

Prex/Crex DAQ Update

3/5/2019

Cameron Clarke

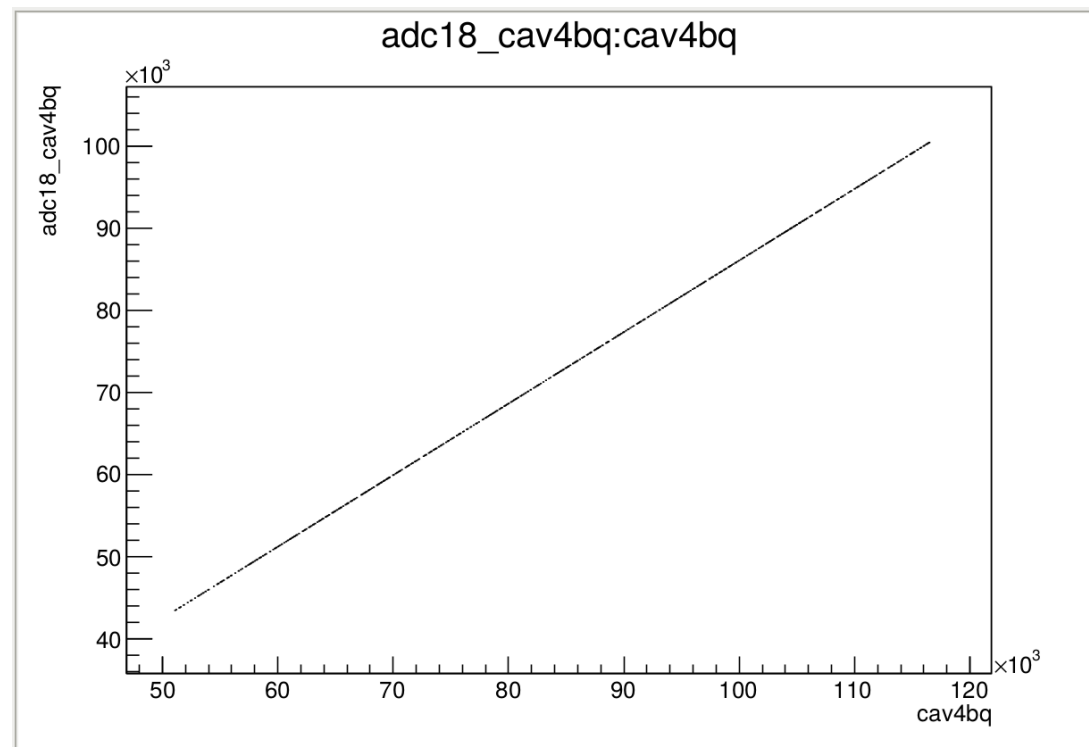
Some recent updates

- Happex ADCs can be decoded in JAPAN
 - Two more QWeak ADCs fixed (Mark Taylor is getting more comfortable repairing)
 - 20 Second sleep at go no longer needed - fix for 64 bit ET as remote tested
-
- Many more updates pending once APEX is done

Happex ADCs in JAPAN

- Wouter and Bob have gotten Happex ADC decoding implemented in JAPAN
 - It works with reasonable voltages: depending on the gain settings the HAPPEX ADCs work from 0 to +5 V at 30 Hz, and generally less
 - Bob and Iris have tested a few HAPPEX ADCs in the TEDf test stand with a standalone decoder
 - Cameron and Chandan previously verified that the currently set gains in vxworks boot scripts are ideal

- Tested in CH crate with random ramping voltage
- Decoding works for ~2 – 5 volt signals
- Error decoding needs to be committed still
- Plugged into both Happex ADC and QWeak ADC →
- So the Happex ADC can match closely and Japan works

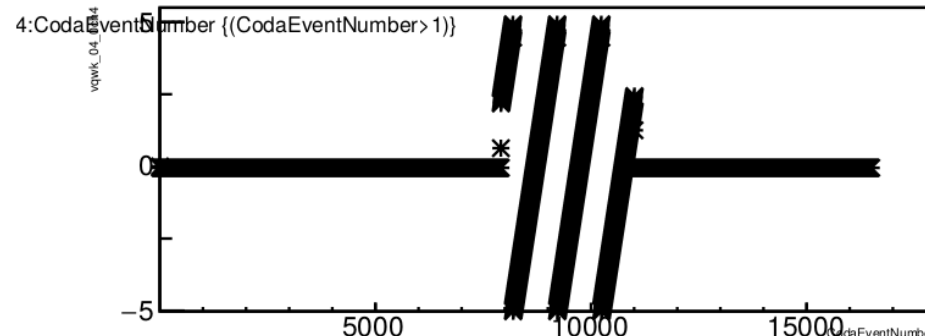
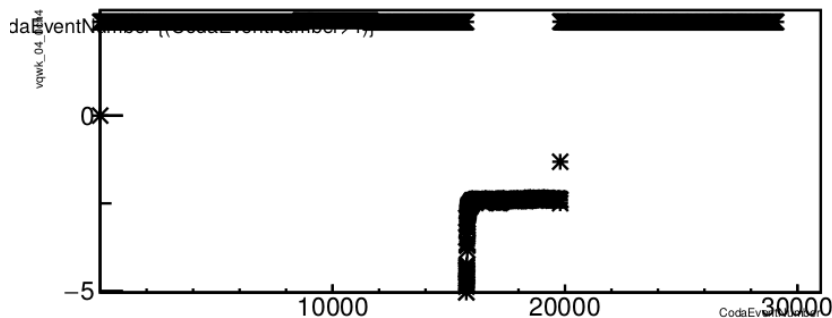


QWeak ADCs in Virginia

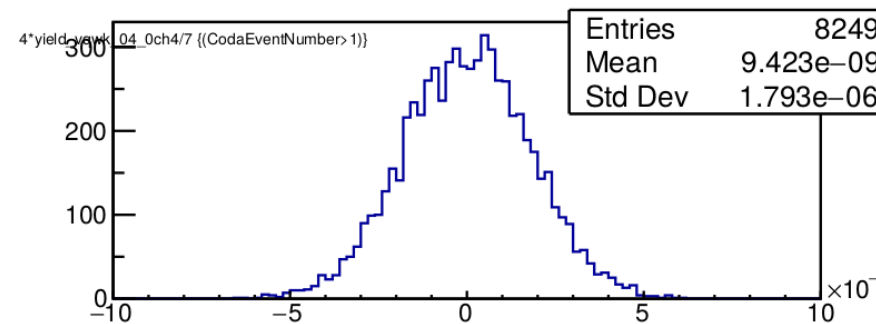
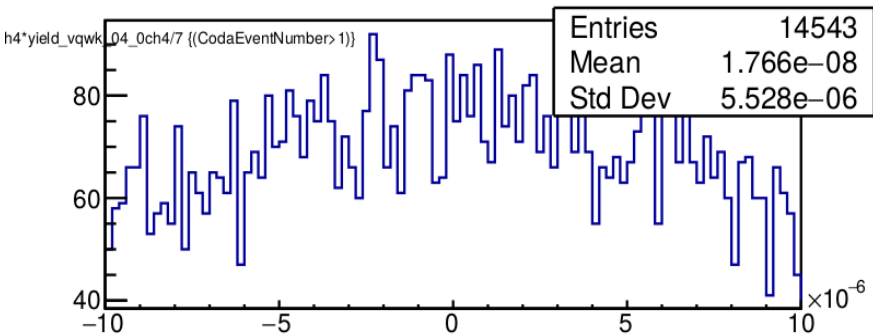
- Mark Taylor has done it again – fixed two more ADCs (at least 6 channels restored)
- New parts have been arrived and further repairs can be done on the remaining two dozen channels

Example plots for ch 4 of ADC 37:

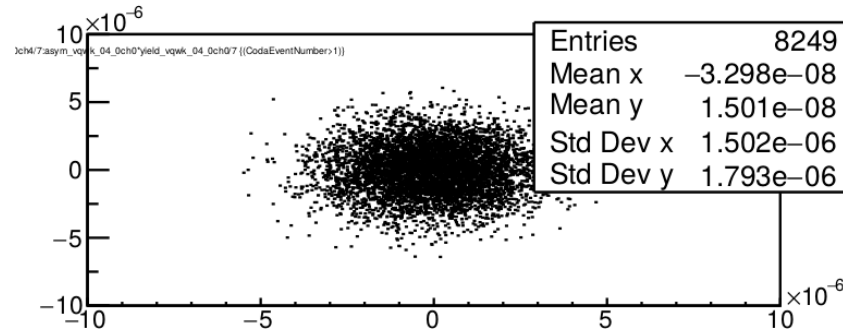
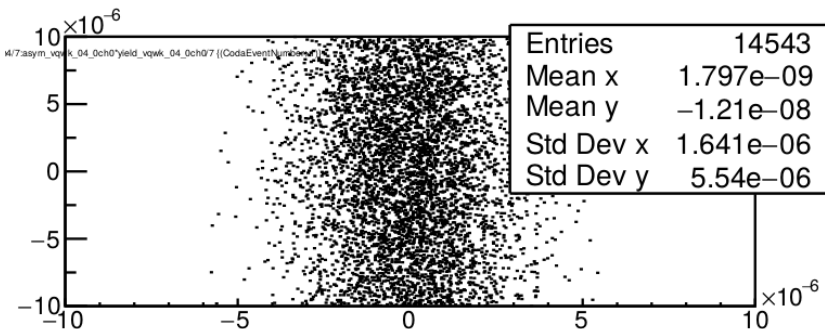
voltage vs. event number



Diff / 7 volts noise floor



Correlation of channel 4 vs. channel 0



← Broken | Fixed →

Sleep for 20 seconds

- CODA 2.6.2 had a bug which we caught and got a fix from Carl Timmer
- Previously wouldn't let you treat a local instance of ET system as remote and ETbridge segfaults
- A few lines of code updated and it now waits for the local-as-remote ET instead of giving up
- Now we can run the Etbridge from “startcoda” rather than at each run's “start”, and 20 second sleep is no longer needed as well
- Running ET systems manually with monitors independently of Rcgiui:
- Ready to implement any day

```
adaq@atedf3:~/cameronc
input list: cnt = 100000, events in = 0
output list: cnt = 0, events out = 0

LOCAL USERS:
  process #0, # attachments(1), attach ids(0,), pid(4283), hbeat(84)

ATTACHMENTS:
  att #0, is at station(GRAND_CENTRAL) on host(atedf3.jlab.org)
    at pid(4283) from address()
    proc(0), blocked(NO)
    events: make(0), get(0), put(0), dump(0)

EVENTS OWNED BY:
  system (100000), att0 (0),

EVENT RATE of GC = 0 events/sec

CREATING STATIONS:
IDLE STATIONS:
STATION CHAIN:      GRAND_CENTRAL,
LOCKED MUTEXES:

*****

```

```
adaq@atedf3:~
attachments: total#(1), ids(0,)
input list: cnt = 0, events in = 0
output list: cnt = 0, events out = 0

LOCAL USERS:

ATTACHMENTS:
  att #0, is at station(TransferOut) on host(atedf3.jlab.org)
    at pid(4283) from address(129.57.37.107)
    proc(-1), blocked(YES)
    events: make(0), get(0), put(0), dump(0)

EVENTS OWNED BY:
  system (1000), att0 (0),

EVENT RATE of GC = 0 events/sec

CREATING STATIONS:
IDLE STATIONS:
STATION CHAIN:      GRAND_CENTRAL, TransferOut,
LOCKED MUTEXES:

*****

```

```
adaq@atedf3:~/cameronc
AAAA
(a) TCP port = 4444
(out) open et = /tmp/et_sys_par2
et_station_create (in) 15245824 TransferOut
att 1111
att 2222
att 3333
Call et_events_bridge, num events = 10000

```

```
adaq@atedf3:~/cameronc
# et-12.0 version of et_start compiled with 64-bit
#~/my_coda_dir/Linux-x86_64/bin/et_start -p $PORT -n 3000 -s 15000 &
~/cameronc/et_start_64 -p $PORT -n 100000 -s 15000 &
[adaq@atedf3 ~/cameronc]$ setenv SESSION par2
[adaq@atedf3 ~/cameronc]$ ./et
et-12.0/ etbridge* et_monitor_64* et_start_32* et_start_64*
[adaq@atedf3 ~/cameronc]$ ./et_s
et_start_32* et_start_64*
[adaq@atedf3 ~/cameronc]$ ./et_start_64 -p 4444 -n 100000 -s 15000

```