Validation

Contents

1 Introduction

This section intent to validate the software production.

2 Bugs to solve

- DIF does'nt run RNG test a second time (after a soft reset).
- Sending configuration sometime loose the last word and push it into the next configuration.
- Reading data show empty chips lines.
- Reading data show strange BCids.

3 ASU configuration

We have to validate a configuration file extracted from the */data/DB/ch_28_26_37_32_36.sqlite* database available on all machine in Lyon. This database is browsable using the **Sqlite manager** Firerfox's plugin. Here we have such a file: *acqilc@lyoac26:/data/online/config/sc_lyon_1.txt*. Gains are about:

- B0: 190
- B1: 300
- b2: 600

4 Read-out chip's data

Tested on Scientific Linux 5.5:

- With one LDA and manual CCC

```
# ./exemple polcaldaq.conf -i1000 -t 100 -d 500000 -m4 -S2 -U10
```

- With one LDA and automatic CCC

```
# tools/utDevice polcaldaq.conf -i1000 -t 100 -d 500000 -m4 -S2 -U10
```

- With two LDA and automatic CCC

```
# tools/utDaq -i1000 -t 100 -d 500000 -m4 -S2 -U10
```

Note that we can capture the ethernet flow and re-inject it slower:

- Capture raw packets

```
# run/config polcaldaq.conf -i1000 -t 100 -d 500000 -m4 -S2 -U10
# tcpdump -xx -s 1024 -i eth0 ether host 5e:70:0c:d2:5c:d6 -w dump.raw
... ^C
```

- Run the libLDA without configuration:

```
# ./exemple polcaldaq.conf -p eth0 -n
```

- Re-inject the packets: **Note** that eth0 cable must be plugged to a switch, but the LDA must be unplugged to that switch. The `-Z` parameter is the mac address of the host which run the libLDA.

```
# ./pcat -I eth0 -M 5e:70:0c:d2:5c:d6 -Z 5c:26:0a:28:73:0a -m replay -i dump.raw
```