

Bologna: activity update

Outline:

- testbench status update
- WR synch preliminary measurements
- DU-base embedded-sw development

Riccardo T., Giuliano P., Tommaso C., Carmelo P.

Testbench status update

4 CLB connected in a full Km3Net WR fashion (fw tag rev20150114)

2 are ver 2.2

2 are ver 2.2.1

(http://wiki.km3net.de/index.php/Bologna_Common_Infrastructure)

2 KC507 used to inject data through:

4 Pseudo-Octopus

2 Octopus

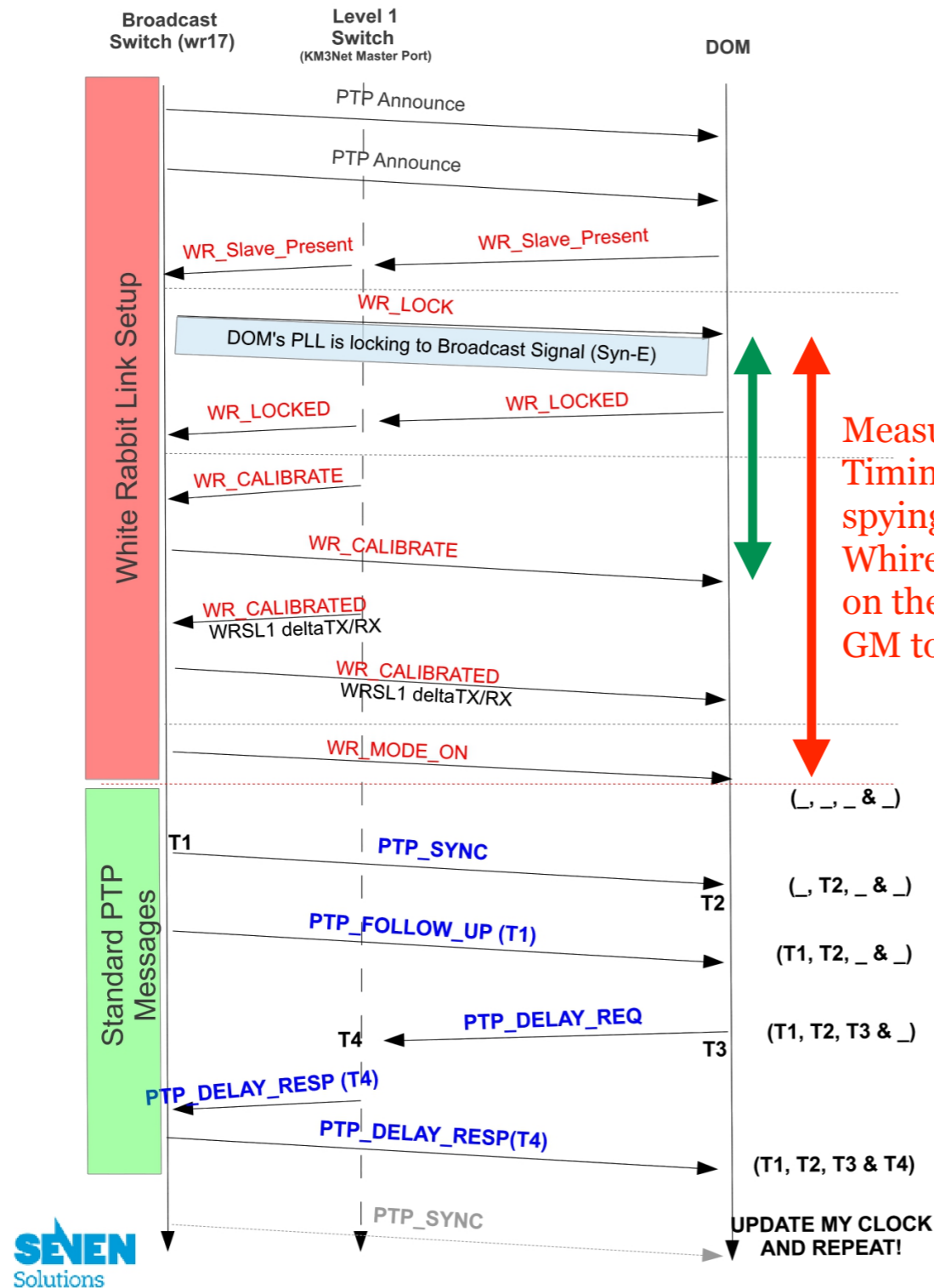
(we still have 2 proto:

one of them is used by me for DU-base embedded-sw development;
we plan to increase later our setup up to 6 CLBs.)

Test Plan

- 1) test DAQ in Full-WR with different data rates (almost ready to start)
- 2) test Slow Control (remotely by Cristiano - in progress)
- 3) test DAQ for Piezo (Carmelo - in progress)
- 4) measure WR starting time in a many CLB environment (in progress)
goal: provide some hints

WR link setup measurements (1)



Measurement:
Timing interval measured by
spying PTP messages with
Wireshark
on the Broadcast (from WRS-
GM to CLBs)

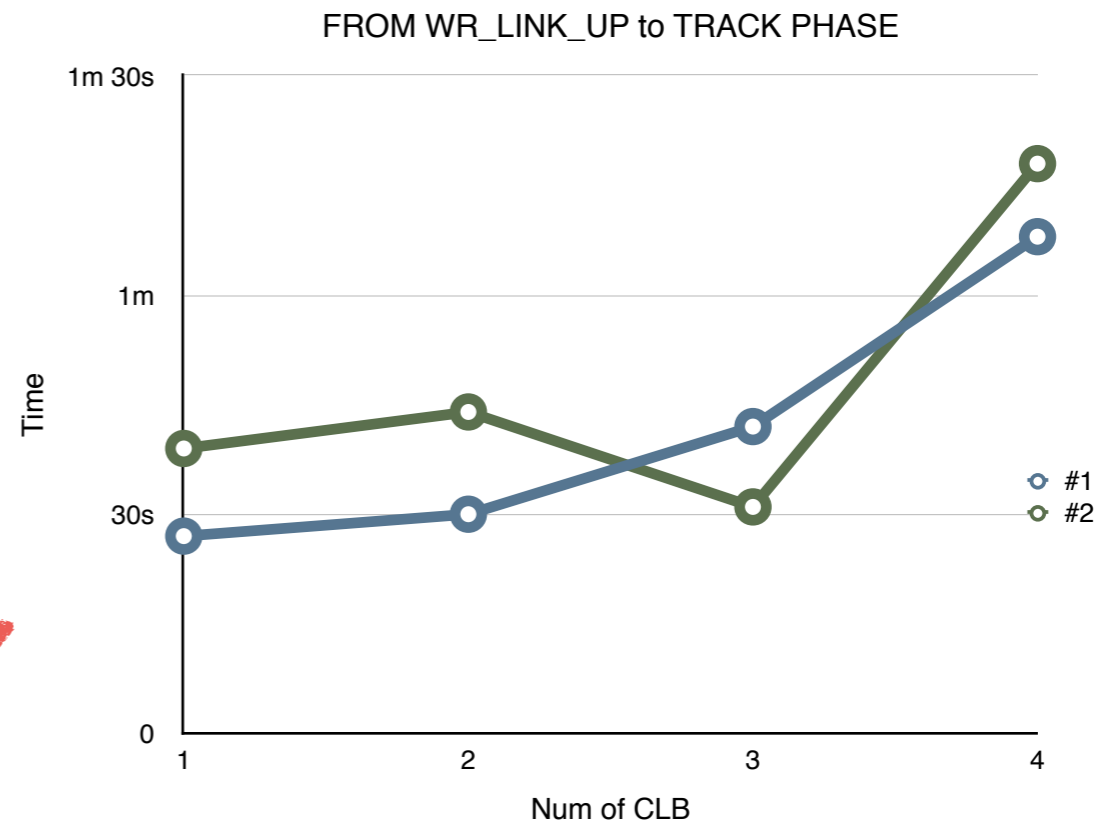
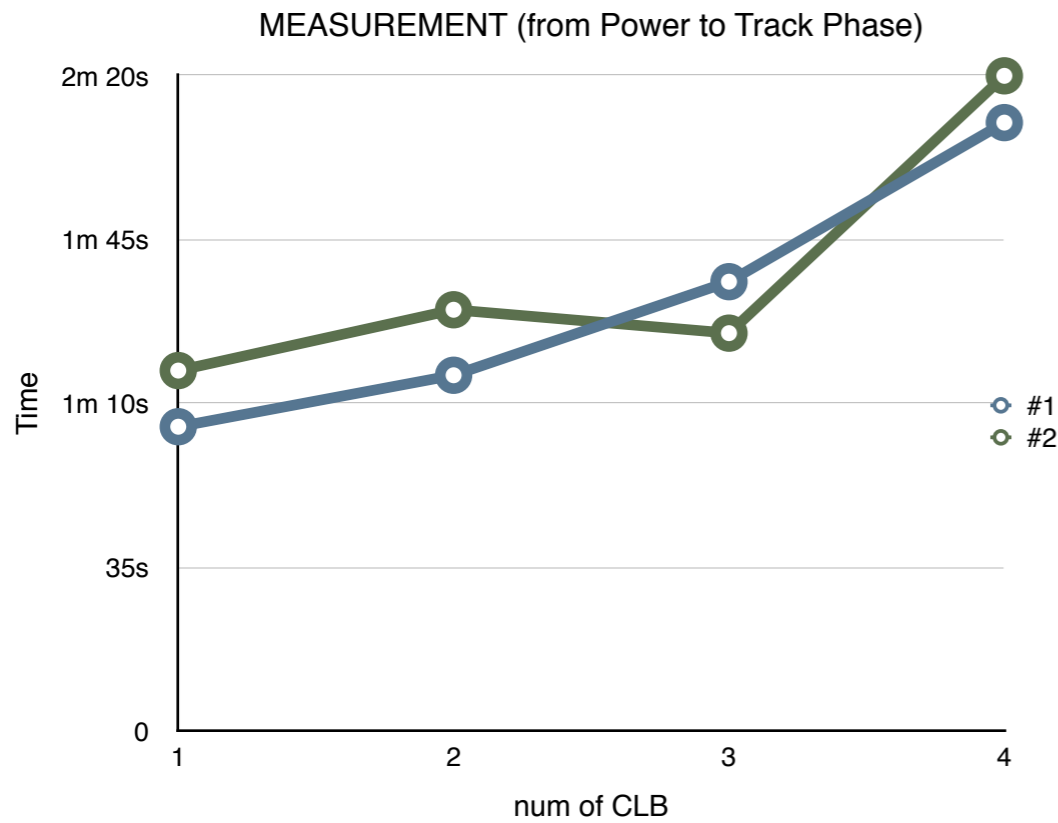
RESULT: 8.2 s

- 1) measurement independent on the number of CLB switched on.
- 2) most of the time (8.1 s) between WR_LOCK and WR_CALIBRATE (green arrow)

Figure 3: KM3Net WR flow message.

From "White Rabbit over KM3Net."
(Seven Solutions).
10 sep. 2014 -

WR off->trackphase measurements (2)



Apply

- 1) Bias: 30 s from power-on to booting runtime fw image
- 2) Bias: $8s * \text{number of CLB}$ (previous slide)

We observe that WR CLB GUI spend most of the time in the following states:
 “master mode or sync info not valid”
 “Servo state: uninitialized”

DU-base embedded-sw development

Preamble: in the last DAQ meeting (December '14 in Bologna) I agreed to provide an initially time-limited support to Vincent about the embedded-sw for the DUbase (time-limited = now 3 months - later it will be re-negotiated accordingly to needs as well as to my availability)

Done a fruitful meeting with Vincent (sw overview and planning)

ToDo list:

- create a separate building placeholder:
 - done and put into SVN : /sw/embedded/CLBv2_App/build/base
 - a carbon-copy of /build/runtime

- remove all sw stuffs not needed (i.e. optical subsystem)
 - in contact with all involved people to do an inventory of what's in the DUbase and how to manage it (Tommaso made a mailing list).
 - in progress : some docs received last Thursday

- create drivers
- create slow control commands
- create variables
-

I prepared a small test bench (1 Proto CLB) to test my changes.

Open issue (in my mind):

- Who's in charge to tune CLB fw for the DUbase (has to be added to the mailing list?)