Discussion on DAQ: my prospects

Taikan Suehara
(Kyushu University)

Our target as a collaboration

1. Realize ILC
2. Realize and optimize ILD detector
3. Develop and optimize ILD SiECAL

CALICOES and modq

CALICOES

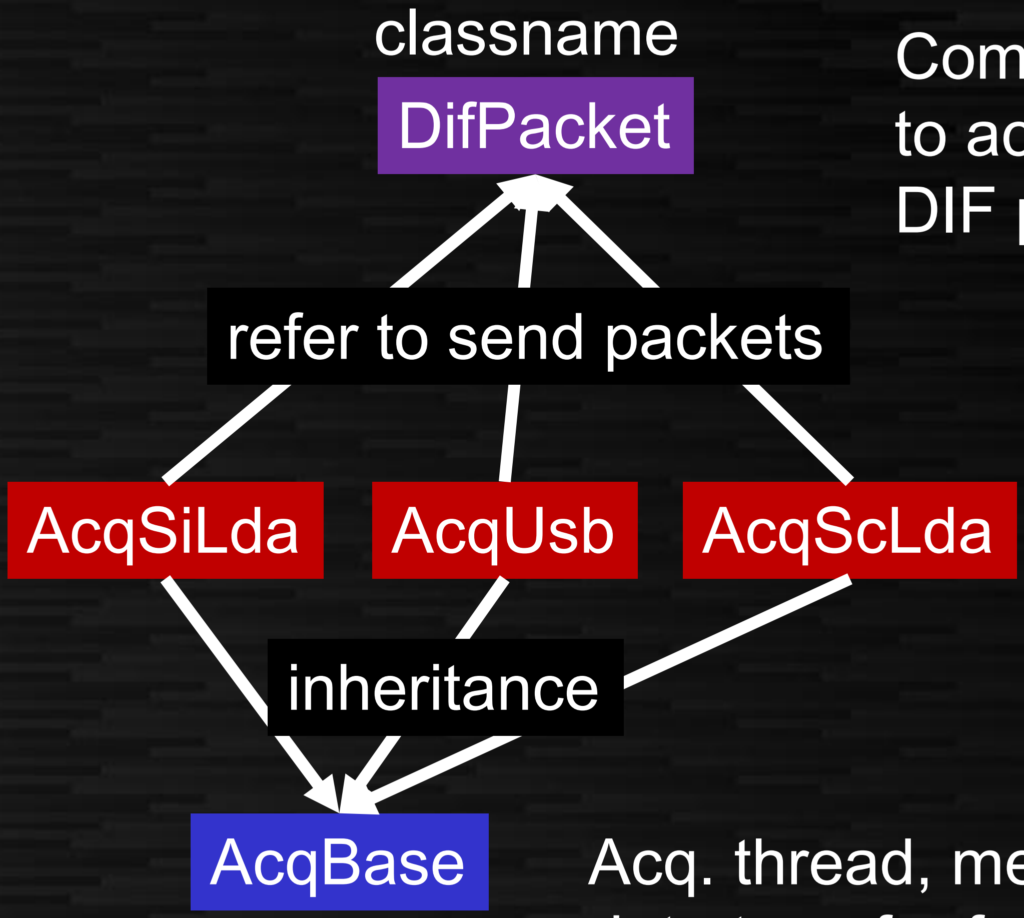
- Full DAQ
- For test benches (not for ILC DAQ)
- Not optimal for Si/Sc code sharing
(at least not optimal if I will develop)

modq

- not full DAQ, just a readout module
- For a component of ILC DAQ
(carefully designed for large-scale application)
- Designed for maximal code sharing of all ROC DAQs

modq ROC code sharing

Developing Sc only or both Si/Sc is NOT SO DIFFERENT

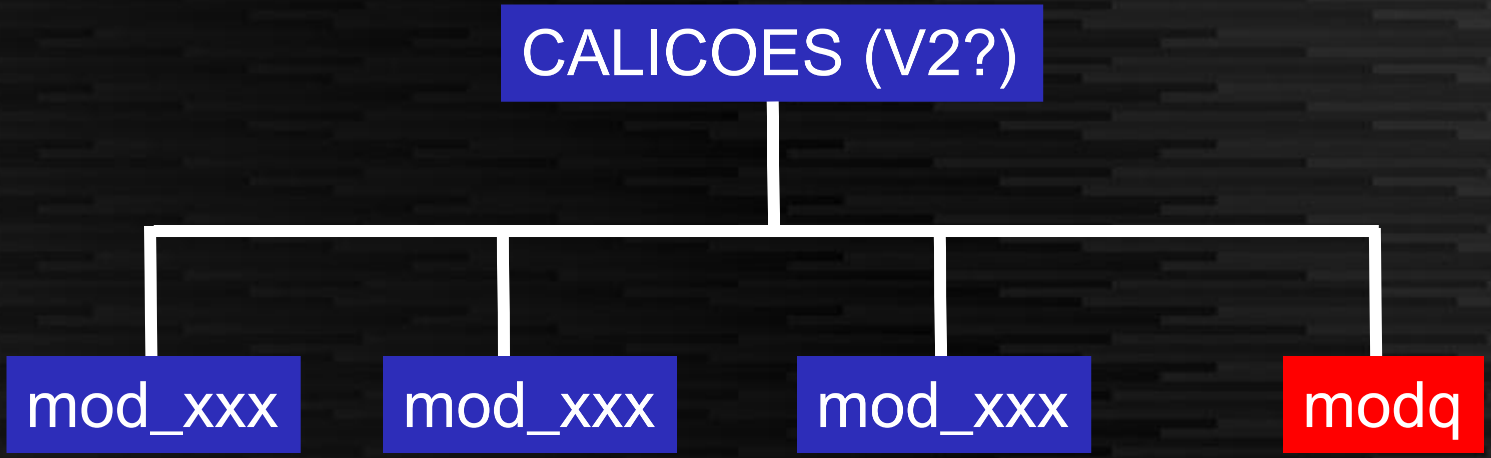


Common packet structure to accommodate/serialize DIF packets (fast, BT, data)

minimal specific header/ device specific control (socket creation etc.) implemented here

Acq. thread, message transfer from outside, data transfer from generic devices

My proposal



- From users (physicists), modq can be seen exactly as one of CALICOES module
  (shell script and xml communication will be available)
- You can select mod_roc/dif/lda or modq (or both) as primary manipulator of packets (I will provide method)
- You can use modq just as a debugging tool or using full feature of readout
- You can use all features of CALICOES as before

Advantage of modq

- Do not need to wait you fixing the CALICOES code (independent code management)
- Adding features is much much easier for me since it's C++ (I have 20 years of exp.)
- You can contribute as well as use (it's open) https://github.com/suehara/modq (very temporal version)

Our timeline

- 14th July, TB meeting with DESY
  - we should propose for the next TB at Oct./Nov.
  - So, we need something for ScCAL by then
- end of July, CALICOES test campaign
  - If you provide core software before then, I will implement the adaptor and participate remotely, if you like
- August, you publish CALICOES v2
  - We will adapt as soon as possible
  - We will publish modq here
- September, finalize first version of modq

modq is NOT all of our contribution

- User interface (online/offline monitor)
  - Independent of modq (and CALICOES, I think)
  - based on data sent from a file or via network
  - maybe, ROOT based (if we will be the core)

- Test ongoing with current CALICOES
  - injection (now), cosmic
  - RI (Sr90 etc.)
- Testboard
- Software of test of a lot of sensors/DAQs

Final: comments for CALICOES

- Logging – essential
  - We should keep all logs (at some verbose level) to see what is going on (esp. in beam tests)
- Repository
  - mercurial? (git? subversion?)
- I propose to get rid of C…
  - Buffer overflow (I found one in xml parser)

If you need to check my ability,

LCFIPlus (flavor tagging package)

https://svnsrv.desy.de/viewvc/marlinreco/LCFIPlus/

60-70% of the code was written by me mainly in several months

Spectroscopy software (commercial company) including inline production control (in windows...)

Accelerator cavity aging system

etc.