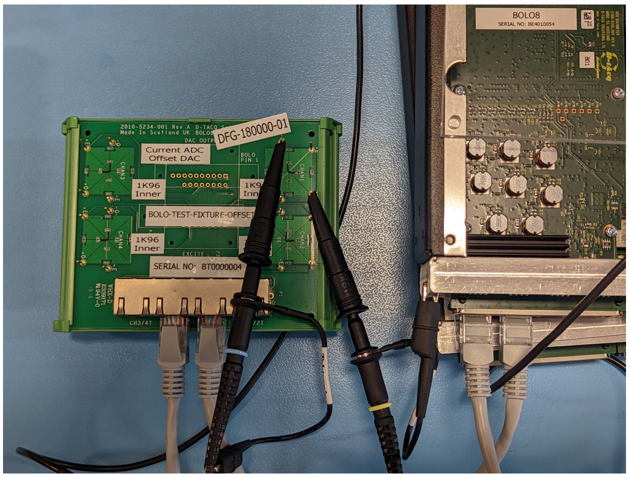
E.2.2 Offset Tester

This test accessory connects the Offset DAC outputs to the Main ADC via a 1.96 kΩ resistor (to provide a load, but control current).

Offset DAC maximum amplitude output is ±7.5 V. Anything above 5V (0x5555) will saturate when looped back to the Main ADC.
This includes 0x5fff and 0x7fff from the `bolo-pro-test` script.

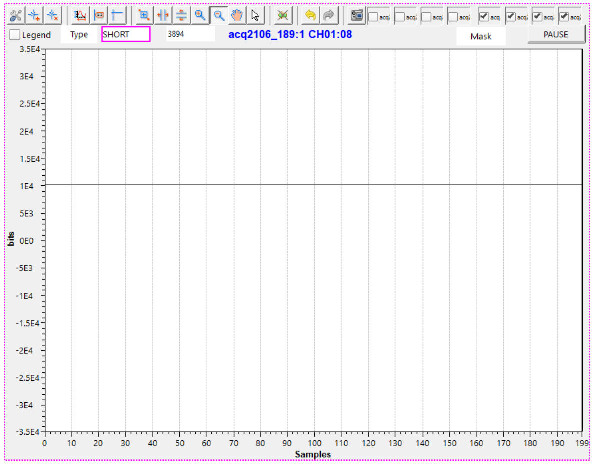


(a) Offset Load/Loopback

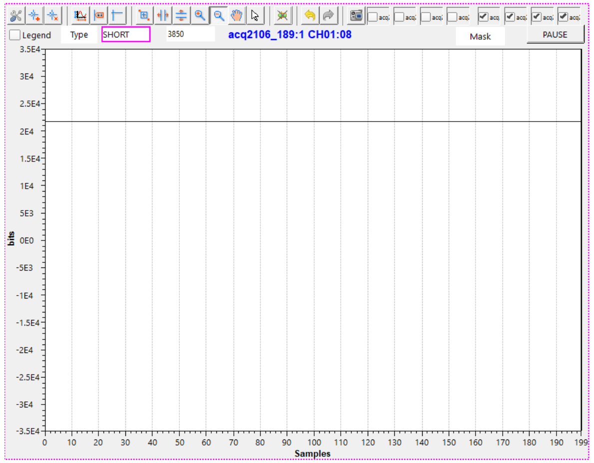


(b) Probes on OS+ and OS- measuring output from Offset DAC

Figure 18: Offset Accessory Circuit and Probe Points



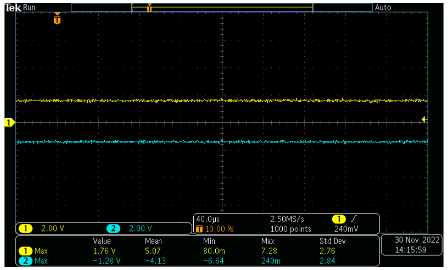
(a) 0x1fff Offset



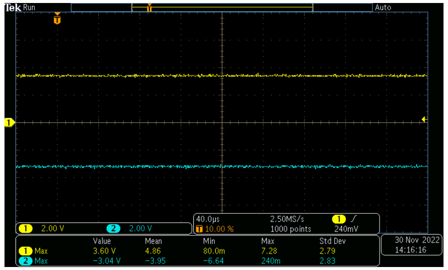
(b) 0x3fff Offset

(c) 0x5fff, 0x7fff Saturated Offset

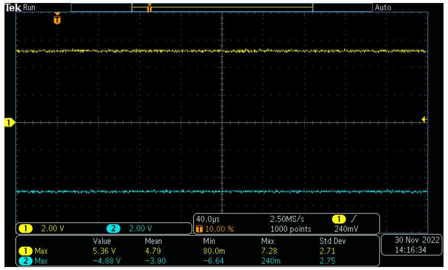
Figure 19: Offset DAC amplitudes displayed in a CSS Raw Plot



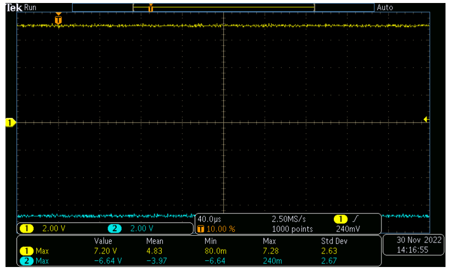
(a) 0x1fff ≈ 1.875 V Offset



(b) 0x3fff ≈ 3.75 V Offset



(c) 0x5fff ≈ 5.625 V Offset



(d) 0x7fff ≈ 7.5 V Offset

Figure 20: Offset DAC amplitudes measured on an oscilloscope

E.3 Current Sense Test

With the Offset Tester accessory in the circuit we have a known resistance between *OS+* and *OS−*, this allows us to compare our Current ADC readings with expected values.

The Current ADC values will appear in CH09-16 in a BOLO mezzanine site.

DAC Offset Codes	DAC Volts	Calc. I (mA)	ADC Codes	Meas. I (mA)	Diff (uA)	Stats OPI
0	0	0	640	0	0	0.194
1fff	1.875	1.9	685	1.5	448.4	0.209
3fff	3.75	3.8	743	3.4	473.7	0.226
5fff	5.625	5.7	800	5.2	531.5	0.244
7fff	7.5	7.7	858	7.1	556.6	0.262

Table 3: 12-bit Current ADC Representative Values

DAC Offset Codes	DAC Volts	Calc. I (mA)	ADC Codes	Meas. I (mA)	Diff (uA)	Stats OPI
0	0	0	10202	-0.0773	77.3	3.113
1fff	1.875	1.9	10997	1.5	373.1	3.356
3fff	3.75	3.8	11917	3.4	414.7	3.637
5fff	5.625	5.7	12843	5.3	444.0	3.192
7fff	7.5	7.7	13772	7.2	467.2	4.204

Table 4: 16-bit Current ADC Representative Values

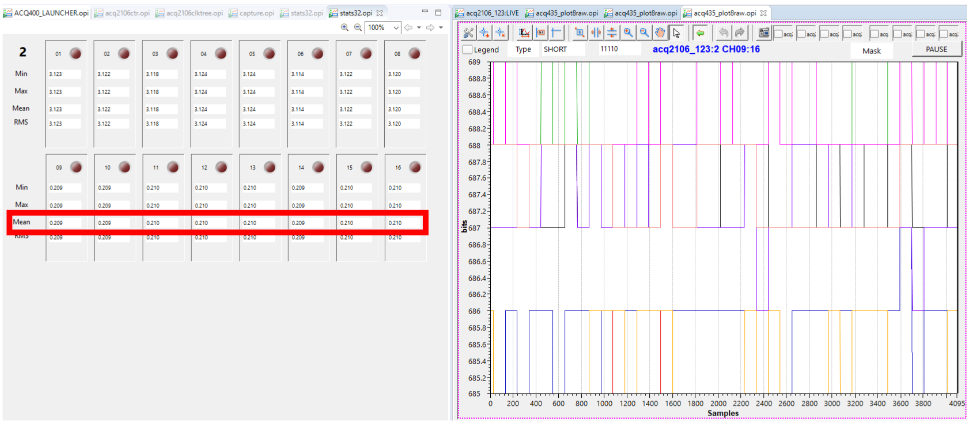


Figure 21: Example CSS OPIs showing a 12-bit Current ADC with the Offset DACs set to 1fff

E.4 RJ45 LED Test

Switch on all LEDs in the mimic and check they all light.

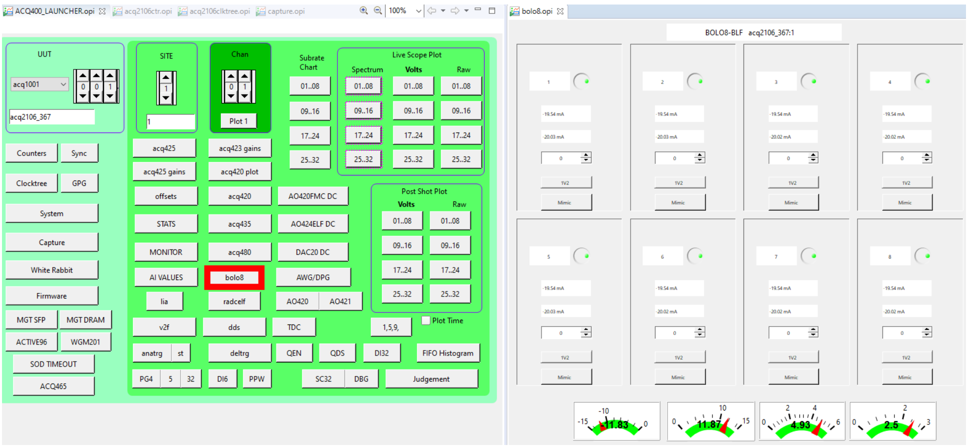


Figure 22: Example CSS OPIs showing B8 Controls for RJ45 LEDs

```
state=1; UUT=acq2106_367; tot_site=4; for site in $(seq 1 $tot_site);do for led in $(seq 1 8);do caput $UUT:$site:B8:LED${led} $state;done;done # ON
state=0; UUT=acq2106_367; tot_site=4; for site in $(seq 1 $tot_site);do for led in $(seq 1 8);do caput $UUT:$site:B8:LED${led} $state;done;done # OFF
```

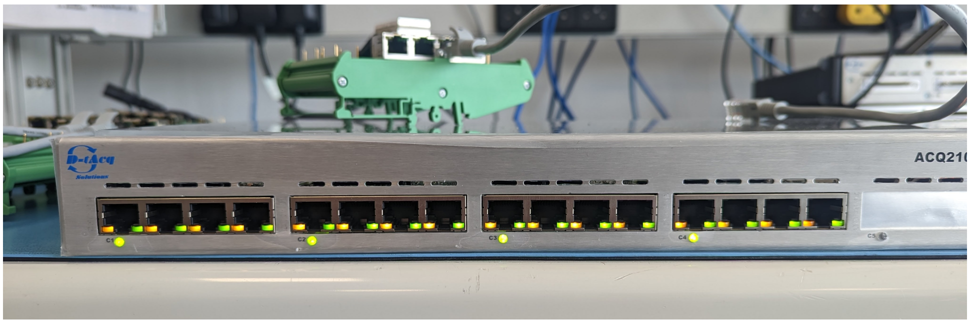


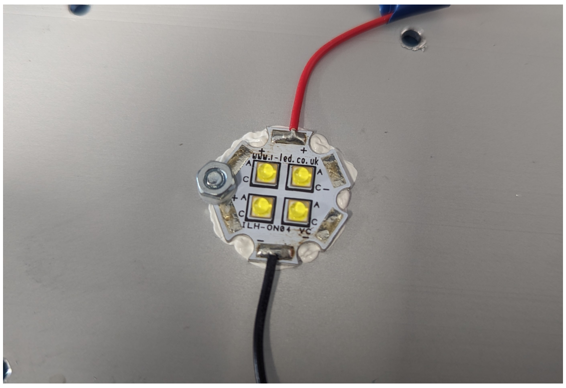
Figure 23: Example front panel showing B8 Mezzanine RJ45 LEDs

E.5 Factory Calibration & Lamp Test

E.5.1 Lamp Box

Connect the LED PWM control to the front panel GPIO LEMO.

(a) Bolometer head with (LEDs overhead)



(b) High power LED module

Figure 24: Component parts of the test apparatus

E.5.2 bolo8_tester.py

```
commit 813d62e7e4cd1b0a240fd608a85ee5c30c52c0ff
```

```
[dt100@eigg-fs acq400_hapi]$ ./test_apps/bolo8_tester.py -h
usage: bolo8_tester.py [-h] [--chans CHANS] [--cycles CYCLES]
                       [--bolo_chans BOLO_CHANS] [--bolo_id BOLO_ID]
                       [--beeper BEEPER] [--calcap CALCAP] [--plot PLOT]
                       [--tocsv TOCSV] [--save SAVE] [--url URL]
                       uutname

bolo8 tester

positional arguments:
  uutname               uut name

optional arguments:
  -h, --help            show this help message and exit
  --chans CHANS         channels to test eg 1,2,3,4,5,6
  --cycles CYCLES       number tests on each channel
  --bolo_chans BOLO_CHANS
                        Number of avalible boloemnter channels?
  --bolo_id BOLO_ID     bolometer id
  --beeper BEEPER       Sig gen to beep when channel done
  --calcap CALCAP       1 = cal, 2 = cap, 3 = cal+cap
  --plot PLOT           0 no plot, 1 plot
  --tocsv TOCSV         save calibration results to csv
  --save SAVE           save capture data to file
  --url URL             remove url to send file
```

By default the test runs a calibration shot on each pair of channels in an RJ45.
It then asks the user to move the cable connecting the bolometer head to the next RJ45.

Results are stored on eigg, here : `/home/dt100/CAL/BOLO_CAL/SERIAL_NUMBER` e.g.

```
[dt100@eigg-fs BOLO_CAL]$ ls -l BE4010080/2023.Oct.25.11.13/
total 24
-rw-r--r-- 1 apache apache  327 Oct 25 11:13 bolo_sh.bak
-rw-r--r-- 1 apache apache  229 Oct 25 11:13 cal_cooked.csv
-rw-r--r-- 1 apache apache  148 Oct 25 11:13 cal_raw.csv
-rw-r--r-- 1 apache apache  440 Oct 25 11:13 fpga.bak
-rw-r--r-- 1 apache apache  595 Oct 25 11:13 packages.bak
-rw-r--r-- 1 apache apache 2223 Oct 25 11:13 test_config.json
[dt100@eigg-fs BOLO_CAL]$ cut -d',' -f1,2,3,11 BE4010080/2023.Oct.25.11.13/cal_cooked.csv
Module,Channel,sensitivity_mean,status
BE4010080,3,4.601,Pass
BE4010080,4,4.6573,Pass
```

E.5.3 Running a calibration

As dt100 on eigg use the alias to get to the HAPI directory.

```
[dt100@eigg-fs ~]$ HAPI
[dt100@eigg-fs acq400_hapi]$ pwd
/home/dt100/PROJECTS/ACQ400/HAPI/acq400_hapi
```

Set the `UUT` and `--chans` arguments to suit your configuration. If you do not wish to save the results to eigg, omit the `--url` argument.

```
./test_apps/bolo8_tester.py --url=http://eigg/endpoint/bolo UUT --chans=1-8
```

E.5.4 Pre-Cal Data Capture

Capture some data (without running a calibration) :

```
[dt100@eigg-fs acq400_hapi]$ ./test_apps/bolo8_tester.py --foils=4 --chans=1-4 --calcap=2 acq2106_424
Testing acq2106_424 with bolo None
Setting active chan to 1,2,3,4
Is bolometer (None) connected to channels 1,2,3,4? [yes, quit]: y

Cycle 1/1

 Running capture on channels 1,2,3,4
Priming strobe
Arming
Triggering
Stopped
operands could not be broadcast together with shapes (55,) (18,)
Fail: CH1 CH2 comparison failed
operands could not be broadcast together with shapes (55,) (18,)
Fail: CH1 CH3 comparison failed
operands could not be broadcast together with shapes (55,) (2971,)
Fail: CH1 CH4 comparison failed
1 cycles took 33.58s

Capture results:
Channels 2,3,4 out of sync
Plotting PWR
Saving test config to results/acq2106_424/2023.Nov.06.12.46/acq2106_424.None.json
```

The shark fin shapes are produced as the foils heat up and cool down when the LEDs run through the strobe pattern.

Note that the channels have not been calibrated and so the offsets have not been accounted for yet. This means that some channels likely will be visually offset from one another.

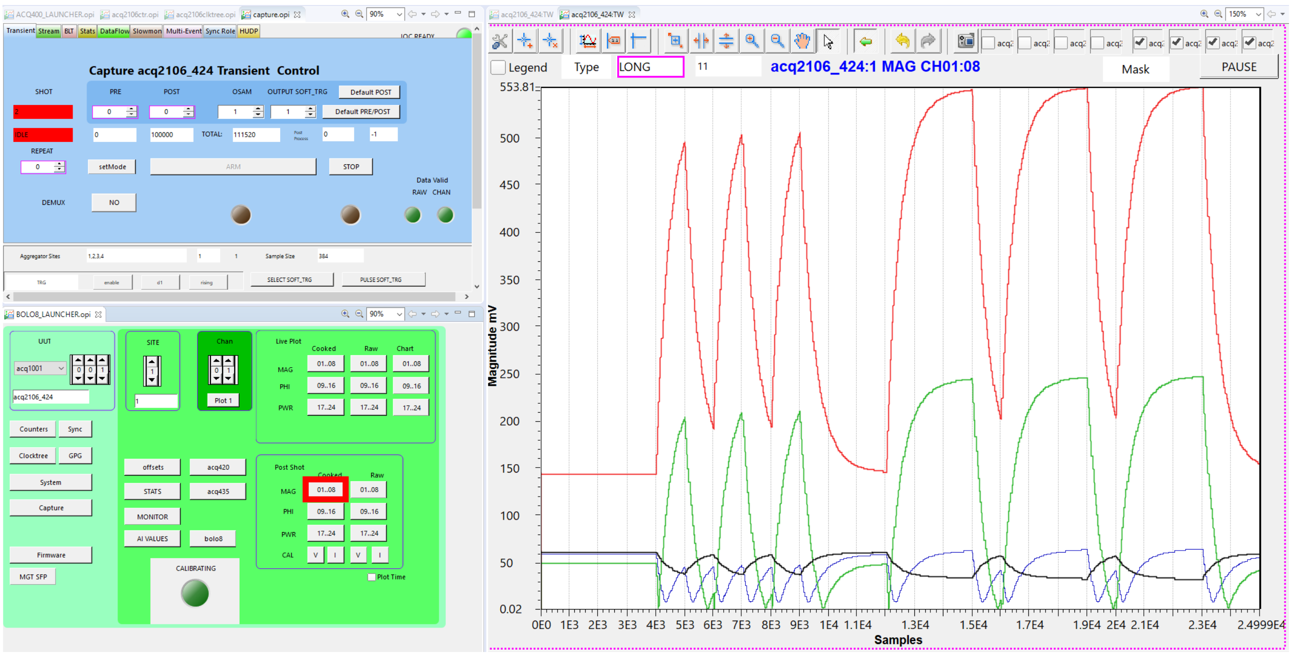


Figure 25: CS Studio screen showing 4 channels without calibration. Note the offsets between channels

E.5.5 Calibration

Beware: In a room with constantly changing light/temperature levels your calibration from 30 mins ago may not still be valid. Customers are recommended to calibrate at the beginning of every shot.

```
[dt100@eigg-fs acq400_hapi]$ ./test_apps/bolo8_tester.py --foils=4 --chans=1-4 --calcap=1 acq2106_424
Testing acq2106_424 with bolo None
Setting active chan to 1,2,3,4
Is bolometer (None) connected to channels 1,2,3,4? [yes, quit]: y

Cycle 1/1
 Running calibration on channels 1,2,3,4
>
>Old : acq2106_424:B8:CALIBRATING     0
>New : acq2106_424:B8:CALIBRATING     1
>filter_status 33 vr 1 pw 1
>
>custom CALIBFITARGS -H 0.6
>Setting GAINS to 1V2 and reset OSDACs before running.
>calibration starts for channels "1 2 3 4"
>filter_status 33 vr 1 pw 1
>
>Fitting data
>calibfit took 0m 7.29s
>1         5.4419        0.049696       -0.017797315  0.028012959
>2         5.7088        0.04952        0.046383116   -0.055406455
>3         6.1951        0.048582       -0.015809065  0.019227464
>4         5.9723        0.047507       -0.019447796  0.023801725
>power_filter_design.tcl 1 5.4419 0.049696
>power_filter_design.tcl 2 5.7088 0.04952
>power_filter_design.tcl 3 6.1951 0.048582
>power_filter_design.tcl 4 5.9723 0.047507
>
>load_offset_channel.tcl 1 -0.017797315 0.028012959 5.4419
>load_offset_channel.tcl 2 0.046383116 -0.055406455 5.7088
>load_offset_channel.tcl 3 -0.015809065 0.019227464 6.1951
>load_offset_channel.tcl 4 -0.019447796 0.023801725 5.9723
>filter_status 33 vr 1 pw 1
>
>Old : acq2106_424:B8:CALIBRATING     1
>New : acq2106_424:B8:CALIBRATING     0

 Calibration results :
Chan 1   Sensitivity 5.4419    Cooling 0.049696
Chan 2   Sensitivity 5.7088    Cooling 0.04952
Chan 3   Sensitivity 6.1951    Cooling 0.048582
Chan 4   Sensitivity 5.9723    Cooling 0.047507
1 cycles took 51.97s

 Calibration results (averaged):
Chan 1   Sensitivity 5.4419    Cooling 0.0497     Pass
Chan 2   Sensitivity 5.7088    Cooling 0.04952    Pass
Chan 3   Sensitivity 6.1951    Cooling 0.04858    Pass
Chan 4   Sensitivity 5.9723    Cooling 0.04751    Pass
Saving calibration to results/acq2106_424/2023.Nov.06.12.50/acq2106_424.None.csv
Saving test config to results/acq2106_424/2023.Nov.06.12.50/acq2106_424.None.json
```

The result of the calibration shows the heating/cooling pulse in both voltage and current terms.
These plot OPIs are available in the BOLO8 Launcher (Post Shot Cooked CAL V & I).

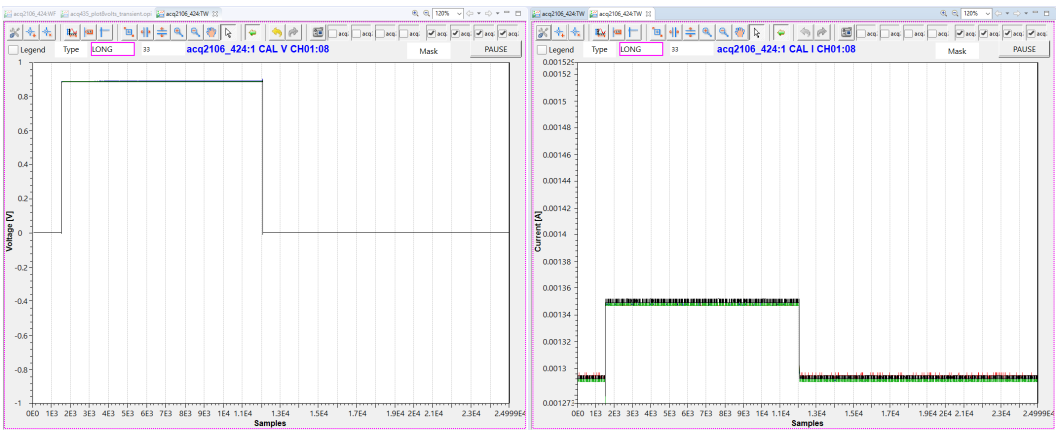


Figure 26: Offset voltage and current for 4 channels during calibration

E.5.6 Post-Calibration Data Capture

The same data capture can be performed after calibrating the system.

```
[dt100@eigg-fs acq400_hapi]$ ./test_apps/bolo8_tester.py --foils=4 --chans=1-4 --calcap=2 acq2106_424
Testing acq2106_424 with bolo None
Setting active chan to 1,2,3,4
Is bolometer (None) connected to channels 1,2,3,4? [yes, quit]: y

Cycle 1/1

 Running capture on channels 1,2,3,4
Priming strobe
Arming
Triggering
Stopped
Pass: CH1 CH2 in sync
Pass: CH1 CH3 in sync
Pass: CH1 CH4 in sync
1 cycles took 33.73s

Capture results:
All channels in sync
Plotting PWR
Saving test config to results/acq2106_424/2023.Nov.06.12.53/acq2106_424.None.json
```

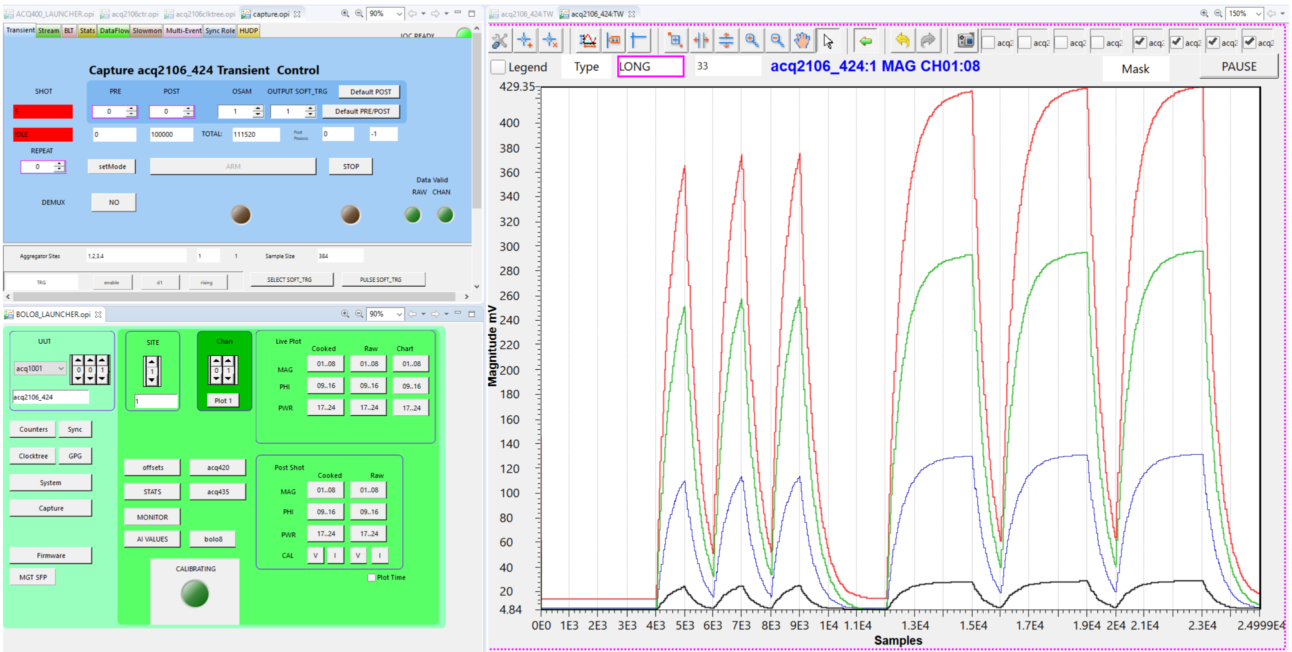


Figure 27: CS Studio screen showing 4 channels post calibration. Offsets calibrated out

Note that all channel offsets have been calibrated out. We assume that the varying fin amplitude may be attributed to imperfect positioning of the LED module above the foils.

E.5.7 Real-Time Power

Once we have run a calibration and characterised the attached foils, we can use some of the figures of merit from the calibration to produce a real-time power plot. This takes our shark fin magnitude trace and converts it to a real-time incident power.

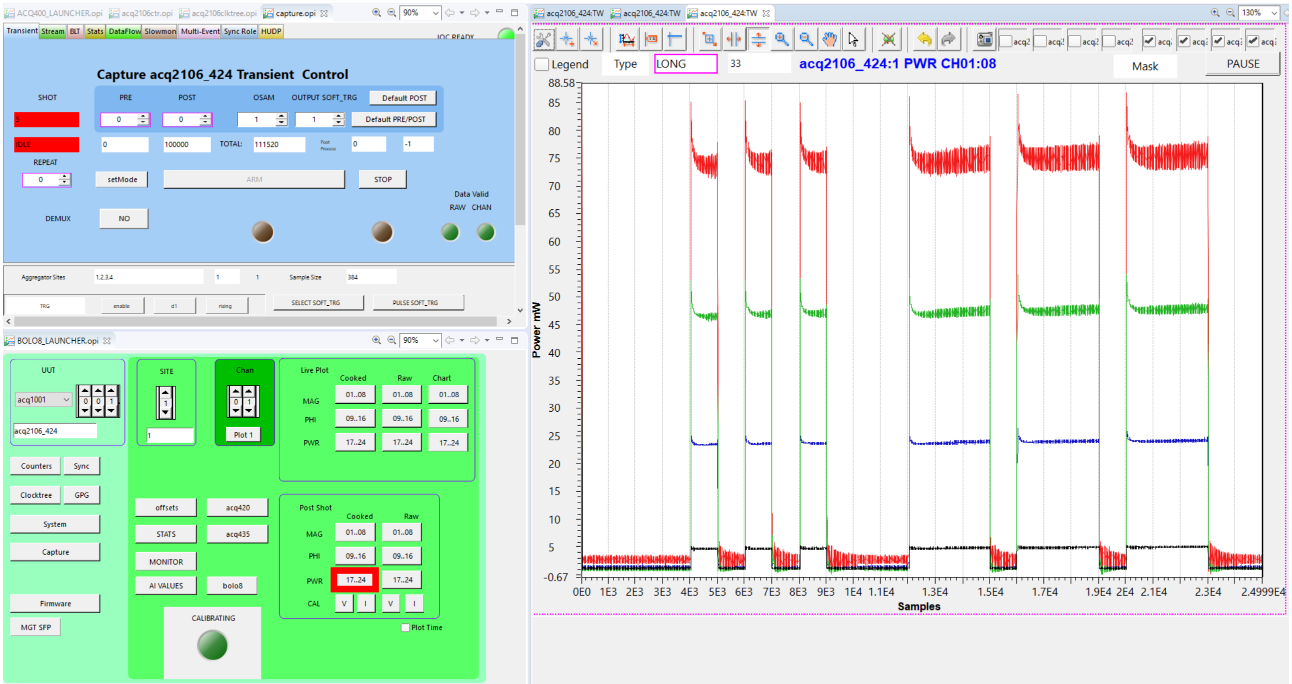


Figure 28: Real-time power CSS plot

The Python script also extracts the data from the UUT and we can plot the power data locally on a host machine.

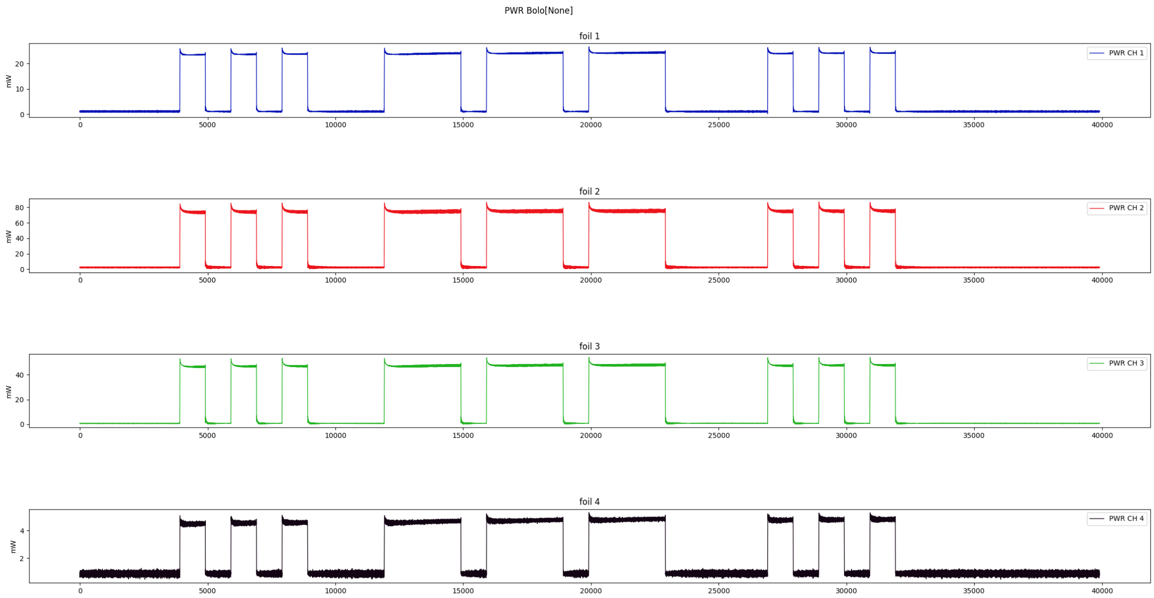


Figure 29: Real-time power data plotted on host

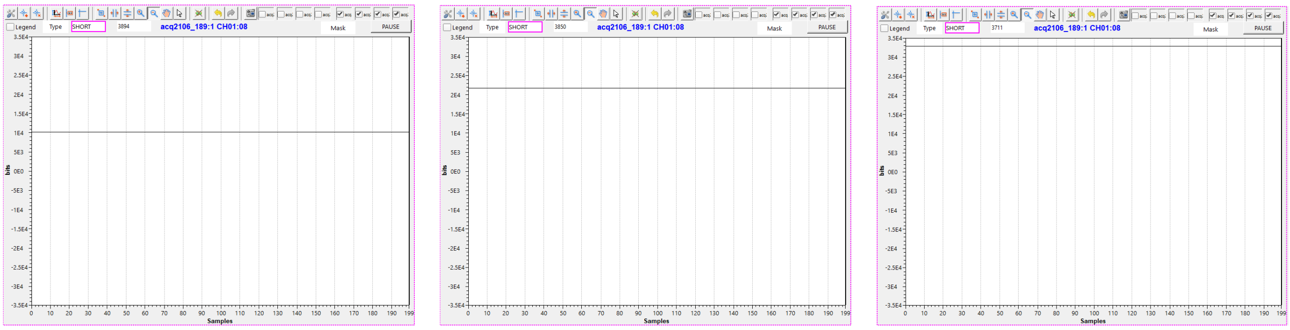
F Streamlined Test Routine for BOLO8 Module

```
Excite Tester
Start the DAC waveform output:
/usr/local/CARE/bolo-pro-test 0
View CH1-8 AI on cs-studio live (or raw) plot.

Exercise the DAC, chaging the waveform amplitudes:
map /usr/local/acq2106.map
md $DSP1+18 1
mm $DSP1+18 0          #  1 Vpp
mm $DSP1+18 1          #  9 Vpp
mm $DSP1+18 2          # 15 Vpp
mm $DSP1+18 3          # 18 Vpp
```

Update the Board Test Sheet “MAIN ADC” Section, Pass/Fail.
Update the Board Test Sheet “MAIN DAC (Including Amplitude Changes)” Section, Pass/Fail.

```
Offset Tester
Use bolo-pro-test with the argument which corresponds to the Offset DAC value in codes:
/usr/local/CARE/bolo-pro-test 0
/usr/local/CARE/bolo-pro-test 1fff    # 1.875 V
/usr/local/CARE/bolo-pro-test 3fff    # 3.75 V
/usr/local/CARE/bolo-pro-test 5fff    # 5.625 V
/usr/local/CARE/bolo-pro-test 7fff    # 7.5 V
View CH1-8 AI on cs-studio raw plot, checking each value in turn.
```



(a) 0x1fff Offset (b) 0x3fff Offset (c) 0x5fff, 0x7fff Saturated Offset

Figure 30: Offset DAC amplitudes displayed in a CSS Raw Plot

Update the Board Test Sheet “OFFSET DAC (TRIM)” Section, Pass/Fail.

```
Current Sense Test
/usr/local/CARE/bolo-pro-test 0
/usr/local/CARE/bolo-pro-test 1fff    # 1.875 V
/usr/local/CARE/bolo-pro-test 3fff    # 3.75 V
/usr/local/CARE/bolo-pro-test 5fff    # 5.625 V
/usr/local/CARE/bolo-pro-test 7fff    # 7.5 V
View CH9-16 AI on cs-studio raw plot and stats.opi, checking each value in turn.
```

DAC Offset Codes	ADC Codes	Stats OPI
0	640	0.194
1fff	685	0.209
3fff	743	0.226
5fff	800	0.244
7fff	858	0.262

Table 5: 12-bit Current ADC Representative Values

Circa 2023 onwards, current ADCs are 12-bit. For old 16-bit values see table in main document.
Update the Board Test Sheet “OFFSET DAC (TRIM)” Section, Pass/Fail.

```
Enable the bolodsp package : 'mv /mnt/packages.opt/*bolodsp* /mnt/packages' and reboot

Calibration - omit --url if you don't want to log the results
[dt100@eigg-fs ~]$ HAPI
[dt100@eigg-fs acq400_hapi]$ pwd
/home/dt100/PROJECTS/ACQ400/HAPI/acq400_hapi

Channel blocks up to the user...

./test_apps/bolo8_tester.py --url=http://eigg/endpoint/bolo UUT --chans=1-48
OR
./test_apps/bolo8_tester.py --url=http://eigg/endpoint/bolo UUT --chans=1-8
./test_apps/bolo8_tester.py --url=http://eigg/endpoint/bolo UUT --chans=9-16
./test_apps/bolo8_tester.py --url=http://eigg/endpoint/bolo UUT --chans=17-24
./test_apps/bolo8_tester.py --url=http://eigg/endpoint/bolo UUT --chans=25-32
./test_apps/bolo8_tester.py --url=http://eigg/endpoint/bolo UUT --chans=33-40
./test_apps/bolo8_tester.py --url=http://eigg/endpoint/bolo UUT --chans=41-48
```

Update the Board Test Sheet “Single Foil Calibration Check” Section, Pass/Fail.

G DSP Block Diagram

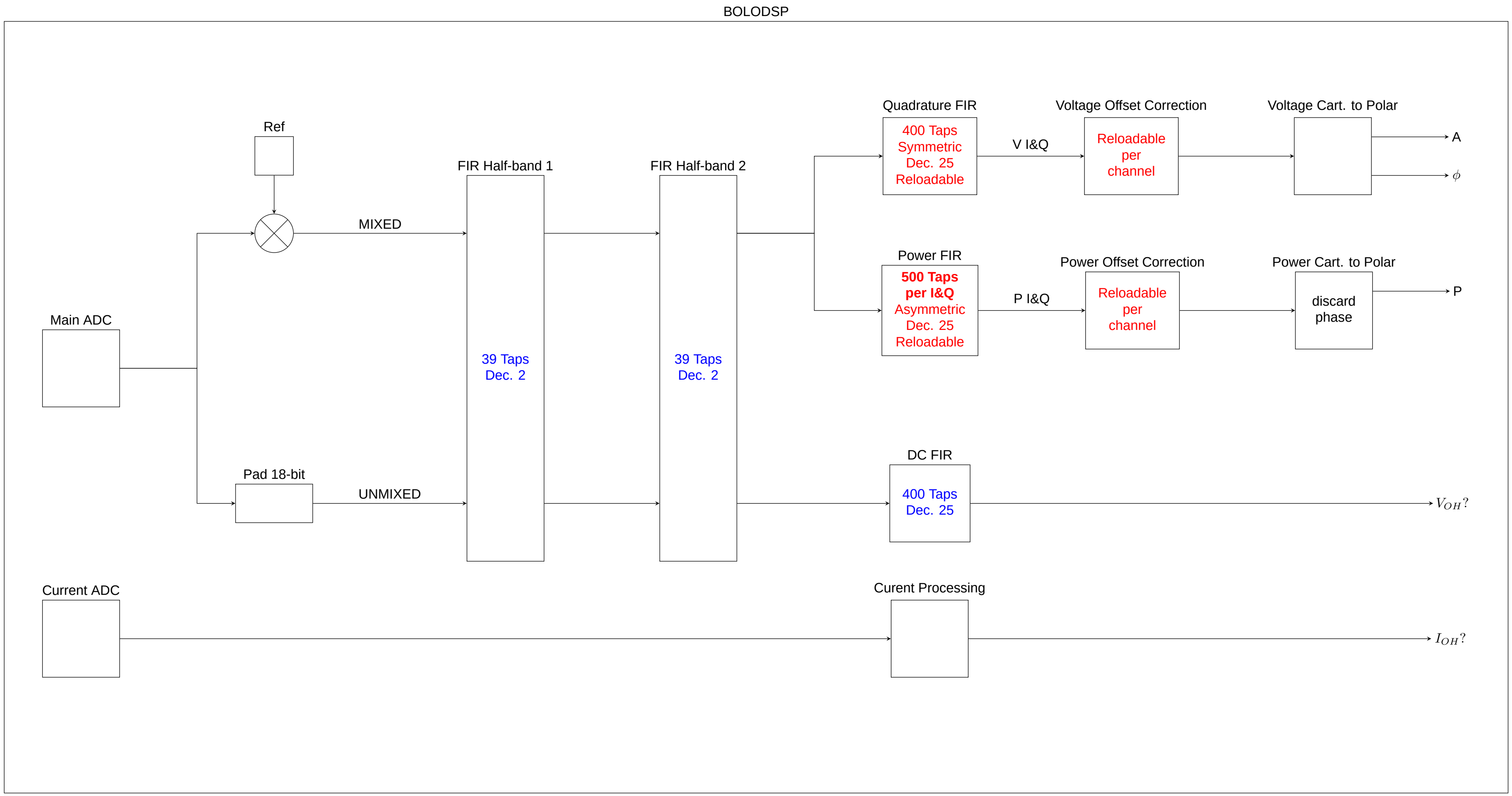


Figure 31: Block diagram showing the inner workings of the BOLODSP Module