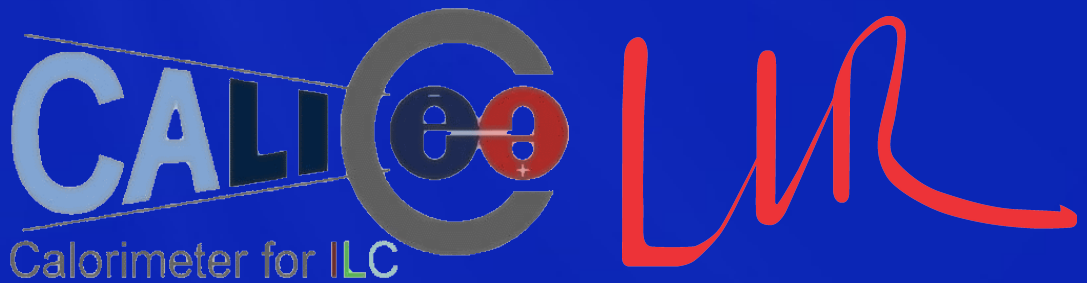
ReadOut & DAQ on European DHCAL

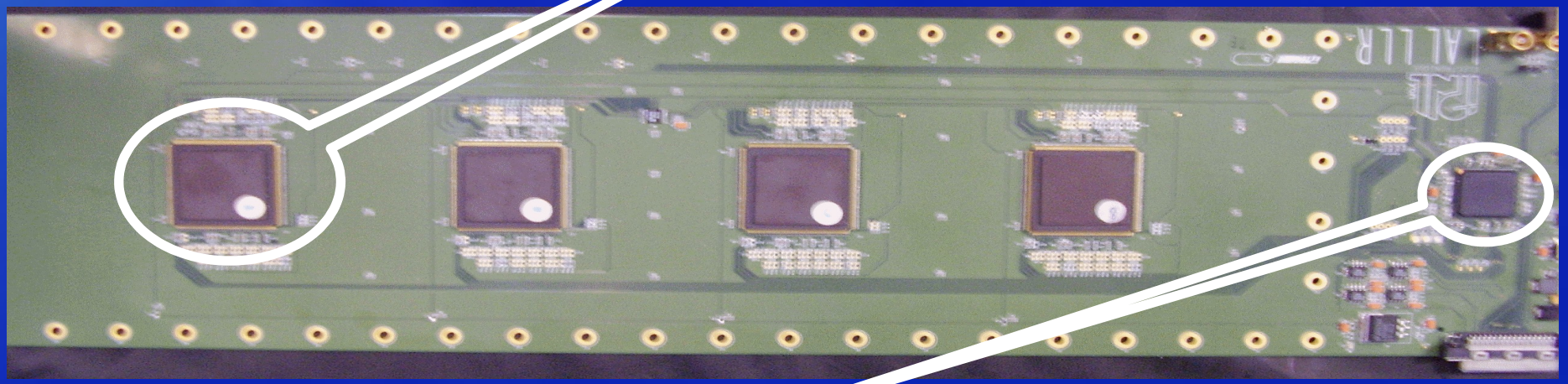


Clément JAUFFRET
LLR, Ecole Polytechnique - IN2P3 / CNRS

Calice Meeting, Prague - 12/09/2007

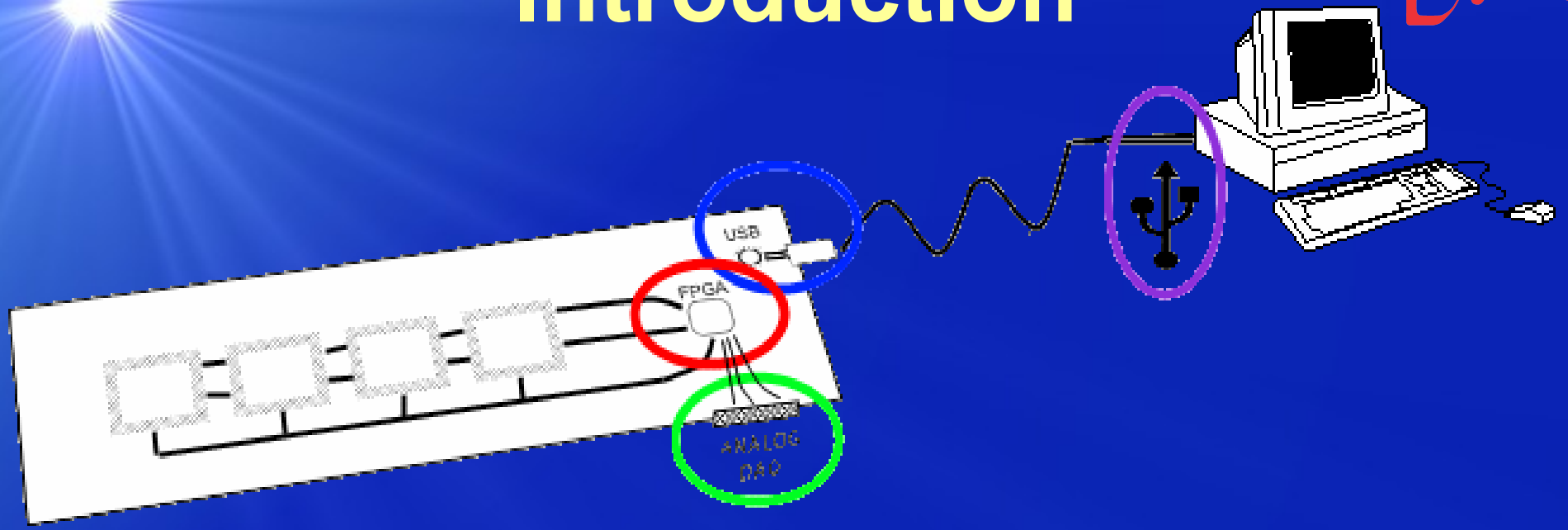
Introduction

- DHCAL Electronics - Objectives:
- DHCAL1 board : 4 LAL HardROC1 64 ch. ASICs over RPC.



- Code for onboard FPGA & PC/[next hierarchical board] by USB (for the moment) ⟺ DIF (Detector InterFace specific)

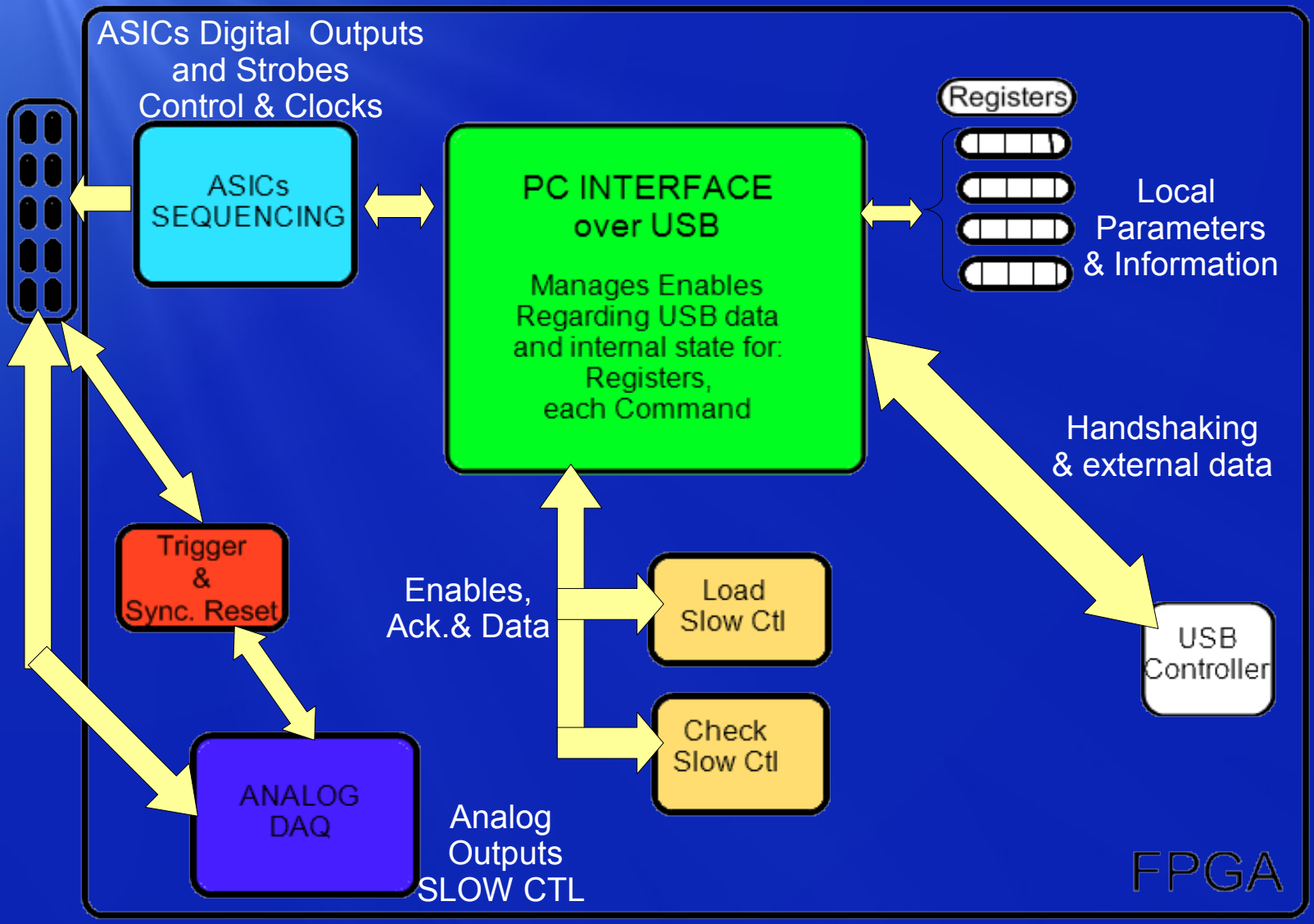
Introduction



- Handle ASICs parameters loading and checking, sequencing, data readout and sending out of detector.
- Initially made for and still compatible with 1 chip board
- Make complete digital acquisition chain.
- Be compatible with analog DAQ (from wich we have to receive trigger and time counter reset)

Firmware Developpement

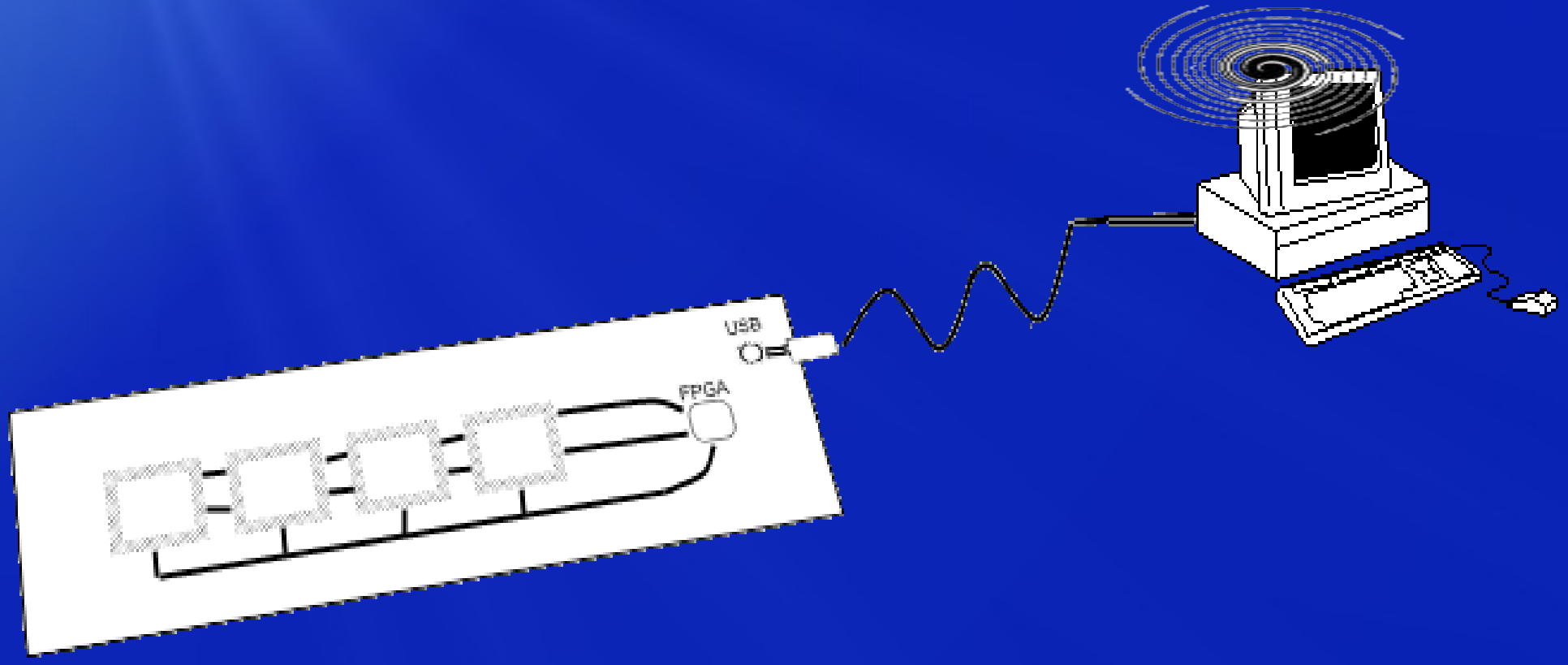
Schematic Block Diagram



Firmware Developpement

An Exemple by Practice

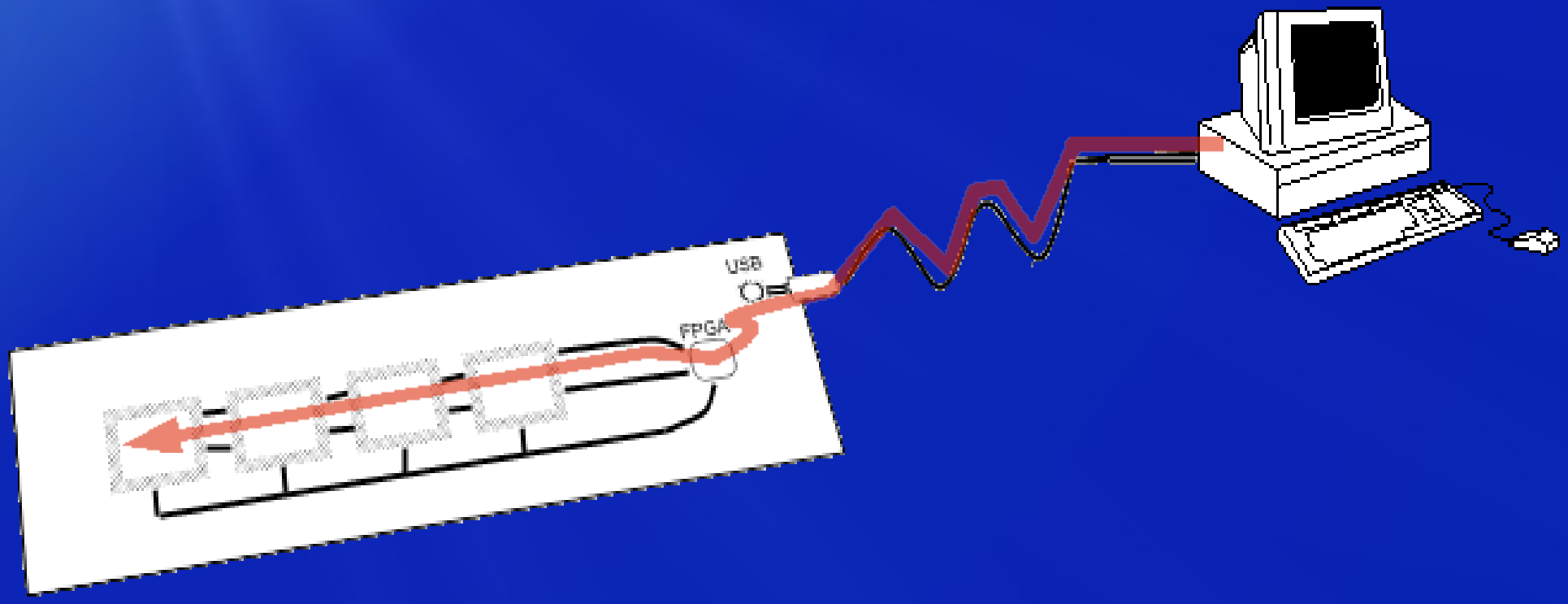
- Initialize shared memory and board



Firmware Developpement

An Exemple by Practice

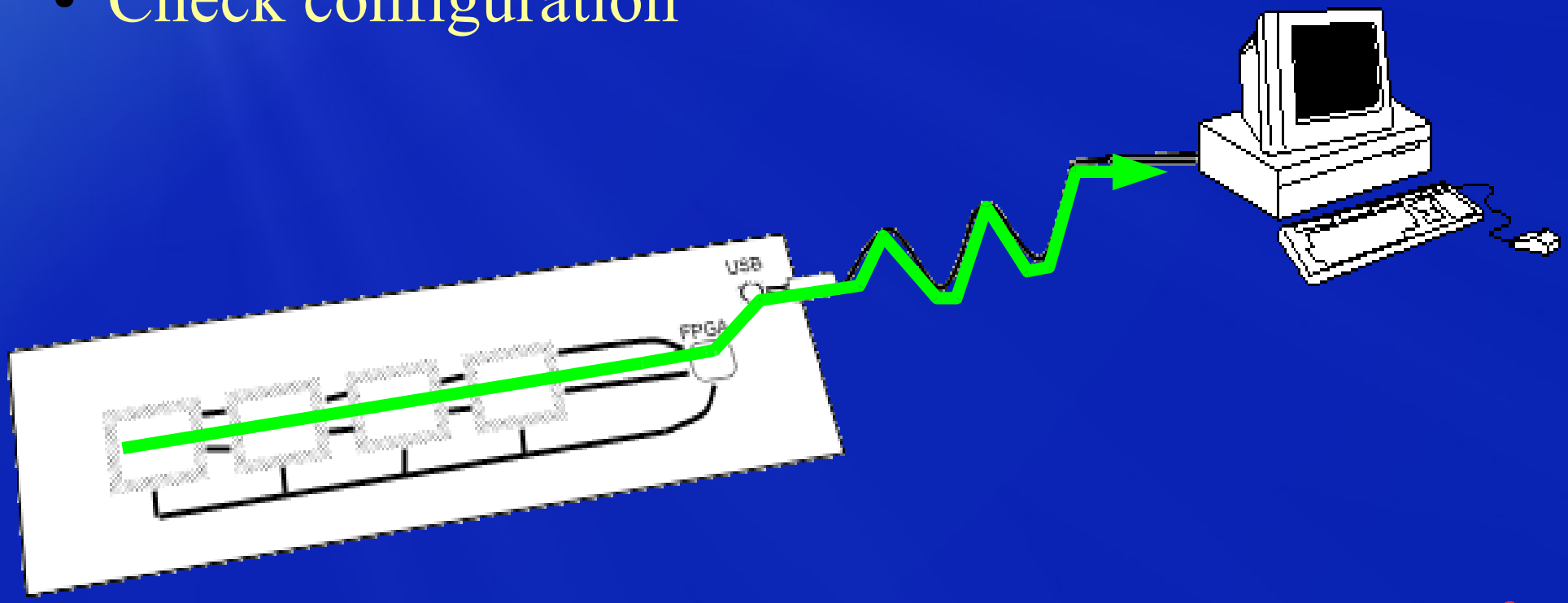
- Initialize shared memory and board
- Load configuration in ASIC



Firmware Developpement

An Exemple by Practice

- Initialize shared memory and board
- Load configuration in ASIC
- Check configuration

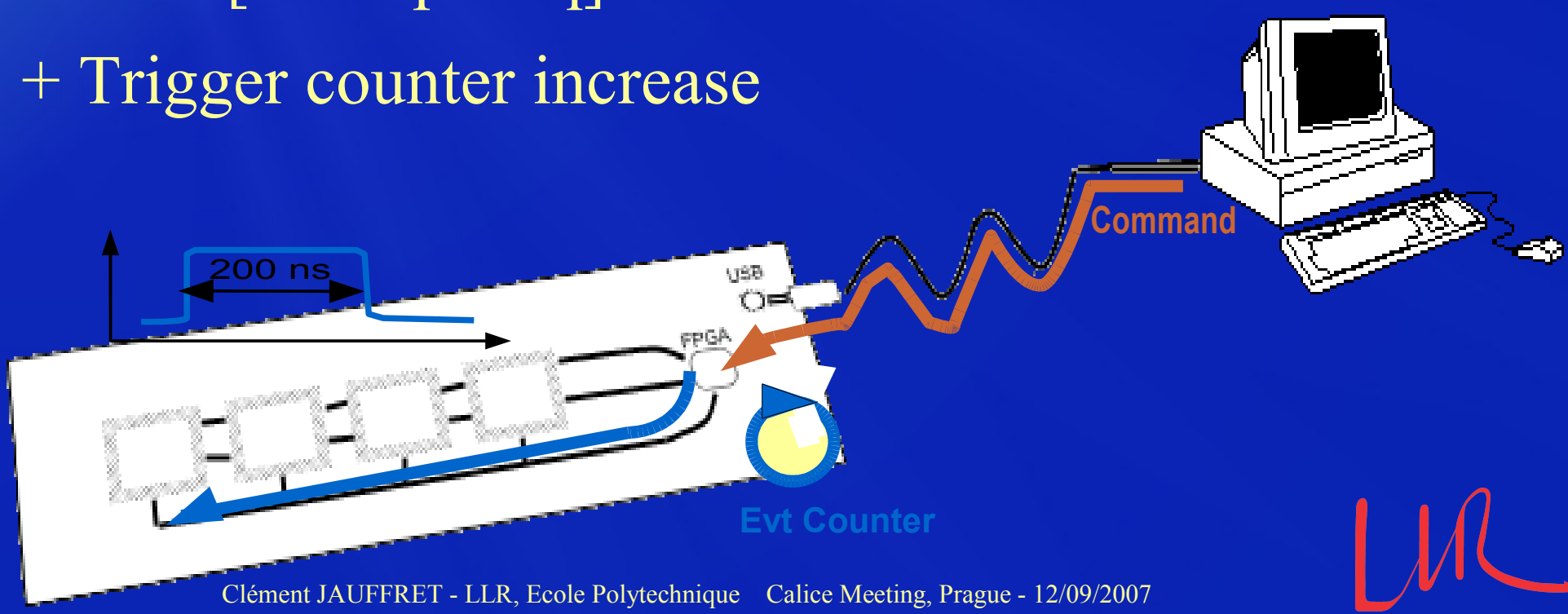


Firmware Developpement

An Exemple by Practice

- Initialize shared memory and board
- Load configuration in ASIC
- Check configuration
- Start [& Stop Acq]

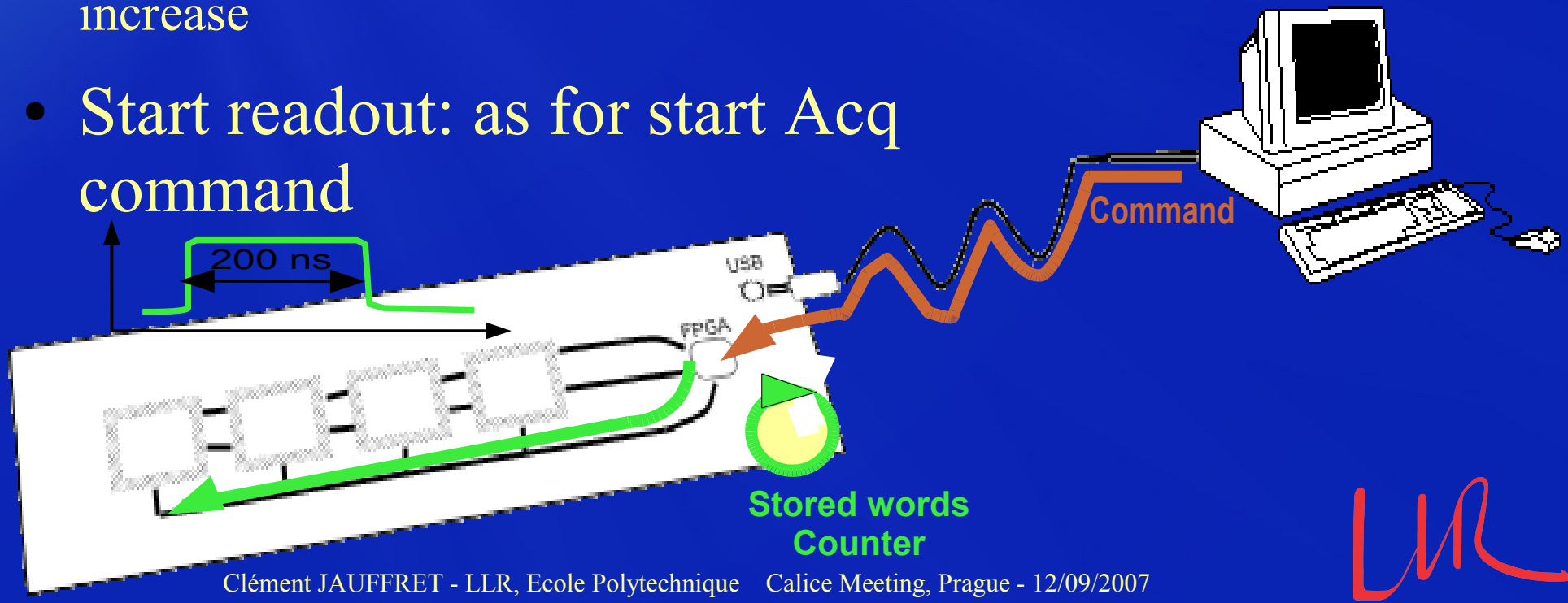
+ Trigger counter increase



Firmware Developpement

An Exemple by Practice

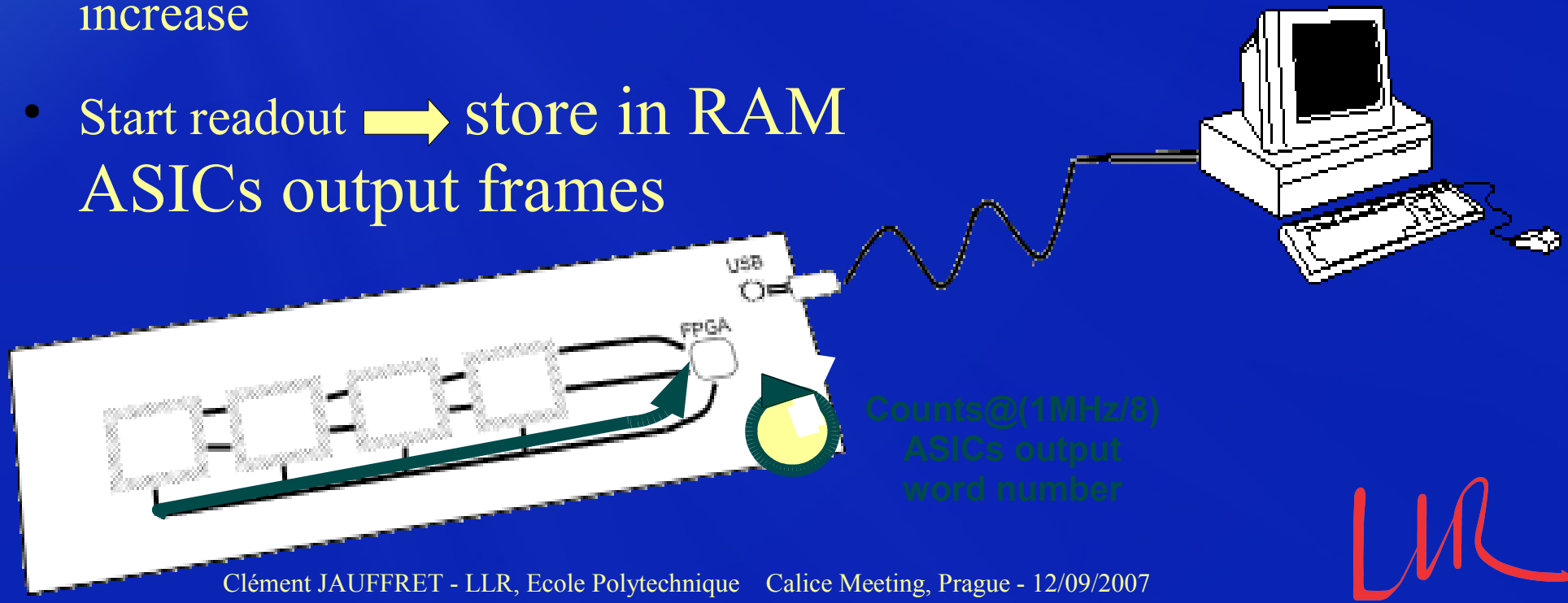
- Initialize shared memory and board
- Load configuration in ASIC
- Check configuration
- Start [& Stop Acq] + Trigger counter increase
- Start readout: as for start Acq command



Firmware Developpement

An Exemple by Practice

- Initialize shared memory and board
- Load configuration in ASIC
- Check configuration
- Start [& Stop Acq] + Trigger counter increase
- Start readout ➡ store in RAM ASICs output frames

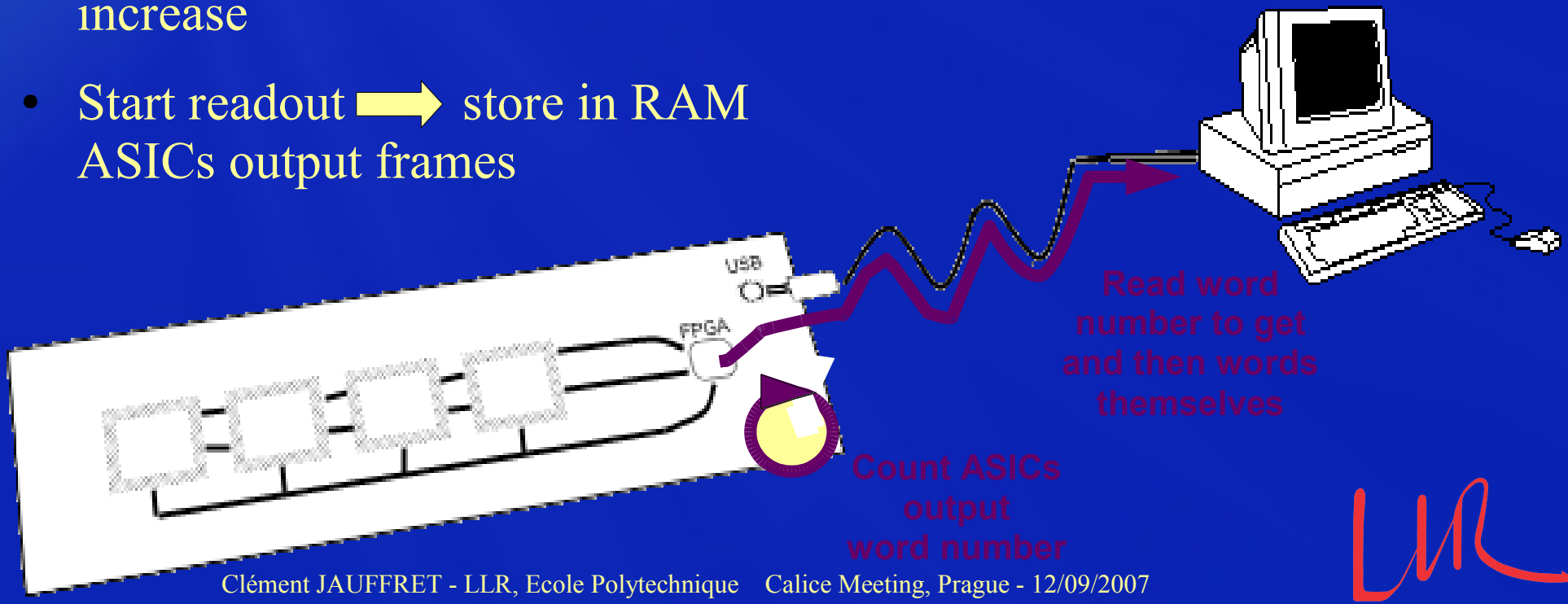


Firmware Developpement

An Exemple by Practice

- Initialize shared memory and board
- Load configuration in ASIC
- Check configuration
- Start [& Stop Acq] + Trigger counter increase
- Start readout ➡ store in RAM ASICs output frames

– Upload Data to PC: USB pass through way



Firmware Developpement

An Exemple by Practice

- Initialize shared memory and board
- Load configuration in ASIC
- Check configuration
- Start [& Stop Acq] + Trigger counter increase
- Start readout
- Upload Data to PC : USB pass through way
- Check CRC and Write Data to disk

⇒ Scheduled by software

Firmware Developpement

- Vhdl Package to set parameters depending on board & ASIC
- Compilator also work better with fixed parameters. there was solutions like parameter sizes depending on parameters loaded at runtime, structure much better optimized
- Make not use of either digital nor analog test points. Useful to debug ASICs but error or failure could be localized even without these test points

Software Developpement
Library

- Access directly LAL hardROC chip from PC
- USB: well understand functioning & timing constraints to adapt it to our situation

  - Try to evaluate parameters that impact on latency
  - Make use of special characters of usb norm for functions like send directly to minimize command throughput
  - Add timeouts as in firmware to be prepared to be faced with stucked access

Software Developpement
Library

- Protocol over USB: 2 types of access
- Registers : fixed length accesses

  32 bits, 10 address bits

  Allow Observation & Control

  e.g. see status, change of functioning mode
- Commands : make use of block transfers

  detector to acquisition distances ➡ non negligeable delays Minimise exchange nbr, make use of Protocole handling data transfer
- Configuration Files (csv) ➡ simple access to parameters

Software Developpement
Library

- Parameters make code reusable for others calorimeters because of use of LAL ASIC :

  ASIC nbr, Slow Ctl & Data Frames length
- Parameters fixed at compilation time to allow better optimization
- Vhdl + C Code soon available on the web for people who would wish to familiarize themselves with it

Software Developpement
Acquisition

- Command line or stream socket connection

  would allow to be piloted by analog DAQ software

  (adaptation kindly organized by P. Daunsey)
- More boards :
  - Identification by Serial number loaded in USB EEPROM
  - each board locked on a connection, differents tasks managed by threads (pthreads). Allow to have exactly same code, scheduling and dealing with possible conflicts

Software Developpement
Acquisition

- Code made to be portable under:
  - Windows NT > 4
  - Cygwin NT 5.1
  - Linux (RHEL 3.0 for the moment but more possible)
- Because use of standard C libraries, POSIX threads, readline
- Parts with conditionnal compilation: FTDI D2XX USB Library, UNIX sockets & Winsock2
- Documentation available

Endpoints and actual State of Tests

- Tests on cosmic bench soon at Lyon IPNL. More to be done...
- Still to do before test beams to be possible (End of October?) :
  - Socket interface with analog DAQ to test and improve
  - Make satisfying use of real trigger and other time critical signals

Conclusion

- Even if more tests are still necessary, results encouraging
- Software tested and optimized
- Acquisition chain of 2nd generation DAQ type almost finalized
- gives interesting informations for further developpements