



# Calicoes

Calice OnlinE System  
Frédéric Magniette



AIDA



# The Needs

- Acquisition Chain :  
developpement from scratch
- Control-Command Chain  
(replacement for buggy Liblda)
- Human-Machine interfaces
  - For DAQ debugging
  - For data acquisition
- Properties
  - Calice compatible (DIF-LDA-Ethernet Calice format – can use different Omega Rocs)



# Design Concepts

- **Internal Modularity**
  - Functional blocks with easy communication channels (on the Unix model)
  - Every block is making few but making it fast and good
  - Easy evolutivity
- **Parallelism**
  - Massively multi-threads approach
  - Pipelined treatments



# Acquisition Chain

Global  
BD

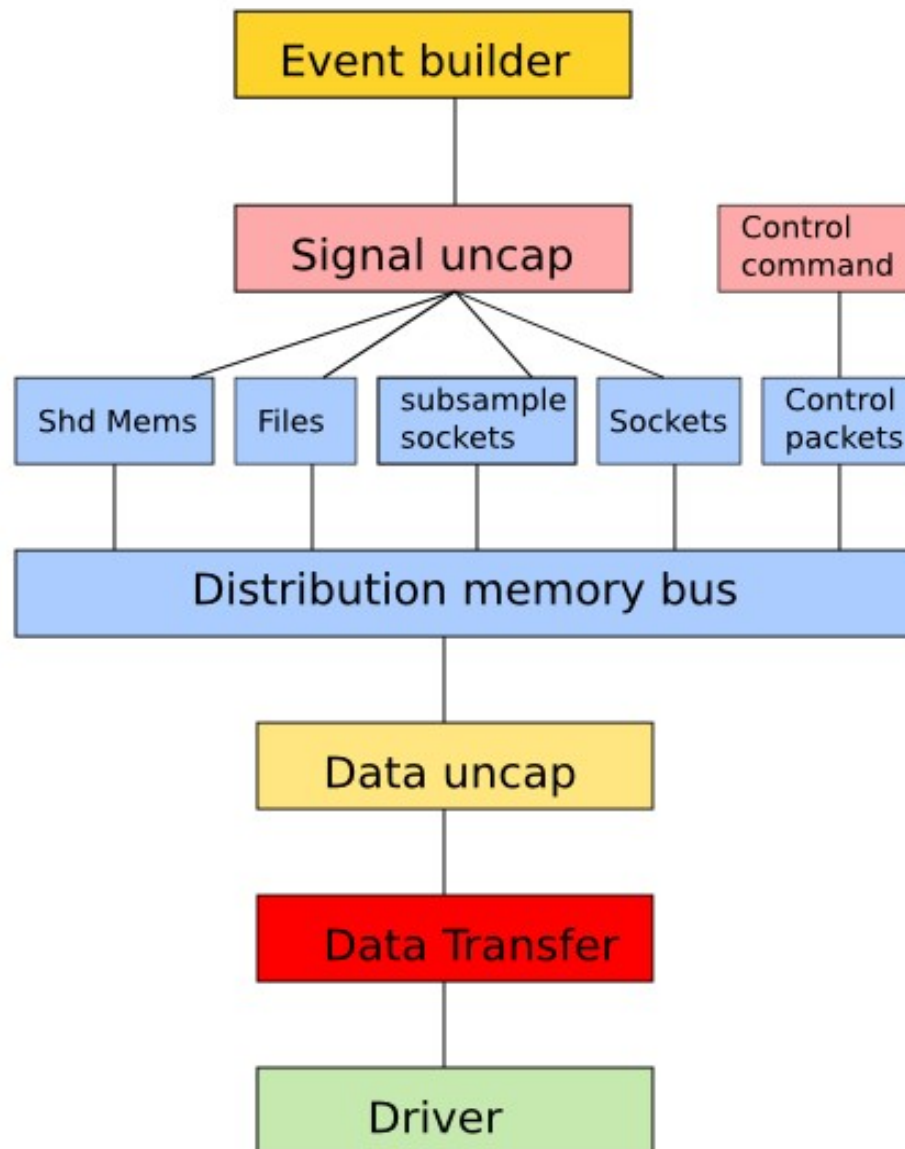
Skiroc

Generic

LDA/  
GCC/  
DIF

Raw  
socket

Ethernet  
Usb  
PCI(e)



building  
events

(de)coding  
ASIC signal  
and control

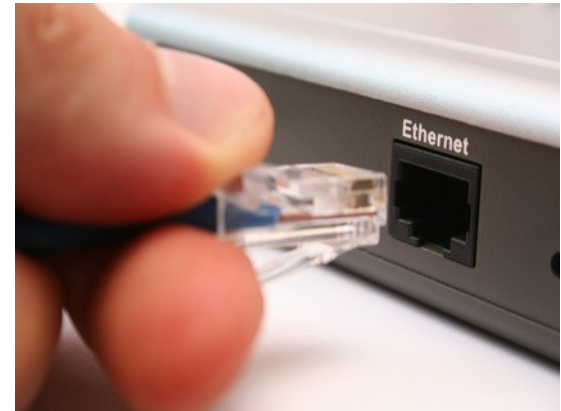
Offering  
various  
distribution  
mode

Checking  
Uncapping  
Reordering  
Tagging

Transferring  
from/to media

Interfacing  
kernel and  
hardware

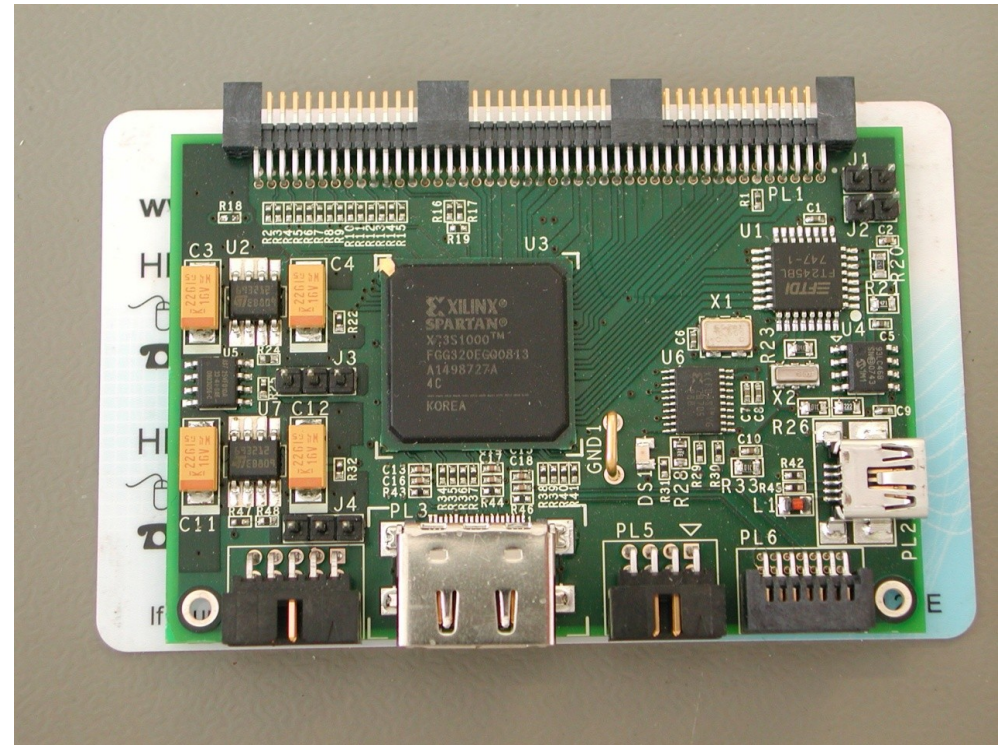
# Low level Transfer



- Standard Ethernet driver (e1000)
- Low-level socket (SOCK\_RAW)
- Limited in rate by the SOCK\_RAW implementation
- Projects to speed up: integrating UDP in GDCC



# LDA/DIF Readout uncap

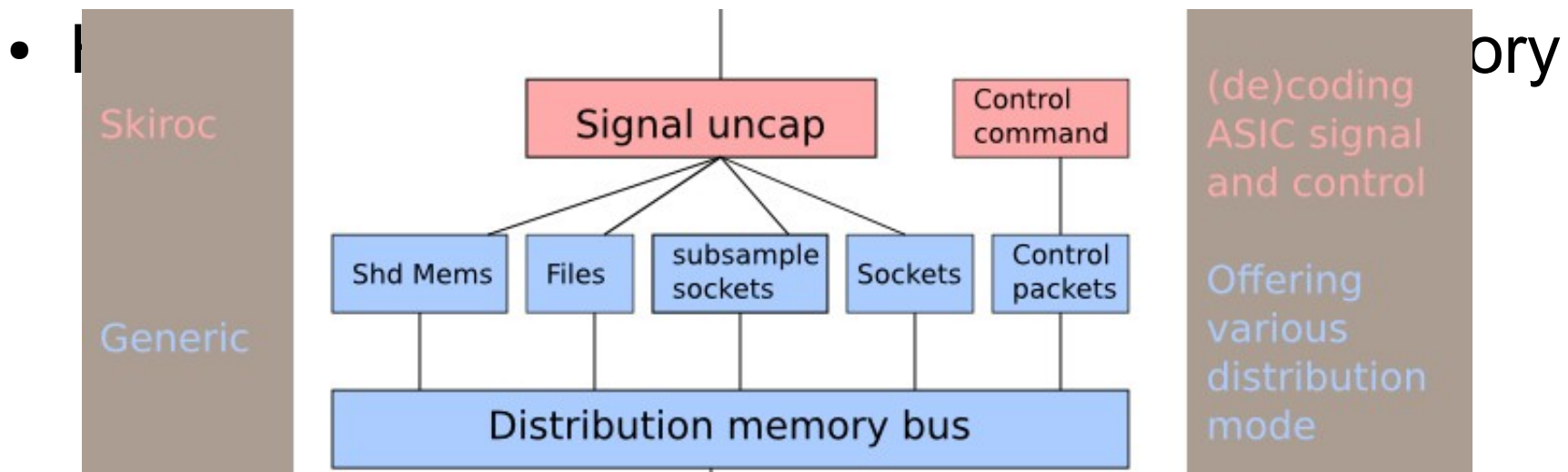


- One of the two specific parts of the system (dependant of the DIF Firmware version)
- Functionalities : uncapping packets, DIF tagging, loss control, integrity control
- Implemented in an external library for genericity and maintainability

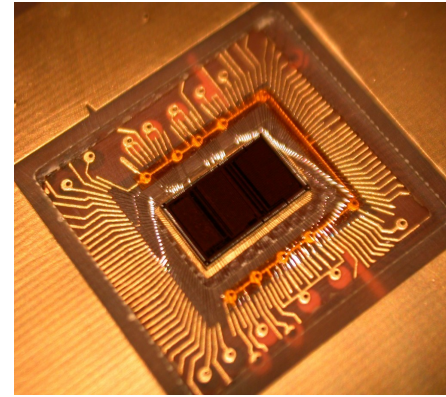
# The distribution bus



- Split data into beams (by dif)
- Offer the maximum flexibility in data treatment
  - Offline through files
  - Remote online through sockets
  - Subsampling sockets for slow online analysis programs



# Signal Uncap



- The second specific block of the system (dependant on the roc chip version)
- Actually adapted for skiroc, can be ported easily to spiroc or any other Omega chip
- Uncaps the skiroc data to produce isolated physical events
- Output a plain text format (without loss of information)



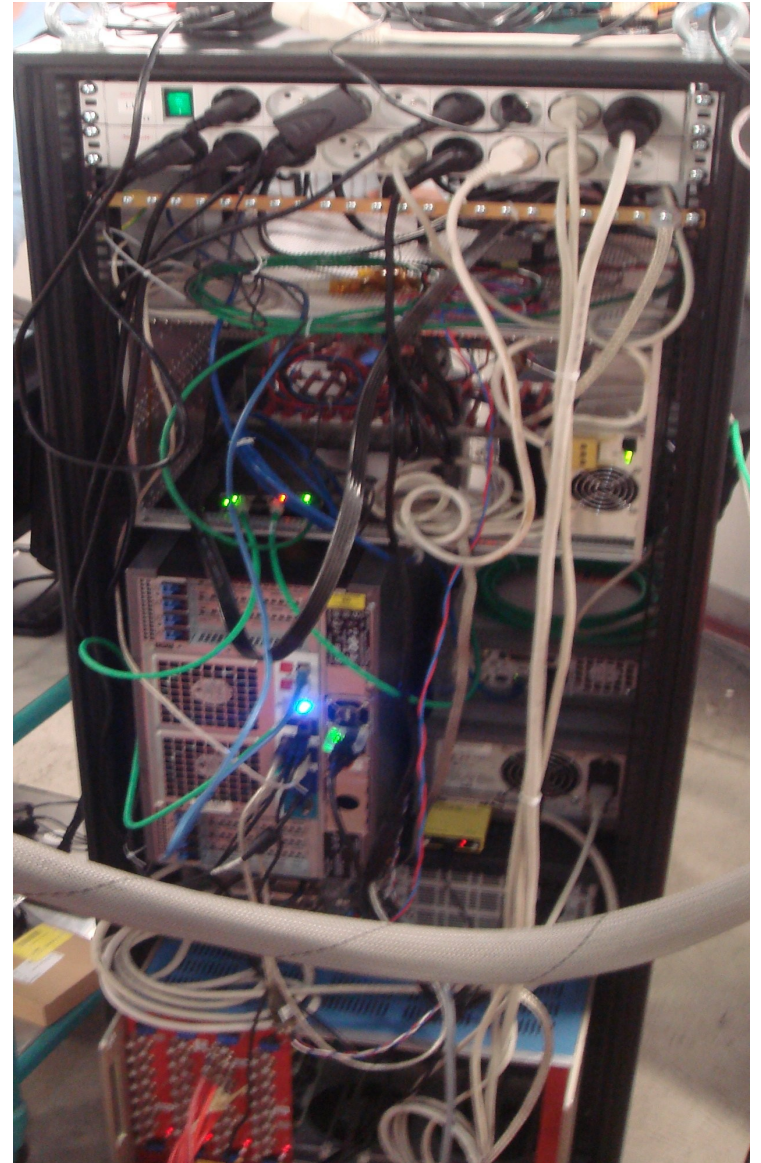
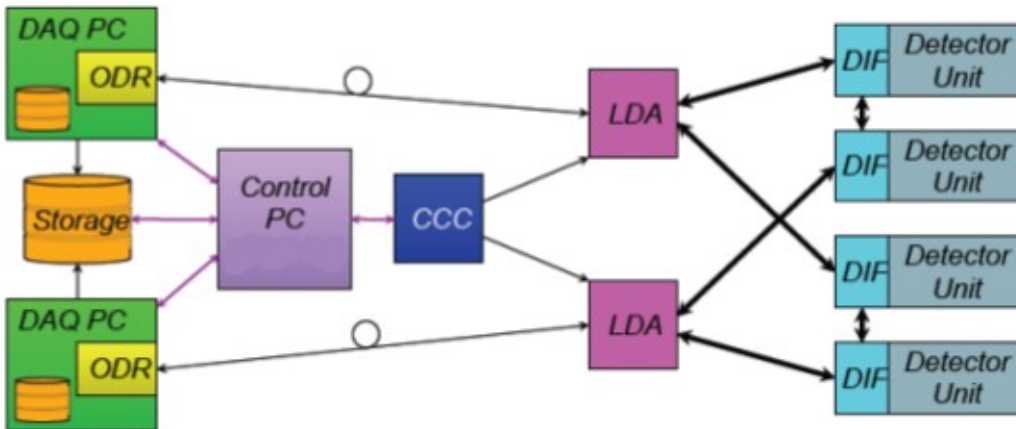
# The Control-Command chain

- All devices has their module in the control-command
- Based on the pyrame framework (LLR)
- All accessible through the pyrame GUI

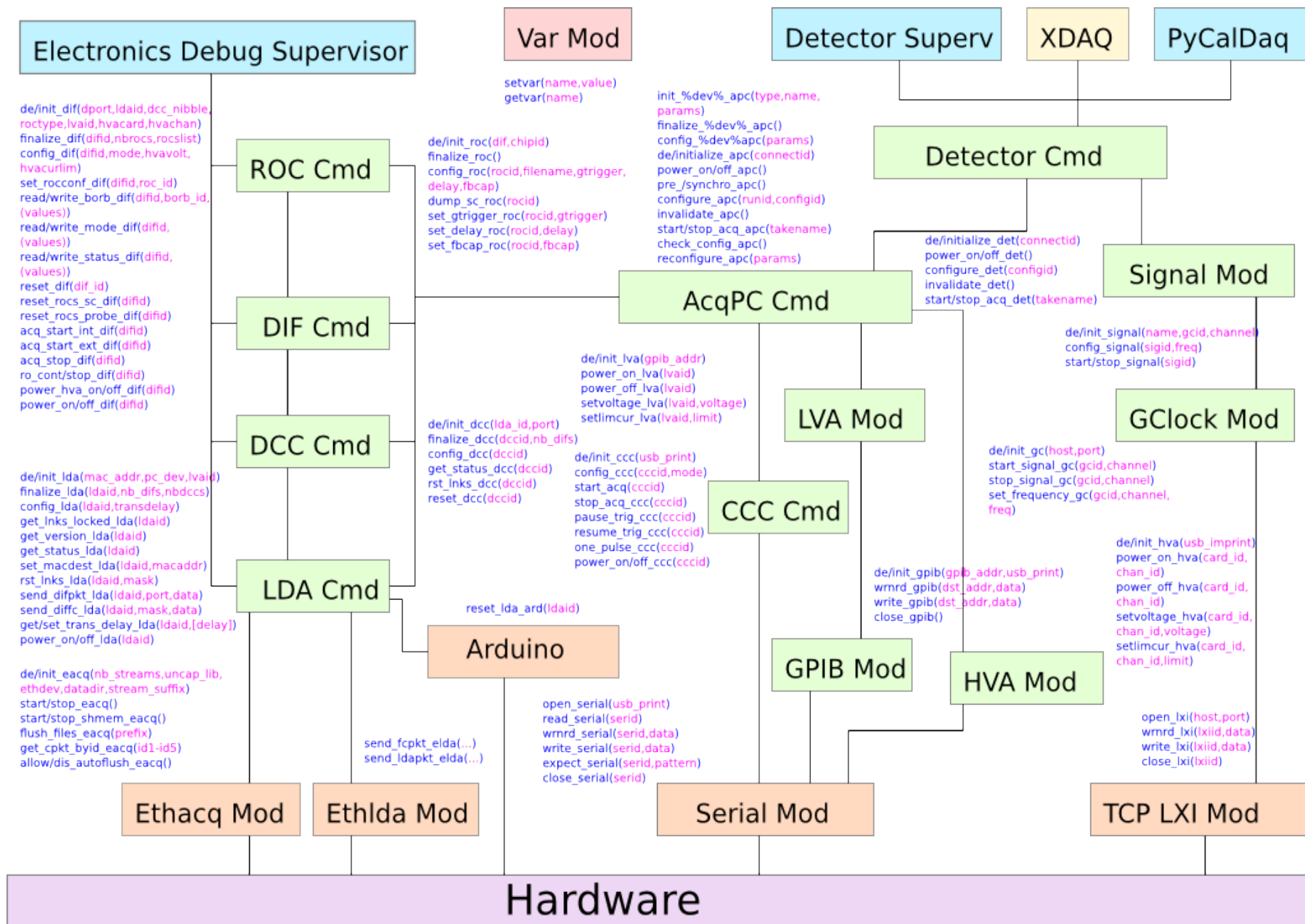


# The actual ECAL hardware

Hardware	Comm Bus
Pulse generator	Ethernet LXI/TCP
High Voltage power supply	RS232
Low Voltage power supplies	GPIB
LDA Reset device	Arduino
CCC	RS232/USB
LDA/DIF	Ethernet



# The Big Picture



# The detector control command

- Detector Abstraction Layer
- Allows only the global detector transitions with integrity checks
- Commands all the other blocks
- Based on a state machine

The screenshot shows the 'supervisor' window with the following sections:

- Calice Local Supervisor**: A table of variables.
- Commands**: A list of control actions with input fields and buttons.
- Connection**: A table for host and port settings.
- Errors**: A section for error messages.

Variables		
state	unknown	
Duty cycle	0	%
Acquisition active	0	
Acquired data	0	packets
Acquired control	0	packets
Unknown data	0	packets
Lost data	0	packets
Number of busy buffers	0	
Data written on disk	0	bytes
Shared memory export active	0	
Data transmitted to shared memory	0	bytes
Data transmitted on sockets	0	bytes
Files autoflush	0	

Commands		
Initialize	<input type="text"/>	initialize_det
Deinitialize		deinitialize_det
Power on		power_on_det
Power off		power_off_det
Configure	<input type="text"/>	configure_det
Check configuration	<input type="text"/>	check_config_det
Invalidate configuration		invalidate_det
Start Acquisition		start_acq_det
Stop acquisition		stop_acq_det

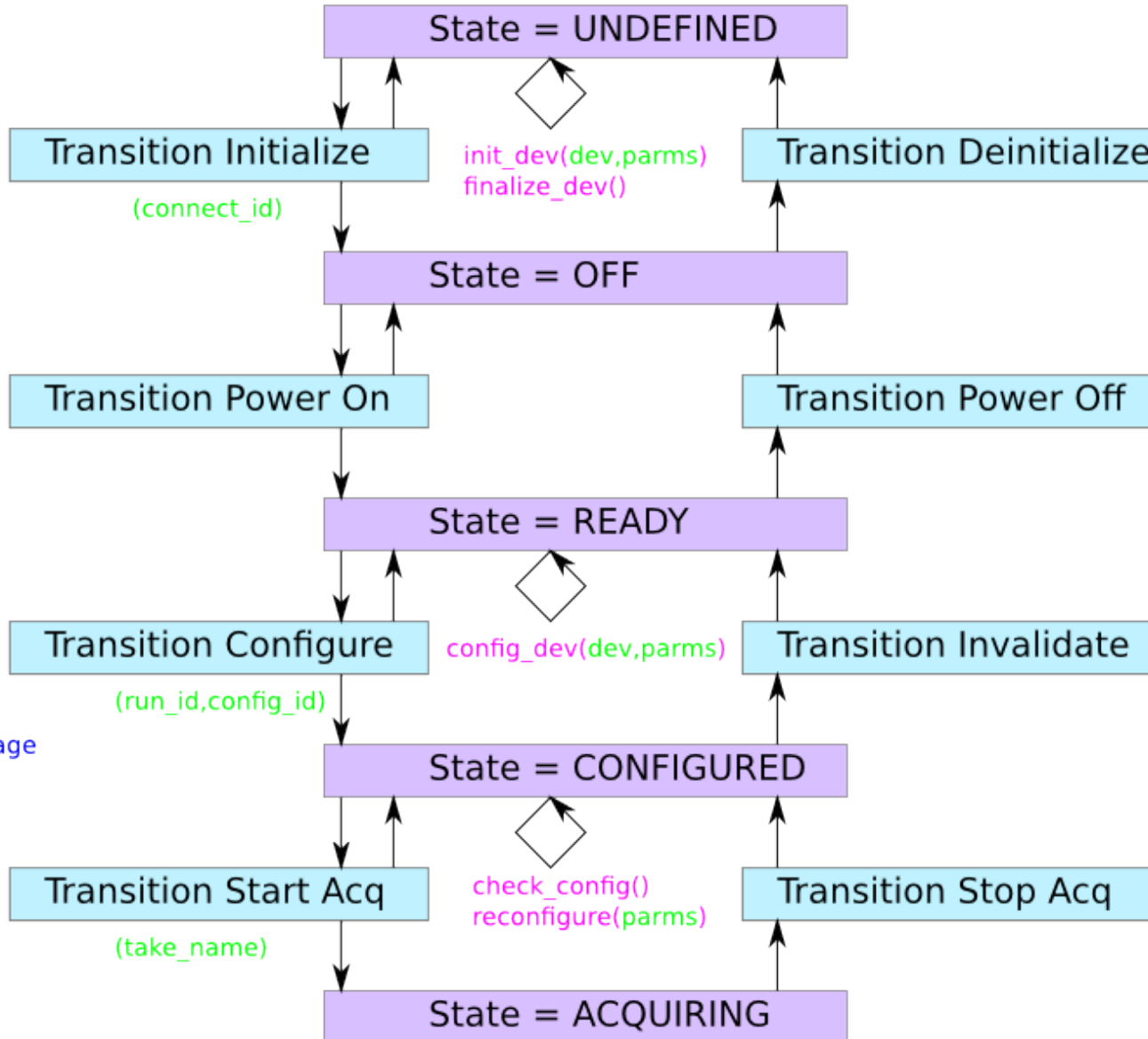
Connection		
Host (disconnected)	Port	Connect
10.220.0.2	9000	Disconnect

Errors

# ECAL state machine

Legend :  
 states  
 transitions  
 parameters  
 cyclic transitions  
 (state internal functions)  
 actions

- Load connectivity Database
- Allocate memory structures



- Power on ccc
- Power on Idas
- Power on difs
- Power on arduino

- Destroy memory structures

- Power off ccc
- Power off Idas
- Power off difs
- Power off arduino

- Reset devices
- Load configuration Database
- Configure devices
- Synchronize clocks
- Zero the chips
- Power on high voltage

- Power off high voltage
- close connections

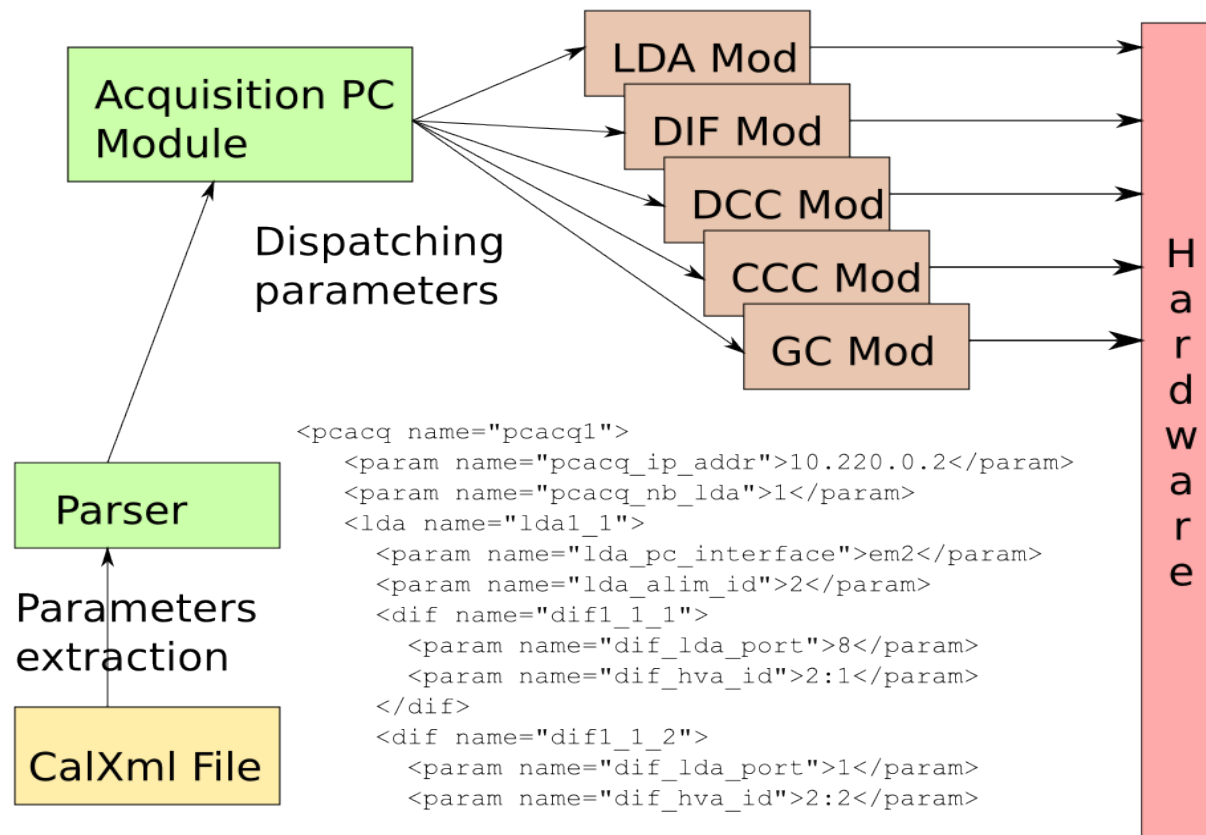
- Start ethacq
- Start spill
- Start trigger
- start cccacq

- Stop cccacq
- Stop trigger
- Stop spill
- Stop ethacq
- Flush files



# Unified XML configuration format

Parser sends « init » and « config » orders to every modules



# Pycaldaq : a scripting interface

- Embedded Python scripts to pilot the whole detector
- Programmed or interactive sessions
- Very useful for calibration



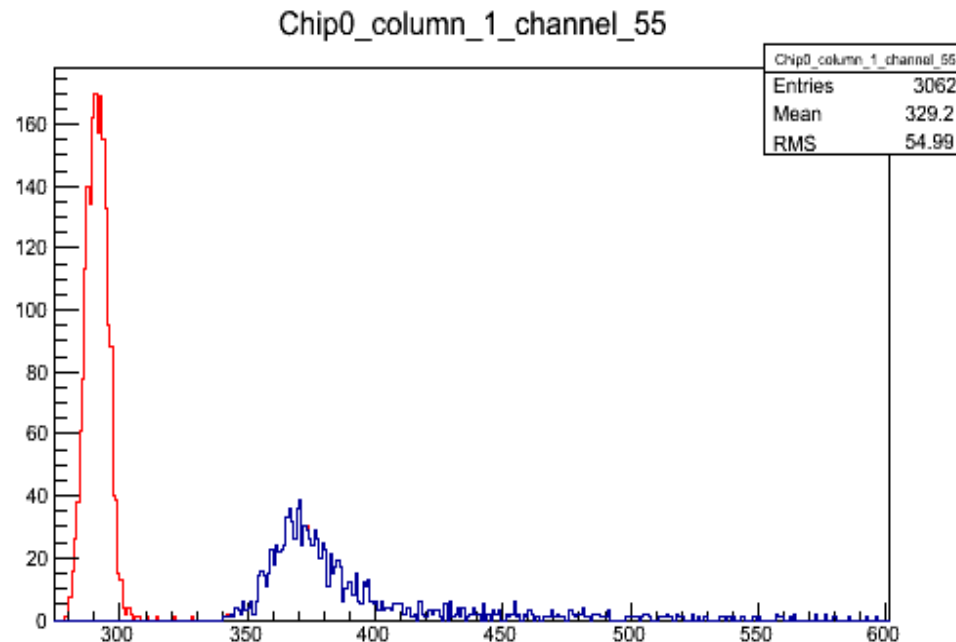
```
import caldaq

def acq_run():
    caldaq.start_acq()
    caldaq.start_spill()
    caldaq.start_trigger()
    time.sleep(15)
    caldaq.stop_trigger()
    caldaq.stop_spill()
    caldaq.stop_acq()

caldaq.flush_files("trash")
for trig in range(150,401,15):
    caldaq.set_gtrigger(trig)
    acq_run()
    caldaq.flush_files("calib_trig%d" % (trig))
```

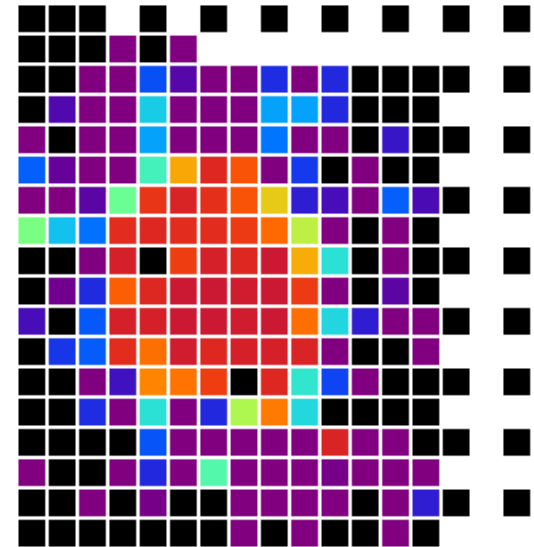
# Online Monitoring

- Ongoing work
- Integrating online data in a circular buffer
- Allows to display real-time physics histograms and curves
- Soon available for Root and R



# Hitcam

- Graphical visualisation of hits
- Online visualisation tool
- Adjustable gain
- Can be used offline : video generation
- Useful for :
  - beam spotting
  - online visual data control
  - beam supervision

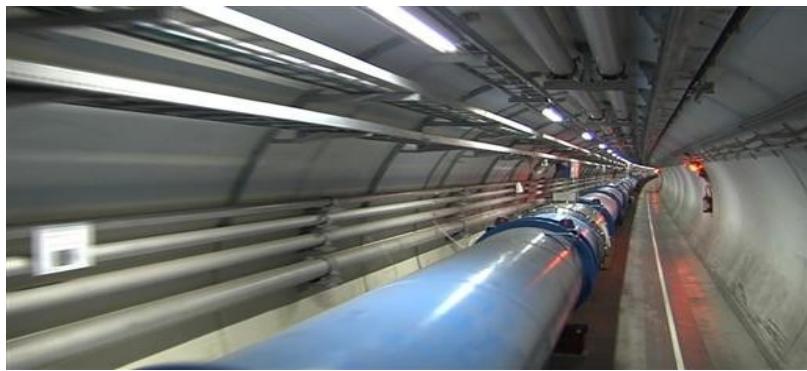


# Results



- Good data rate : perfect for 10 Hz
- Very good reliability
  - 15 days of beamtest at DESY without bugs
  - 3 weeks of calibration without a problem
- Easy to use
  - Multiples graphical interfaces
  - Programming interface
- Good evolutivity
  - Any block can be improved and modified easily by anybody





# Perspectives ?

- Increasing the flow rate by specific modification :  
plan to use UDP in GDCC
- Online event building : Eudaq / AIDA
- Integrating the system in a global SCADA system  
(XDAQ / AIDA)

# The Framework connection

- Not implemented now
- Connection to XDAQ for Lyon SDHCAL compatibility
- Can be easily compatible with any other framework Doocs, Tango...
- High level integration : replacing the detector GUI (the state machine steering)



# Output Perspectives

	Online Monitoring	Storage	Analysis
Future	AIDA Global Monitoring	LCIO Eudaq integration	Global Framework LCIO2Root
		AIDA Event Builder	
Summer 2013	ECAL Hitcam ECAL online R	Aggregated Plain Text one file per spill	AgregPlain2Root
		ECAL Aggregator	
February 2013	DIF Hitcam DIF online R	Plain Text ReadOut one file per dif	Plain2Root
		Skiroc Uncapper	
July 2012	DIF Hitcam	Raw Chip ReadOut one file per dif	Raw2Root



**Thank you for your attention, any questions ?**