DAQ Software Development

Giovanni marsella, **Italo Epicoco**, Fabio Convenga and Roberto Assiro

SDEU Meeting
Paris, France - December 13-14, 2018

University of Salento and
National Institute of Nuclear Phisics - INFN

Objectives and Motivations

The main motivations that drive the new DAQ developments are:

- Improve hardware resource utilization: interrupts vs polling.
- Improve software resilience and fault tolerance.
- Enhance the data logging.
- Enrich the remote monitoring.
- Enhance source code maintainability, documentation and versioning.

/dev/mem vs UIO

The access to device memory was handled by using the */dev/mem* which has the following characteristics:

- it can access all device memory address space at the same time;
- it has no access limits to different registers and is, therefore, subject to system crashes due to coding errors;
- data acquisition is carried out using the polling technique, since */dev/mem* is not aware of interrupts.

The UIO interface drivers introduces the following characteristics:

- the UIO allows direct and preferential access to memory;
- each concerned register (eg trigger, memory shower etc.) has its UIO port;
- the muon and shower trigger registers have a dedicated UIO interrupt enabled.

The use of interrupts reduces the CPU utilization and hence the energy consumption.

Fault Tolerance

- DAQ follows a multiprocess software approach
  - Inter Process Communication (IPC) is based on shared memory
  - One process controls all the others
  - Signals are mainly used by "Control" to start / stop other processes
- To improve fault tolerance
  - Deeply debug each single DAQ process to avoid uncontrolled crashes
  - In case of crash the station, switch in "stage 0" state. Only Control, MsgSrv_in and MsgSrv_out must be active
- Estimated effort: 6 Person-Months

Thread based approach

- Some of the processes are strictly correlated each other with frequent data exchanges through shared memory
  - E.g. FeShwr, Muonfill, Trigger2
  - E.g. Control, MsgSrv_in, MsgSrv_out
  - ...
- Gathering those processes in one single multithread process should bring the following benefit:
  - efficient synchronization among threads,
  - faster access to shared memory
  - safer management of semaphores and critical regions
- Estimated effort: 6 Person-Months

Data Logging

- Enhance the data logging system. The idea is to use the new logging system only for the Engineering Array.
  - Collecting each significative status changes for each DAQ process in the permanent filesystem (i.e. flash or USB)
  - Each process stores its log file in distinct directories
  - Keep track of the last month activities: automatic log rotation discards old files
  - Three levels of severity: Info, Warning, Error
- Offline data analytics of log files
  - Number of failures in last period
  - Acquisition rate
  - ...
- Effort: 1 Person-Month

Online monitoring system

- Enrich the current monitoring data sent to CDAS with:
  - status of the hardware, the status of the software and eventually some pre-alarms.
- An enriched monitoring system can be very useful for the engineering array. So our idea is to implement the new monitoring system only on the EA. After the EA experience, we can re-discuss if the monitoring could eventually be applied to the rest of the array
- Effort: 4 Person-Months

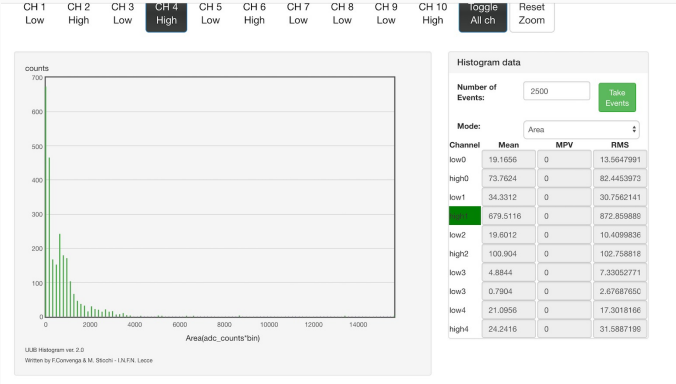
Software maintainability

- The process of software validation and test could be improved
  - Develop **off-line** test cases for an automatic testing on the development environment (outside the board). [alpha test]
  - Develop **on-line** test cases for testing the software on board in the laboratory [beta test]
  - Develop **on-field** test cases for testing the software on a tank in the field
    - The software which passes the *on-field* tests is considered the release candidate
- Enhance software documentation using Doxygen
- Report the software version number on request
  - DAQ, Petalinux, FPGA bitstream
- Effort: 12 Person-Months

Web complementary tools development

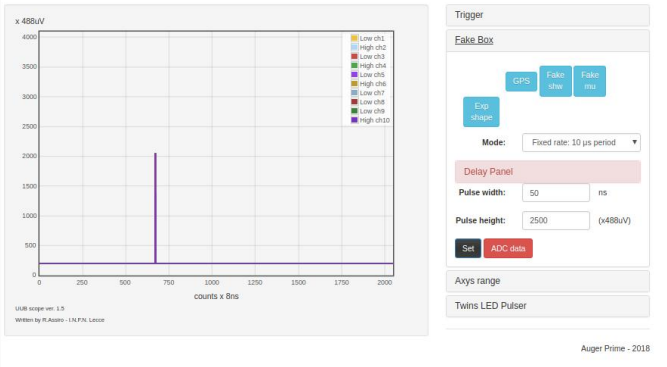
Web Tools

New web tools have been developed and integrated in the web platform designed by Roberto Assiro:



SSD test web tool which display peaks and area distribution using the values reported in the FPGA's registers

Fake data web tool for setting the parameters related to the fake data generation



A web page for monitoring the GPS

Thanks

Backup slides

Flash memory endurance

- Typical SLC/MLC flash memory endurance rate: 100.000 erases/bock
  - 1 block = 16KB
- Estimated data log writing rate: 16MB/day
  - 1024 blocks written per day (i.e. 1024 erases per day)
- Flash memory size: 128MB (8192 blocks)
- Total number of erases allowed before failure:
  $100.000 * 8192 = 819.200.000$
- Number of days before failure: 819.200.000 / 1024 = 800.000 days