ECAL LED system commissioning

A. Celentano

ECal LED system commissioning

The LED system is used in the first step of the ECal commissioning, before measurements with cosmic-rays and beam.

Tasks:

- Verify proper operation of all ECal channels, including the full amplification chain (crystal+APD+amplifier), the DAQ system, the cabling. Identify and correct cable-swaps (both signals and HV).
- Determine the working point of all LEDs, for both colors.
- Measure the cross-talk for each channel.
- Align channels in time.
- Check the effect of the recovery mode (LEDs continuously ON with blue color for radiation damage recovery) on the LED response (hysteresis?)
- Measure the LED system stability.

Pre-requirements

Calorimeter:

- Calorimeter installed in Hall-B.
- HV and LV system.
- Signal cabling to DAQ crates.
- Chiller connected + temperature readout (LED behavior is temperature-dependent).

LED system:

- Controllers installed in Hall-B and connected to the slow-controls network via Ethernet.
- EPIC software installed and working.

DAQ system:

- DAQ system configured to acquire data from FADC in both modes: RAW (full-waveform) and PULSE (pulse integral + timing).
- DAQ system configured to accept trigger from the LED system (feasible, already discussed with Sergey and Ben).

Software:

- Each task has specific software (analysis) requirements.

Task 1

Goal: verify proper operation of all channels, including the DAQ system. Identify and correct cable-swaps (signals and HV).

Specific requirements:

- Correct mapping of the LED system.
- First, preliminary working-point setting for each LED, resulting in a visible pulse at the amplifier output (signal amplitude is not critical at this stage!)

This information is available from tests performed by Ilaria Balossino in June.

Procedure:

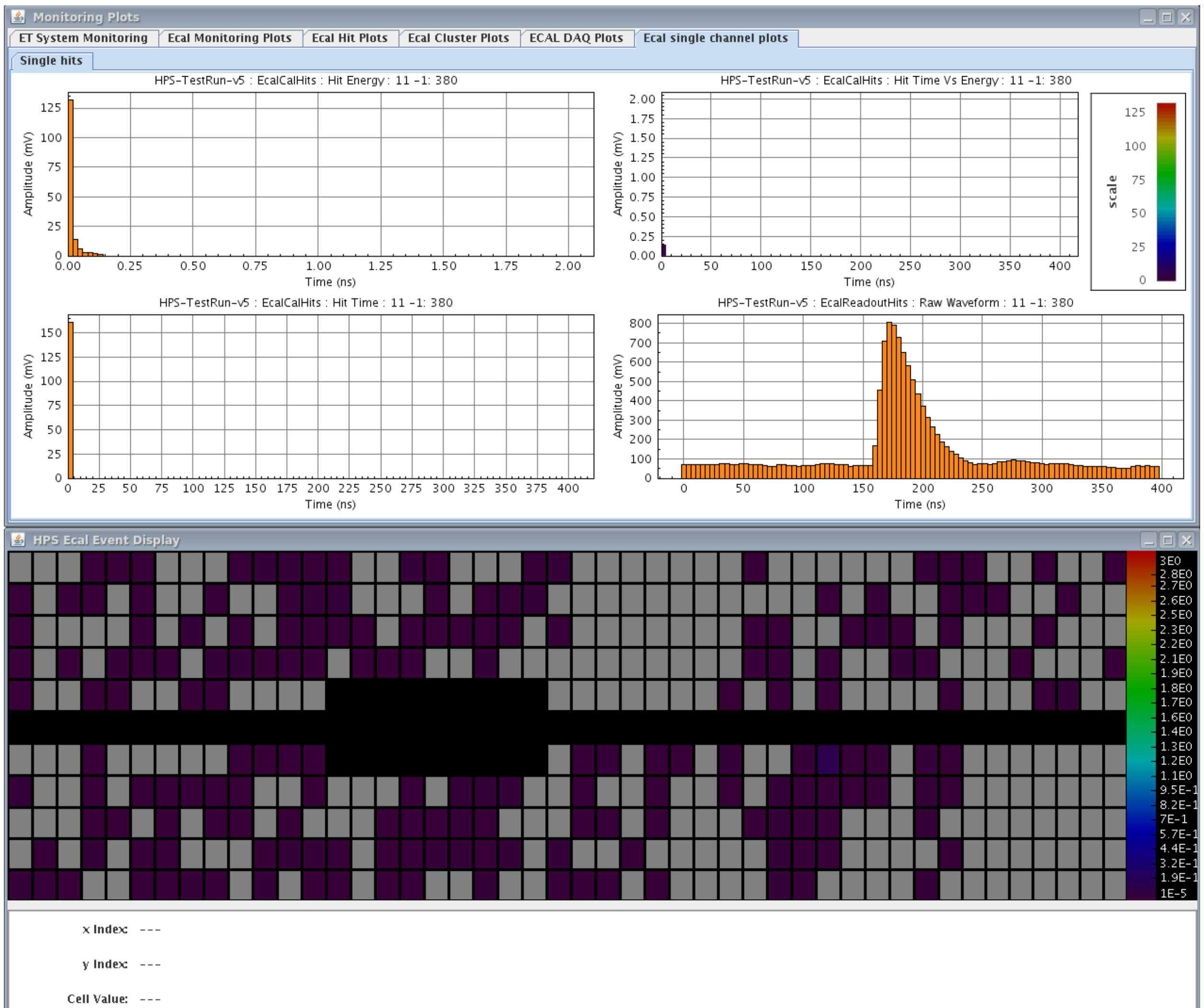
1) Turn on the DAQ system, with all channels enabled, taking data in RAW mode.
2) Switch on the LED connected to channel “n”.
3) Switch on only the corresponding HV group.
4) Use the online monitoring application to check the waveform of channel “n”.
5) Switch off the corresponding HV group and verify the pulse is no longer present.
6) Switch off the LED.
7) Repeat (manually) for all 442 channels.

Time required: 2-3 day (including time to identify and fix problems).

Task 1

Software:

- Use the online event monitor to check each channel wave-shape (amplitude vs time).
- Select crystal using the event display.
- No calibration constants are required at this stage (1 bit=0.4884 mV, FADC-hardcoded)



EVIO channel info:
Crate/slot/channel

Conversion to crystal id and position:
In hps-java (evio.ECalEvioReader), interacting with the conditions system

Currently:
Software is working assuming the test-run signal cabling scheme. Are we keeping the same scheme (yes)?

Software ready.

Task 2

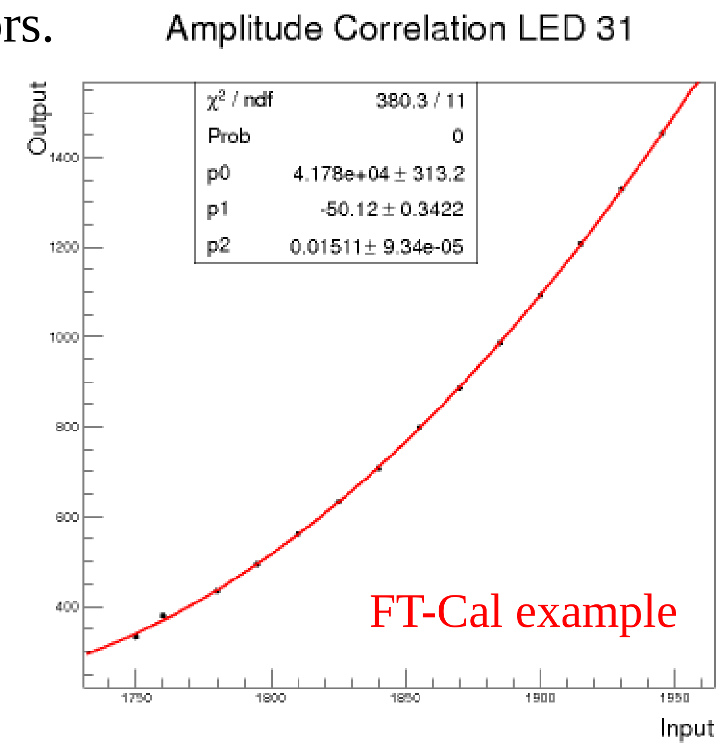
Goal: determine the working point of all LEDs, for both colors.

- The LED system is intrinsically non-linear.
- The function “amplifier output charge” vs “LED DAC setting” is LED-dependent, the difference is not a simple scale factor.
- We don't have an independent measure of this function.

→**Can't use the system for linearity checks.**

However, we need to determine for each LED a working point:

- Stability periodic checks
- Cross-talk measurements



Proposal: for each LED 2+2 working points: “high-blue”, “high-red”, (“low-blue”, “low-red”)

“High”: ~ 1.8 V, to measure stability and cross-talk.

“Low” : ~ cosmics, to periodically check if cosmics signal is visible.

Task 2

Procedure:

Ideally, we would have an automatic procedure that iteratively finds the LED working point, by starting from a certain configuration, taking data, measuring the resulting output amplitude, changing the LED working point (EPICS), taking data again, … , until convergence.

Is this feasible within October? Don't think so, unless someone volunteers to implement this procedure. **Otherwise, go manually:**

1) Turn on the DAQ system, with all channels enabled, taking data in RAW mode.
2) Switch on the LED connected to channel “n”.
3) Use the online monitoring application to check the waveform of channel “n” and measure the amplitude (in mV).
4) Iterate, until the amplitude is 1.8 V $\pm$ 10%. When done, save the working point.
5) Repeat for the other color.
6) Repeat for all 442 channels.

Time required: a lot!!! Considering only the “high” working point, 5 minutes/LED, 442 LED → **37 h**

But: can gain up to a factor 8, if more people work in parallel (different drivers are independent)

- Need to check if EPICS support this.

Software: use the same software as for Task 1, i.e. monitoring application

Task 3

Goal: measure the cross-talk between different ECal channels.

Specific requirements:

- LED "high-blue" working point (from Task 2)
- Configuration for the FADC PULSE mode (NSB,NSA).

Procedure:

1) Turn on the DAQ system, all channels enabled, PULSE mode, minimal readout threshold.
2) Start a sequence over all the LEDs (buit-in system feature), with 1 minute of data.
   - Switch on 8 LED at time (1 per driver).

Time required: ~1 h (1 minute/LED, 8 boards/time, 56 LEDs/board).

Analysis/software:

- Consider the events with LED "n" ON, compute the average energy for channel "n".
- For the same events, compute the average energy for all neighborhood channels.
- Obtain the cross-talk as the ratio.

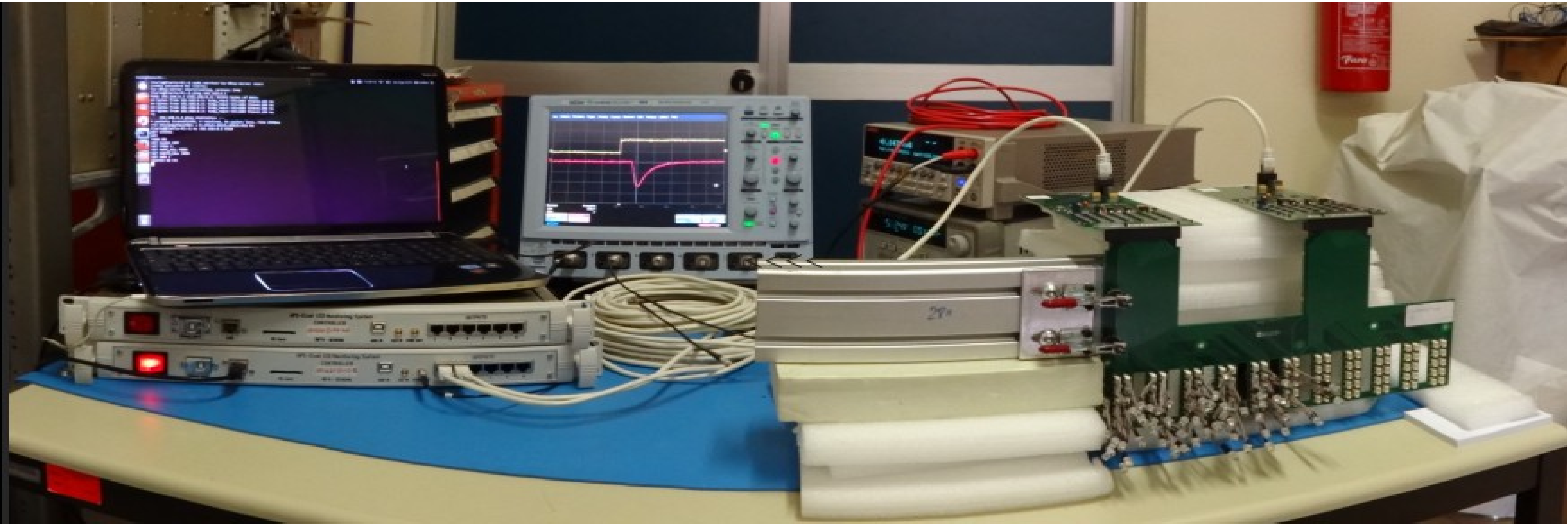
I suggest to proceed within the HPS-Java framework for this. Volunteers?
Calibration constants are not essential at this time, the analysis can be performed on "raw" energy (FADC units), and later on "calibrated" energy when cal. constants will be available.

Task 4

Goal: align channels in time.

Specific requirements:

- LED "high-blue" working point (from Task 2)
- Configuration for the FADC PULSE mode (NSB,NSA).
- Intrinsic time differences between different LEDs wrt the common clock (measured by Ilaria Balossino in June with an independent system)



Task 4

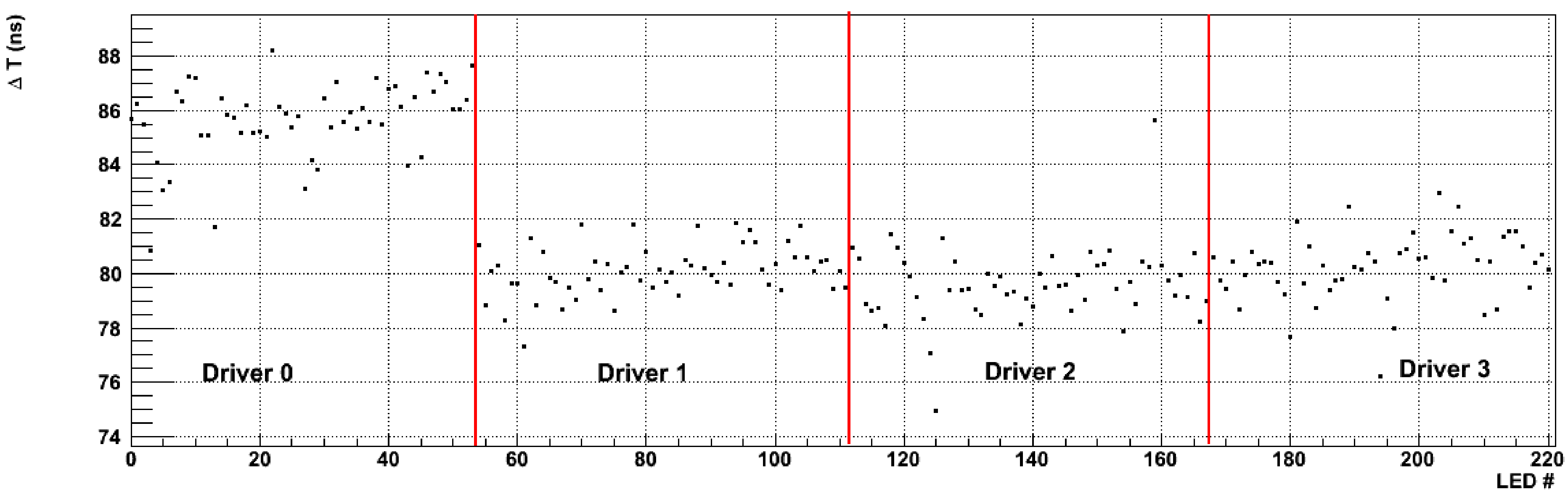
Procedure: the same as task 3, can use the same dataset

Analysis/software:

- Consider the events with LED "n" ON, compute the average time T_n for channel "n".
- Compare the relative delays with the intrinsic LED system delays.

I suggest to proceed within the HPS-Java framework for this. Same person doing task 3?

Caveat: the accuracy is O(1 ns). The intrinsic LED system delays were measured at a different LED amplitude (with a constant-fraction setup), and we saw a dependence of dT of ~ 1 ns over the whole amplitude range.



Task 5

Goal: check the effect of the recovery mode. For a given LED setting, is the channel response the same before and after the LED has been turned ON continuously for a long time (~1 day) with the blue color? Is there an hysteresis?

Specific requirements:

- LED “high-blue” and “high-red” working point (from Task 2)
- Configuration for the FADC PULSE mode (NSB,NSA).

Procedure:

1) Turn on the DAQ system, all channels enabled, PULSE mode, “reasonable” readout threshold.
2) Turn ON all LEDs in a sequence, as in task 3, with the blue color. Then, repeat with the red color.
3) Switch to recovery mode for ~ 1 day
4) Repeat again the two sequences before.

Time required: ~ 2 h + 1 day + 2h

Analysis/software:

- Consider the events with LED “n” ON, compute the average energy before and after, for both colors.
- Compare the results.

I suggest to proceed within the HPS-Java framework for this. Volunteers?

Task 6

Goal: measure the LED system stability.

Specific requirements: as task 5 + temperature data saved in the EVIO data-stream.

Procedure:

1) Turn on the DAQ system, all channels enabled, PULSE mode, “reasonable” readout threshold.
2) Start a sequence over all the LEDs as in task 3, with the blue color.
3) When done, start the sequence again from the beginning (built-in feature of the LED system).

Time required: ~1 h per iteration. Need O(20) iterations (the more, the better)

Analysis:

- For channel i, isolate in the data the events corresponding to each iteration j.
- Compute the average energy: $E_{i,j}$
- Plot $E_{i,j}$ VS j (i.e. vs time) and evaluate the stability. Correlate with temperature.

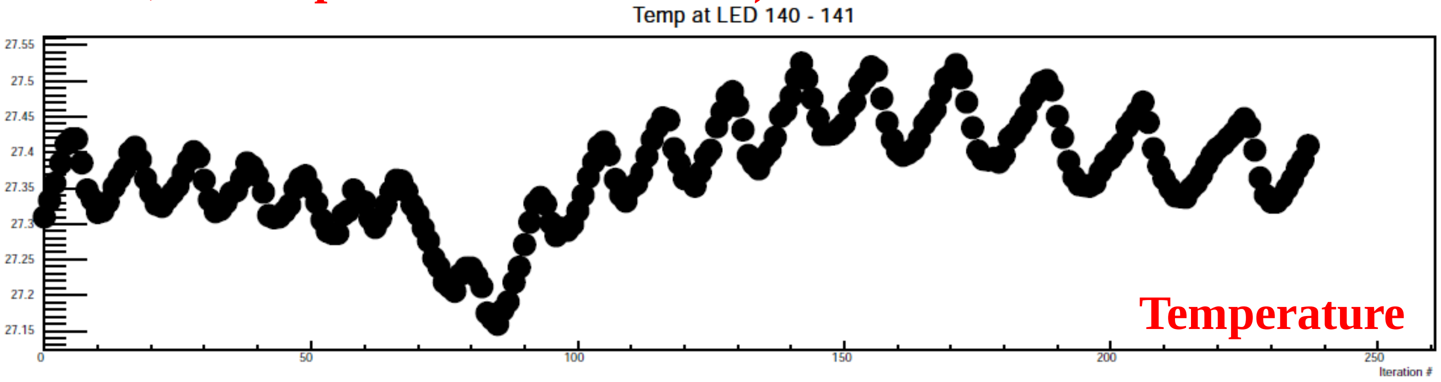
Task 6

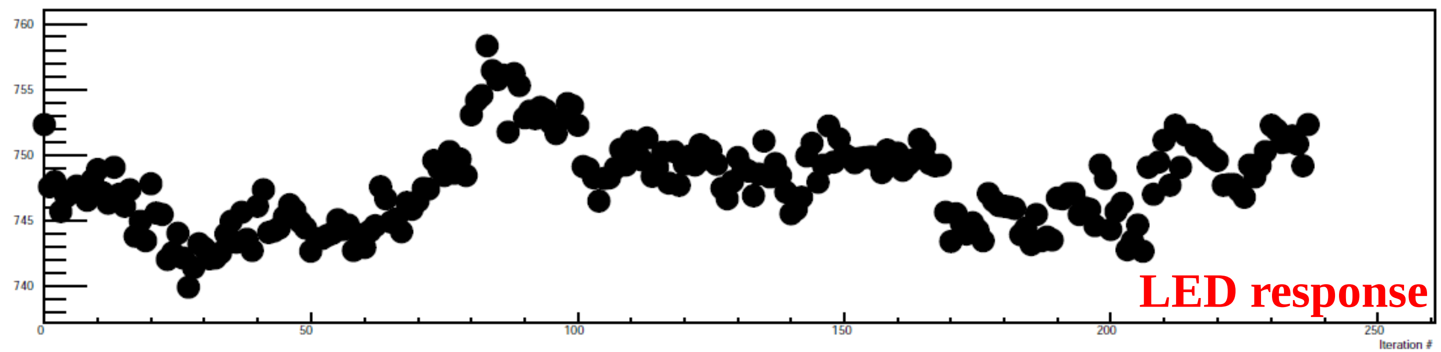
Software:

We already have a software to perform the analysis, matched to the FT-Cal setup, using C++ and ROOT.

My suggestion: adapt this software to HPS, reading directly the EVIO output file.

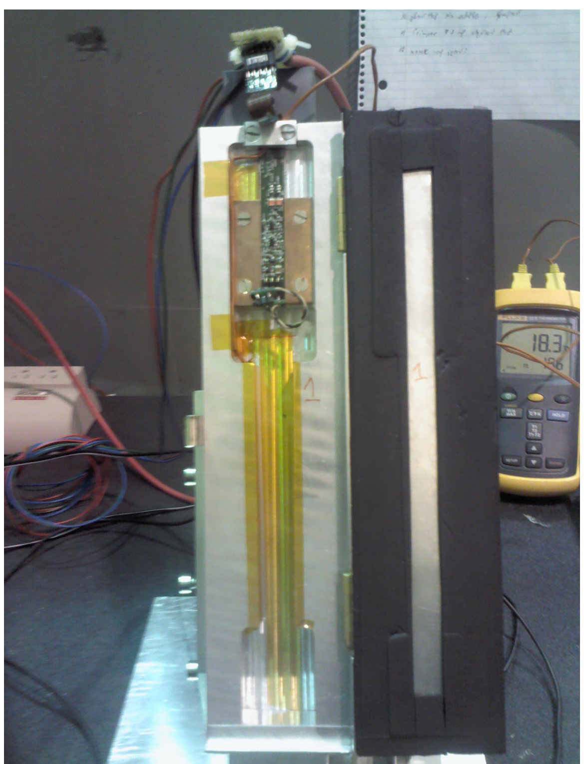
FT-Cal example (1 LED only, 1 iteration every 5 minutes, no temperature stabilization)





Software requirements summary

Task	What	Status	Who?
1 (cabling test)	Monitoring app (waveforms) + Event display	100%	Andrea
2 (working point)	Monitoring app (waveforms) + Amplitude histogram + Event display	80%	Andrea
3 (cross-talk)	HPS-java. Read hit energy / make histogram / compute average / make ratio with neighborhood channels.	0%	?
4 (timing)	HPS-java. Read hit time / make histogram / compute differences / compare with LED intrinsic delays saved in the conditions database	0%	?
5 (DC effects)	HPS-java. Read hit energy / make histogram / compute average (before and after DC mode). Compare the two results.	0%	?
6 (stability)	Re-use existing code. Need to adapt to specific EVIO format.	50%	Andrea



Question:

While waiting for Ecal in the Hall, is it possible to connect the crystal test setup to the DAQ and acquire some data, with any LED amplitude configuration?

- Equivalent LED system setup (trigger): configure DAQ
- Data is useful to prepare software for task 6