TELL1 command line tools

Guido Haefeli
EPFL, Lausanne

Command line commands

- Five functions are defined which are installed on the CCPC.
- Be careful using the correct version of the functions!
- Some of these functions depend on the TELL1 FPGA firmware !

Things to know about these functions:

- daq_tell1, cfg_tell1, console_tell1

(depend strongly on the version of the FPGA and the c-code)

- version_tell1 , reset_tell1

(independent of the versions)

Use the --help option to find the accepted options

What are they used for

- These functions can be extended by the user for specific needs, to test (debug) new a VHDL implementation!
- As long as there is only one TELL1 board to control and monitor it is a very efficient way to operate the boad.

I use the functions for:

- Running the selftest, running the TELL1 board with a predefined .cfg file which allows to process data with the data generator to the DAQ
- Checking the event data in the MEP buffer, the decoding functions for all specific banks need to be profided
- Checking the processing by running "bit perfect c-models" and comparing them to the MEP buffer contents.

Monitoring is easier on the PVSS project

Known problems with these functions v2.1 and earlier!

- In the c-code version earlier than tell1lib_release_v2.1, the functions if executed at the home directory of the users are very-very slow, the makefile for the later releases has been modified and it runs without searching for 1' the libraries.
- The earlier version of the reset_tell1 were not re-loading the FPGAs but only resetted the FPGA logic, with option –p the reload was possible, after version tell1lib_release_v2.0 the reset also reloads the FPGA with the current EEPROM content.
- daq_tell1, cfg_tell1, console_tell1 and version_tell1 check the correct initialization of the FPGAs. The correct initialization is new checked by a write-read and check access on one a register on the FPGA, this check avoids the FPGA ini problem that was observed and reported as a bug. tell1lib_release_v2.1,

version_tell1

- # TELL1 library [v2.2 @Nov 8 2007 12:28:28]
- # ------- Fixed Information in FPGA/EEPROM --------
- # TELL1 ID [4ODLAUTL000331] : (at: 31/3/2007 11:6:0)
- # GBE ID : 98 00 21 10
- # FPGA InitDone : 1F [OK]
- # User PP_FPGA code version: 2.2
- # Common PP_FPGA code version: 2.2
- # Common SL_FPGA code version: 2.2
- # SL Built Number : 72
- # PP Code generated at :23/11/2007--08:32
- # SL Code generated at :23/11/2007--08:32
- # Detector ID of TELL1 : VELO
- # ChipAddr of 4PPs : 3210 [OK]

- [lphe1tell14] /home/cc/tell1lib_release_v2.2.1 >

daq_tell1 or cfg_tell1

```
[lphe1tell14] /home/cc/tell1lib_release_v2.2.1/ccpc_daq > ./cfg_tell1 VELO4.v23.cfg

CFG: Read  VELO4.v23.cfg

======================================== TELL1 for [VELO]
    ========================================
ACTION :  ini_GLUECARD_PLX                                    [     OK]
ACTION :  reset_TELL1                                       [     OK]
ACTION :  reset_TELL1_logic                                   [     OK]
ACTION :  load_TELL1_FPGA                                     [     OK]
ACTION :  reset_GBE                                         [     OK]
ACTION :  ini_GBE_MAC                                         [     OK]
.
.
.
ACTION :  config_PHI_Reorder                                  [     OK][ Disable]
ACTION :  ini_Reorder                                       [     OK]
ACTION :  ini_VELO                                          [     OK]
ACTION :  ini_TELL1                                         [     OK]
======================================== TELL331  Ready to GO
    ========================================
[lphe1tell14] /home/cc/tell1lib_release_v2.2.1/ccpc_daq >
```

console_tell1

```
[lphe1tell14] /home/cc/tell1lib_release_v2.2.1/ccpc_daq > ./console_tell1
              =======Running Console on TELL1======
ACTION :  ini_GLUECARD_PLX                                          [      OK]
ACTION :  read_TELL1_CONSTANT                                          [      OK]
                      ************************************
                      *   C : TELL1  Control             *
                      *   M : TELL1  Monitor              *
                      *   T : TELL1  Test                *
                      *   P : TELL1  Programming          *
                      *   F : TELL1  export Files        *
                      *----------------------------------*
                      *   D : Detector specific          *
                      *----------------------------------*
                      *   A : About                      *
                      *   Q : Quit                       *
                      *   ? : show this manual           *
                      ************************************

CMD--->   [?]
```

console_tell1

- Warning: Don't start the console_tell1 while an other function does intensive read-write access to the board, this can create access conflicts
- Use ^h (ctrl-h) to delete wrong typed characters

- Most commonly used option to check on the board status:
  - **M (monitor) 8 (Read ttc monitor counters)**
  - **M (monitor) a (Read memory max usage)**
  - **D (detector specific) l (Read and Parse MEP during run)**

Warning: Reading the data from the board takes ~ms which is enough to create throttles, (if there is no throttle enabled and feed back to the Readout Supervisor, buffer overflows can be caused

Note: All the registers can also be monitored via PVSS

Running the tell1 commands from a local directory

- Copy into a local directory the contents of the tell1lib_v2.2.zip
- Execute all command in tell1lib_v2.2/ccpc_daq/
- For example: ./daq_tell1 VELO4.v23.cfg
- In ccpc_daq there is one .cfg file per sub-detector as an example configuration

Note: This is the way to run the tell1 c-code to assure consistent versions without the need of the crtl PC yum update. This is a way to implement a hardware test function

➔ To provide a hardware test function.

cfg_tell1 is used for the PVSS project

- Warning: For the PVSS the two of the tell1 commands are also used, cfg_tell1 and reset_tell1 . PVSS is using the installed functions on the ctrl PCs.
- Problems observed:
  - We must use an up-to date version in order to avoid that PVSS requires to write a new updated .cfg file (you can see this by using the cfg_tell1 command on the command line.

```
[lphe1tell14] /home/cc/tell1lib_release_v2.2.1/ccpc_daq > cfg_tell1 VELO4.v21.cfg

CFG:  Read  VELO4.v21.cfg

>>>>>>>>>>>>
A new DB version is used now. We will convert to this new DB format.
All your settings will be kept, but the comment you add will lost.

CFG:  Write VELO4.v23.cfg

CFG:  Write VELO4.v23.cfg

======================================== TELL1 for [VELO]
    ========================================
ACTION :  ini_GLUECARD_PLX                                  [      OK]
[lphe1tell14] /home/cc/tell1lib_release_v2.2.1/ccpc_daq >
```

Selftest

- The selftest for the boards is provided using a set of .cfg files which are used to configure the board of a particular sub-detector into a "known to work" state.
  - ST2.v23_selftest.cfg with tell1_ST_v2.3.1.pof
  - VELO4.v23_selftest.cfg with tell1_VELO_v2.3.1.pof
  - EHCAL3.v23_selftest.cfg with tell1_EHCAL_v2.3.1.pof
  - LOCAL2.v23_selftest.cfg with tell1_LOCAL_v2.3.1.pof
- Trigger sent by ECS, accepting the TTC information from the Odin means changing 3 parameters only
  - Change lines 15,17,18 each to 1 is changing the selftest to be done with TTC

Running the selftest overview (with local tell1lib)

1. Make a local copy of the tell1lib on your ccpc ctrlPC eg. itdaq01 for IT
2. Program the FPGAs with the required version
3. Run the selftest in your local folder with your local tell1lib

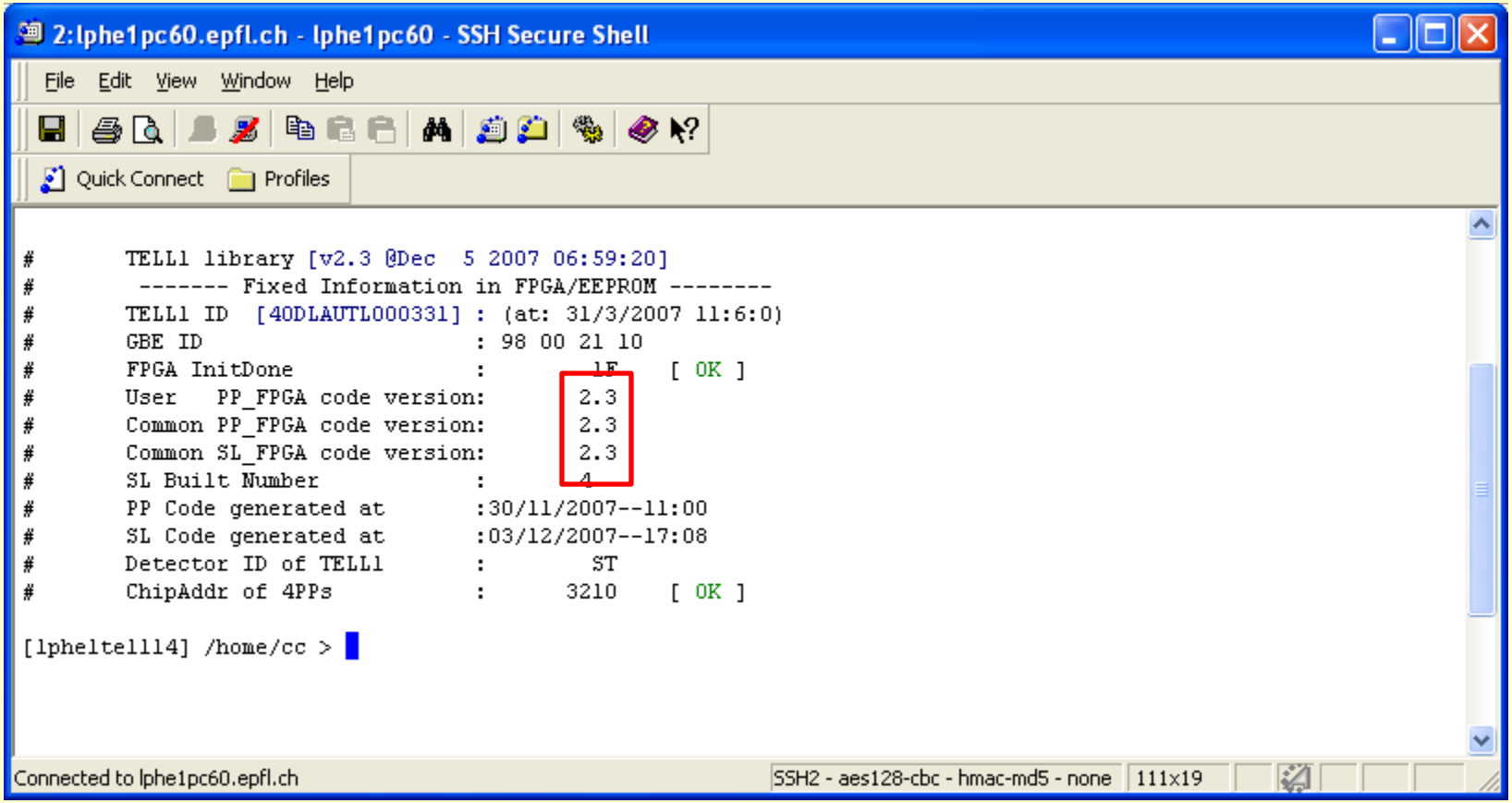
Running the selftest (with local tell1lib)

1. Download the c-code zip file for the tell1lib from: http://lphe.epfl.ch/tell1/VHDL%20Framework.htm
2. In the folder tell1lib_release_v??? The .pof files for each subdetector is given.
3. Make sure that the FPGAs are programmed with the version needed.
   - Check with version_tell1 the FPGA version.
   - If needed flash the EEPROM with EPCHandling –e –p 1 file.pof
4. Run the selftest in ccpc_daq, using ./daq_tell1 ST2.v23_selftest.cfg

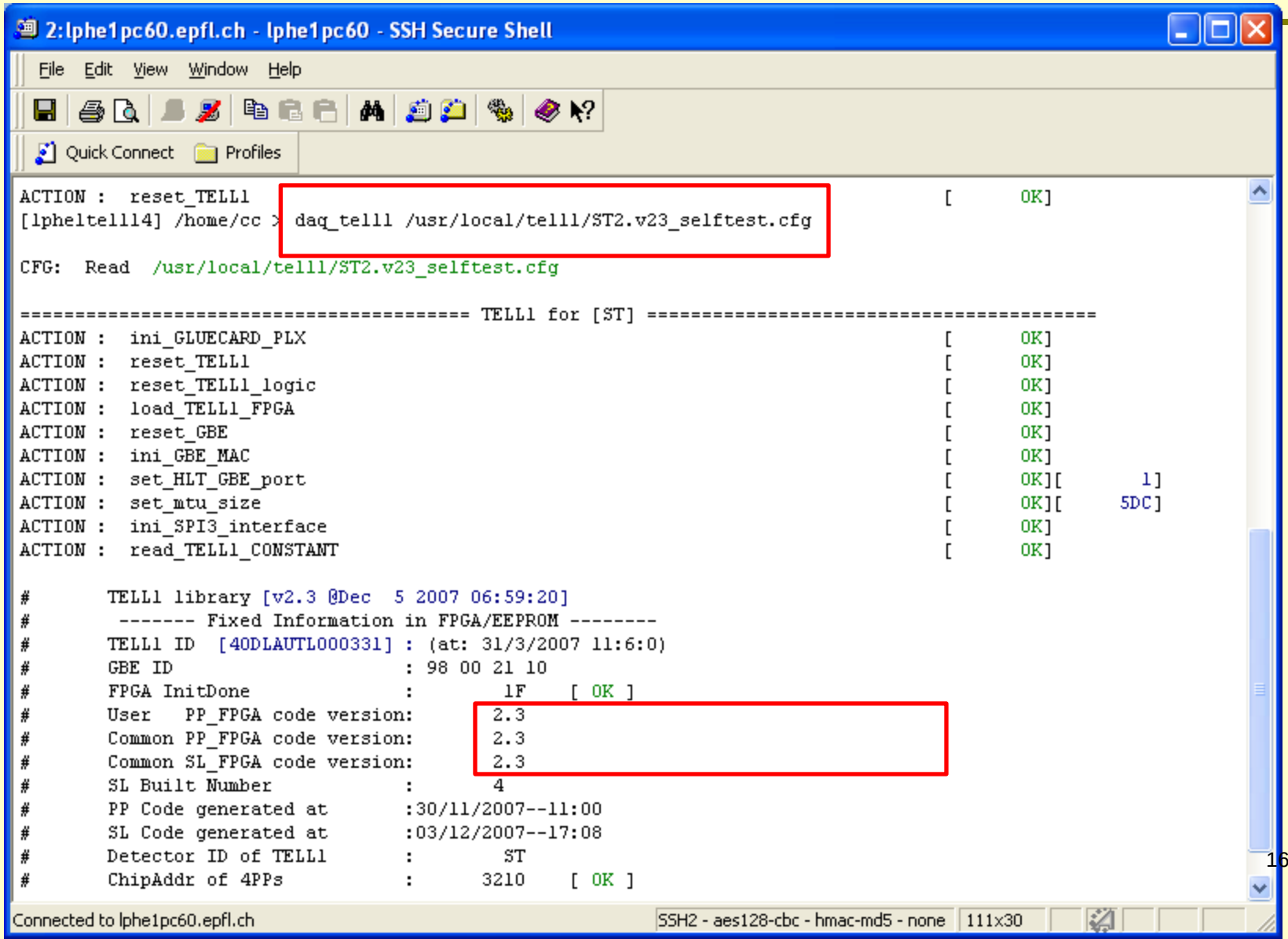
Running the selftest overview (with installed tell1lib)

1. Make an upgrade of the tell1lib installed on your ctrlPC eg. itdaq01 for IT
2. Program the FPGAs with the required version, a copy of the .pof files is always distributed and placed in: /usr/local/tell1/
3. Run the selftest using the .cfg files also available in /usr/local/tell1/

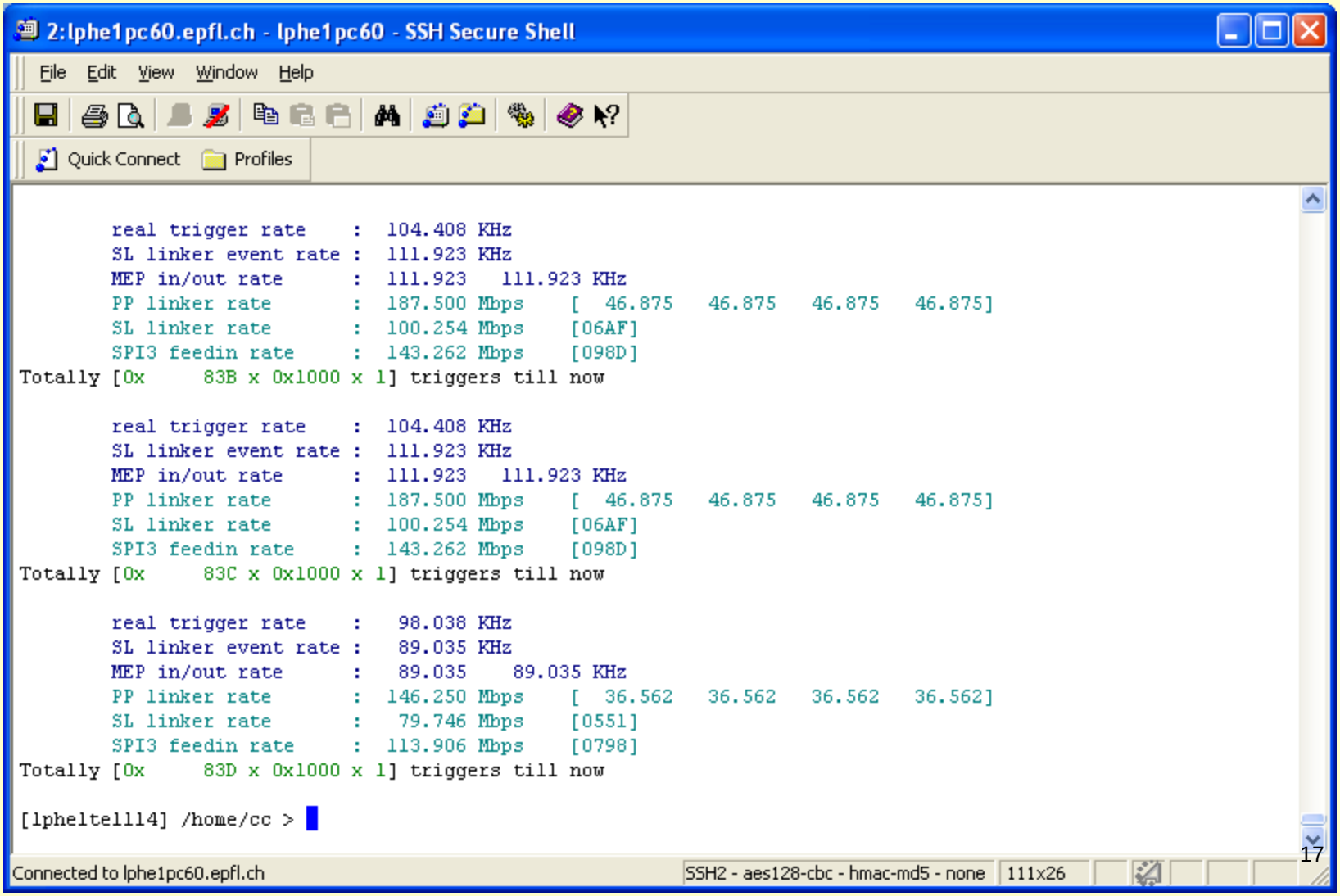
Running the selftest (version check)



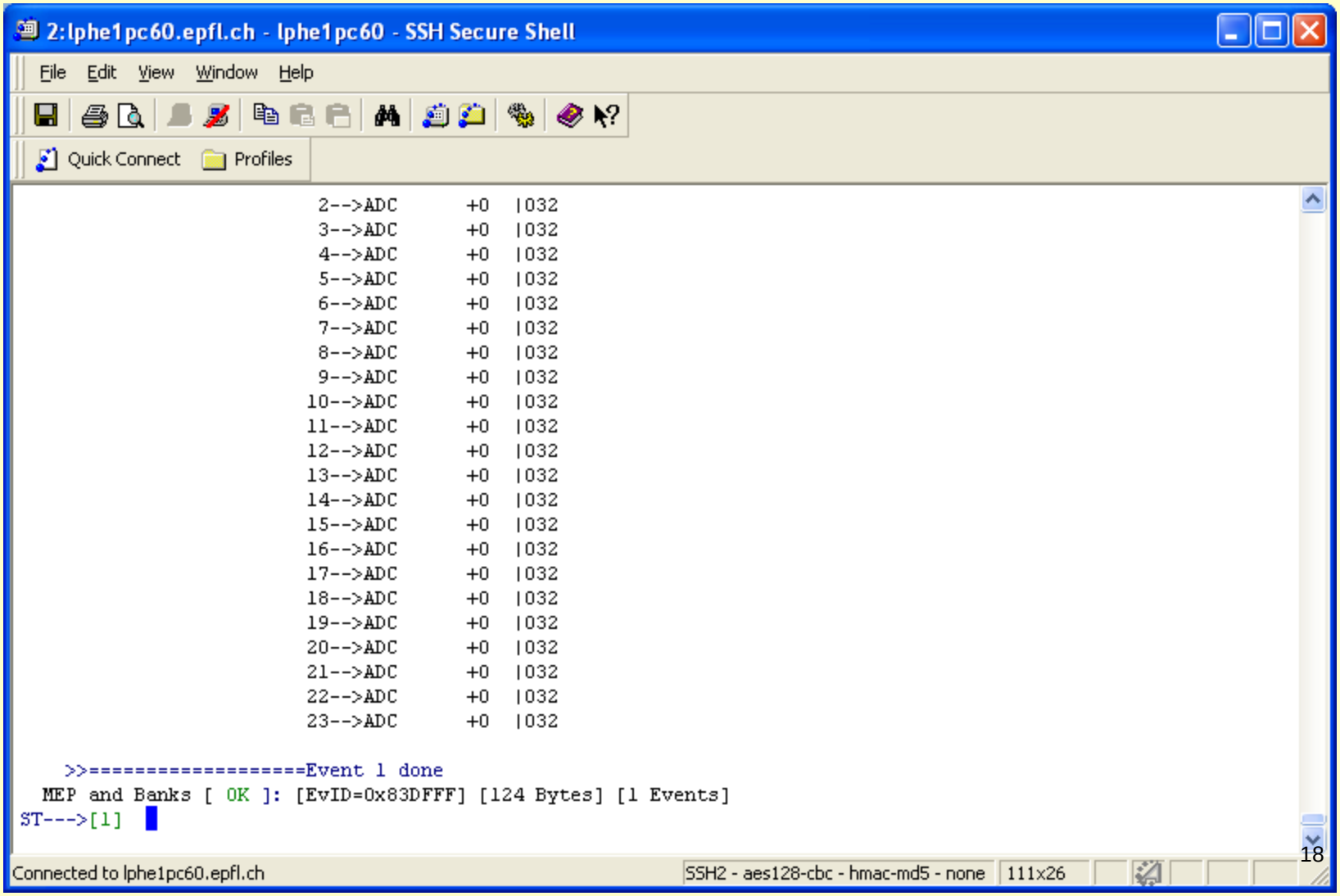
Running the selftest (running the test)



Running the selftest (results)



Running the selftest (results, use console_tell1, option D, l to monitor the event data)



Running the selftest (monitor processing counters option M, d in console_tell1)

2:lphe1pc60.epfl.ch - lphe1pc60 - SSH Secure Shell

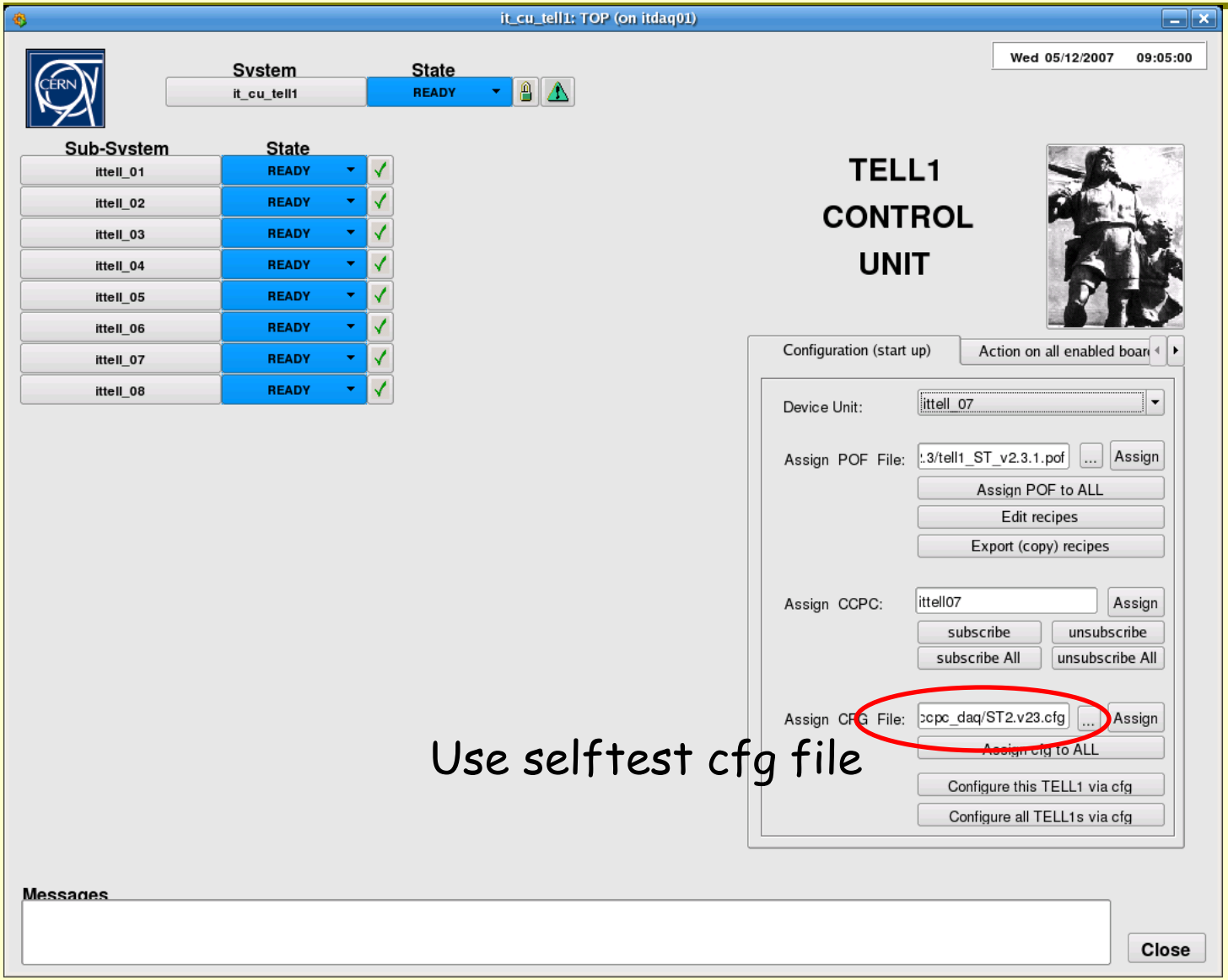
File Edit View Window Help

Quick Connect Profiles

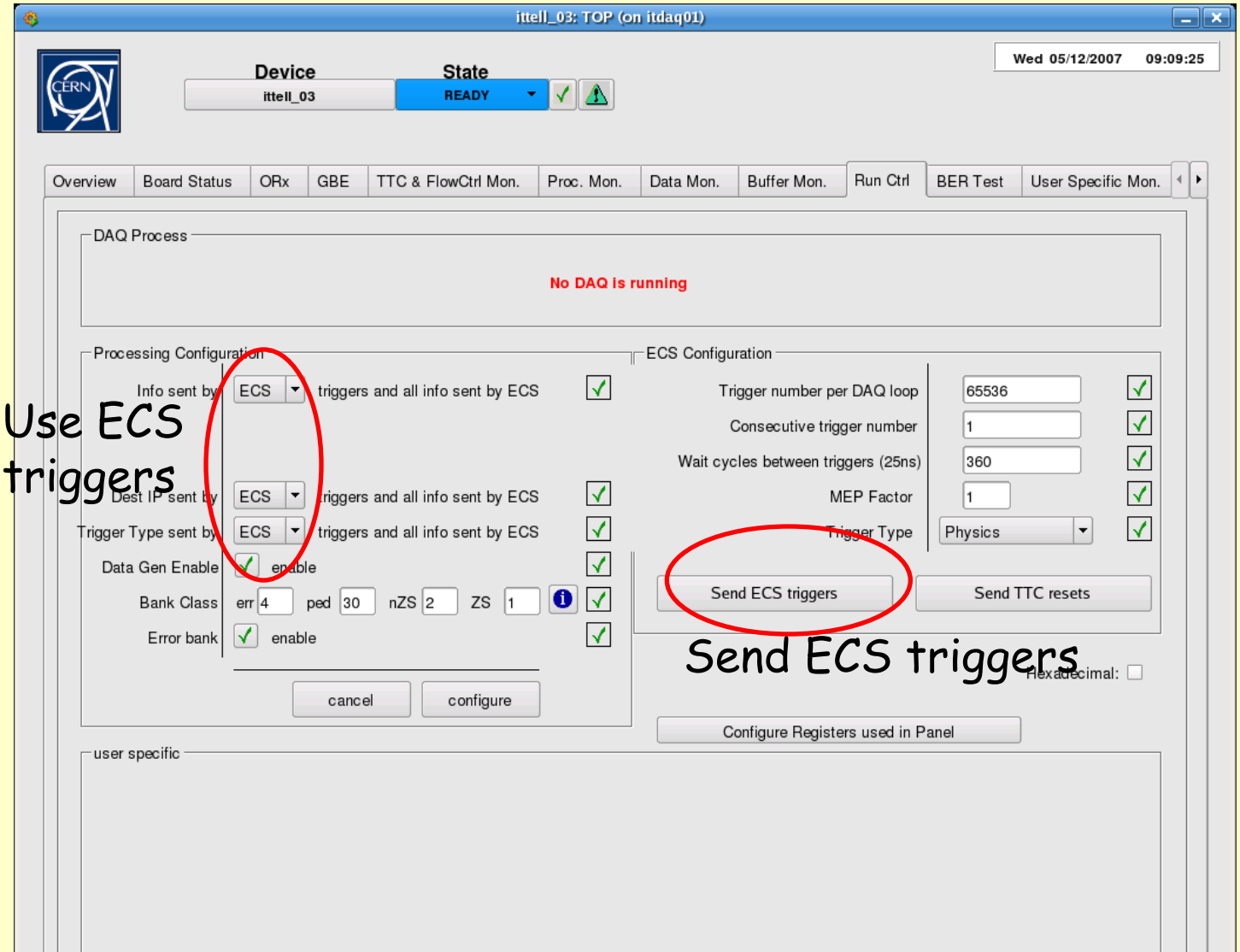
```
                        ESC : CCPC DAQ routine control
                          Q : Quit
                          ? : Show this manual
MONITOR--->[?]  d
-----Monitor registers in PP_FPGAs-----
VELO:  Detector Frames sent to          :
ST:    Evt incomplete (no data on o-link:  [       0] [       0] [       0] [       0]
Monitor Frames accepted in the PP_Rx    :  [    E000] [    E000] [    E000] [    E000]
Monitor Trigger_Request_number (PP->SL) :  [    E000] [    E000] [    E000] [    E000]
Monitor Trigger_Answer_number  (SL->PP) :  [    E000] [    E000] [    E000] [    E000]
Monitor Info Bank number in PP     (16b):  [    E000] [    E000] [    E000] [    E000]
Monitor Cluster Bank number in PP  (16b):  [    E000] [    E000] [    E000] [    E000]
Monitor ADC  Bank number in PP     (16b):  [    E000] [    E000] [    E000] [    E000]
Monitor RAW Bank number in PP       (8b):  [       0] [       0] [       0] [       0]
Monitor Pedestal Bank number in PP  (8b):  [       0] [       0] [       0] [       0]
Monitor number of events sent from PP   :  [  83E000] [  83E000] [  83E000] [  83E000]
Monitor number of ECS error in PP       :  [       0] [       7] [       F] [       0]
-----Monitor registers in SyncLink FPGA-----
Monitor number of FEM events counter    :  [  83E000]
Monitor number of banks from each PP    :  [   BA000] [   BA000] [   BA000] [   BA000]
Monitor max inFIFO usage for each PP    :  [       D] [       D] [       D] [       D]
Monitor number of events from each PP   :  [  83E000] [  83E000] [  83E000] [  83E000]
Monitor number of events assembled in SL:  [  83E000]
Monitor number of MEPs written into QDR :  [  83E000]
Monitor number of MEPs read from QDR    :  [  83E000]
Monitor number of valid triggers        :  [  83E000]
Monitor number of trig_info sent to PPs :  [  83E000]
Monitor number of trig_req from each PP :  [  83E000] [  83E000] [  83E000] [  83E000]
Frozen L0EvID        :  0x0083E000
Frozen BCnt          :  0x5C1
Monitor number of ECS error in SL       :  [       0]
MONITOR--->[d]
```

Connected to lphe1pc60.epfl.ch | SSH2 - aes128-cbc - hmac-md5 - none | 117x32

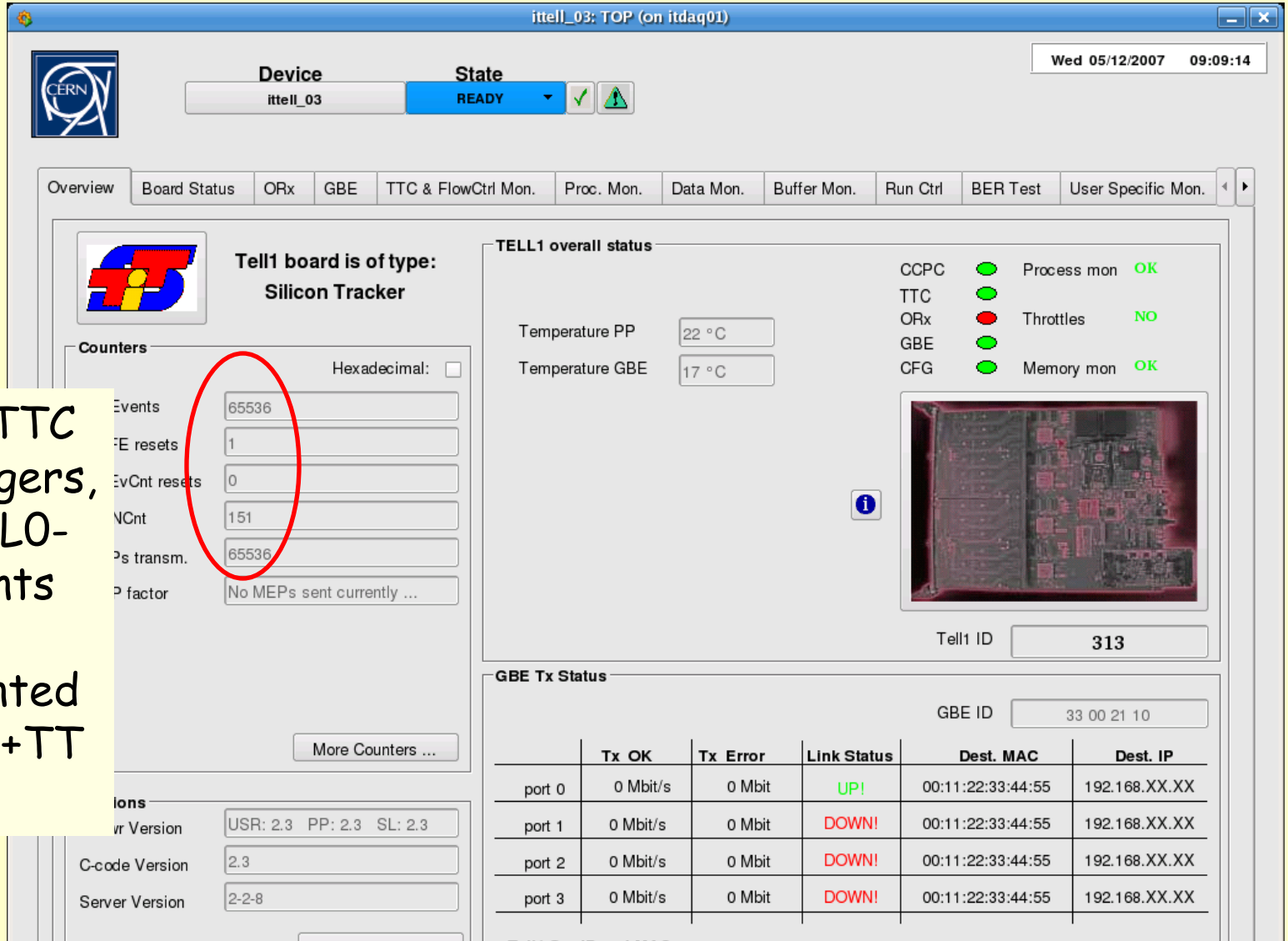
Selftest with PVSS



Sent ECS triggers to perform selftest

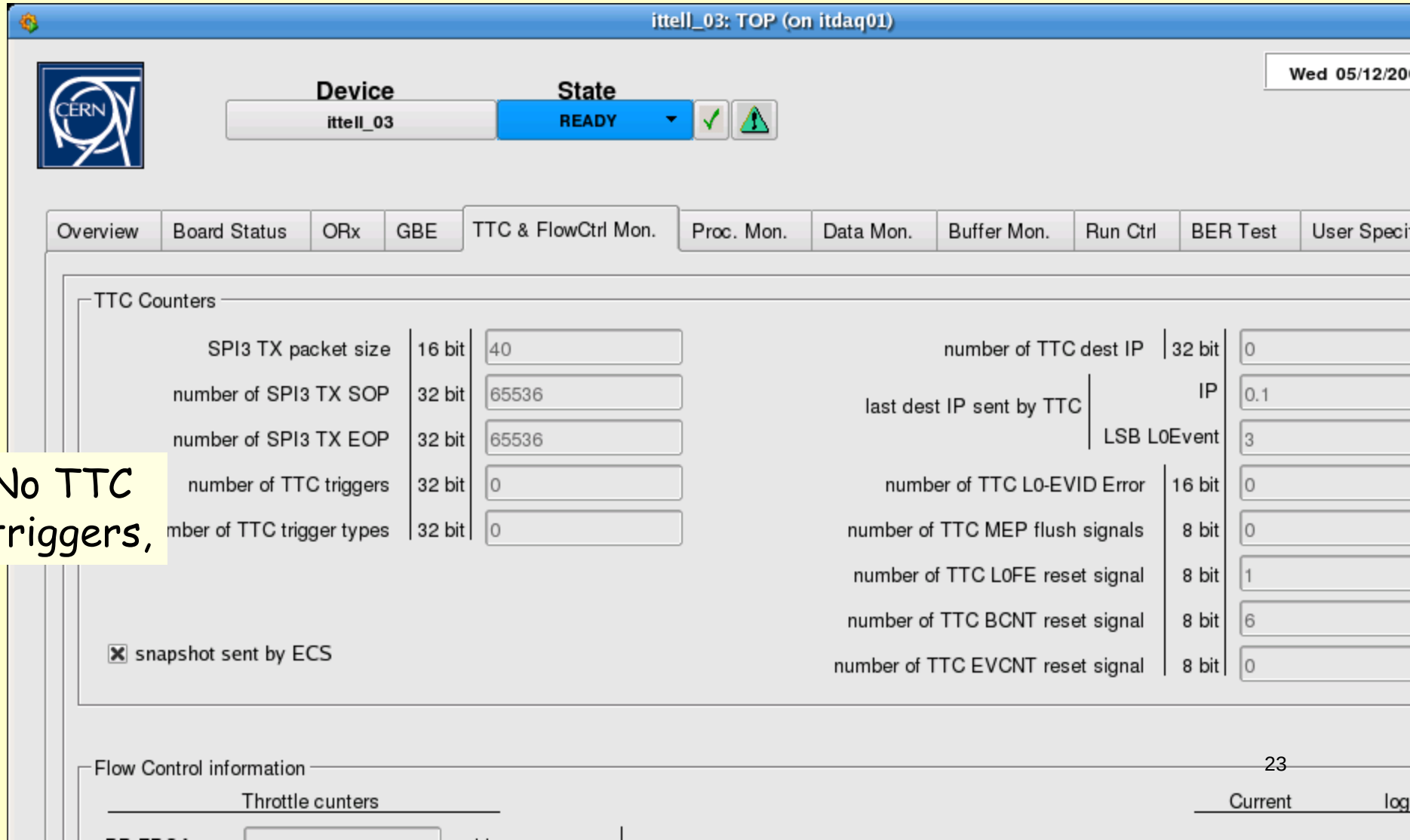


Monitor counters



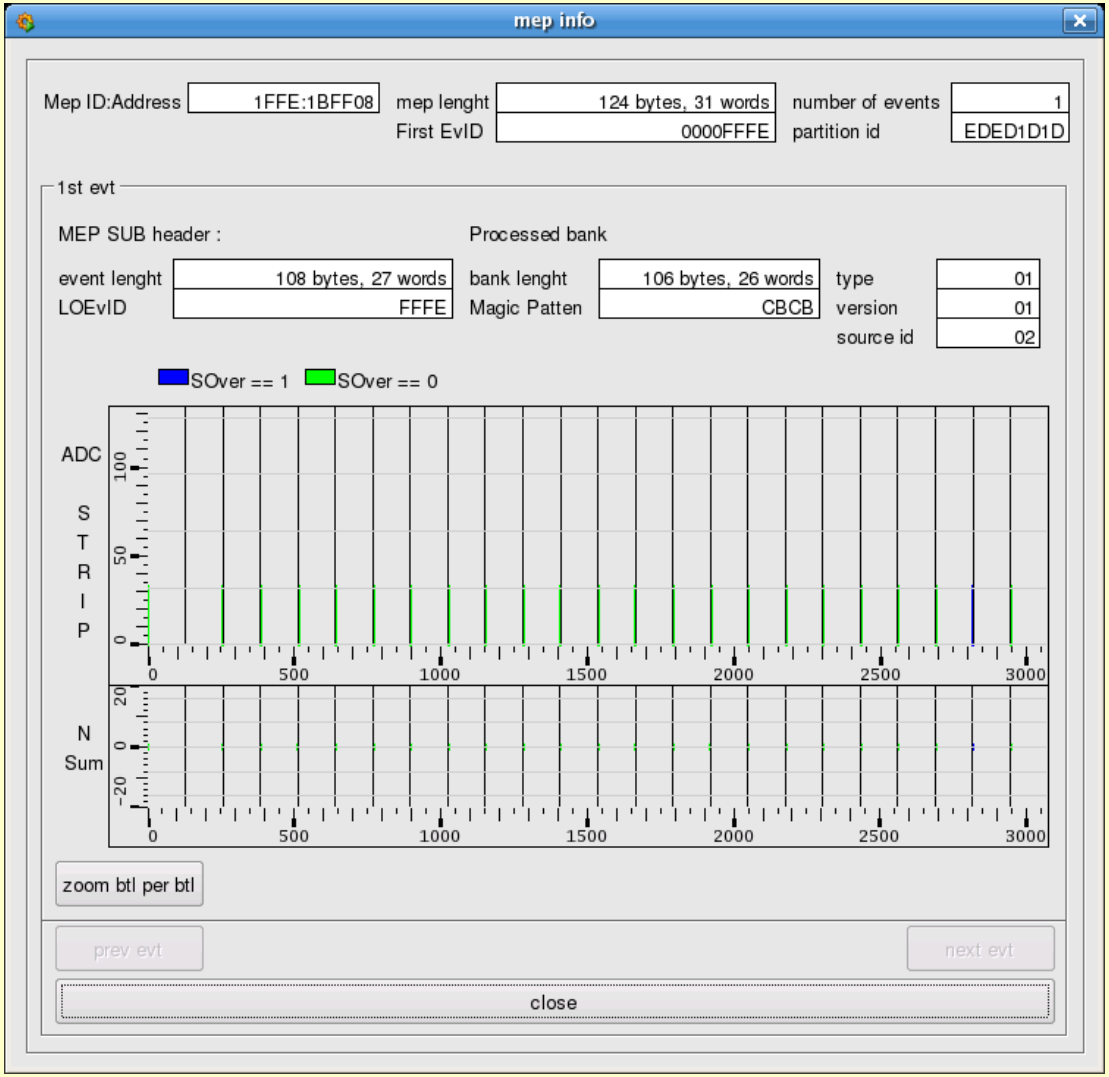
No TTC triggers, but L0-Events are counted ECS+TTC

Monitoring the selftest via PVSS



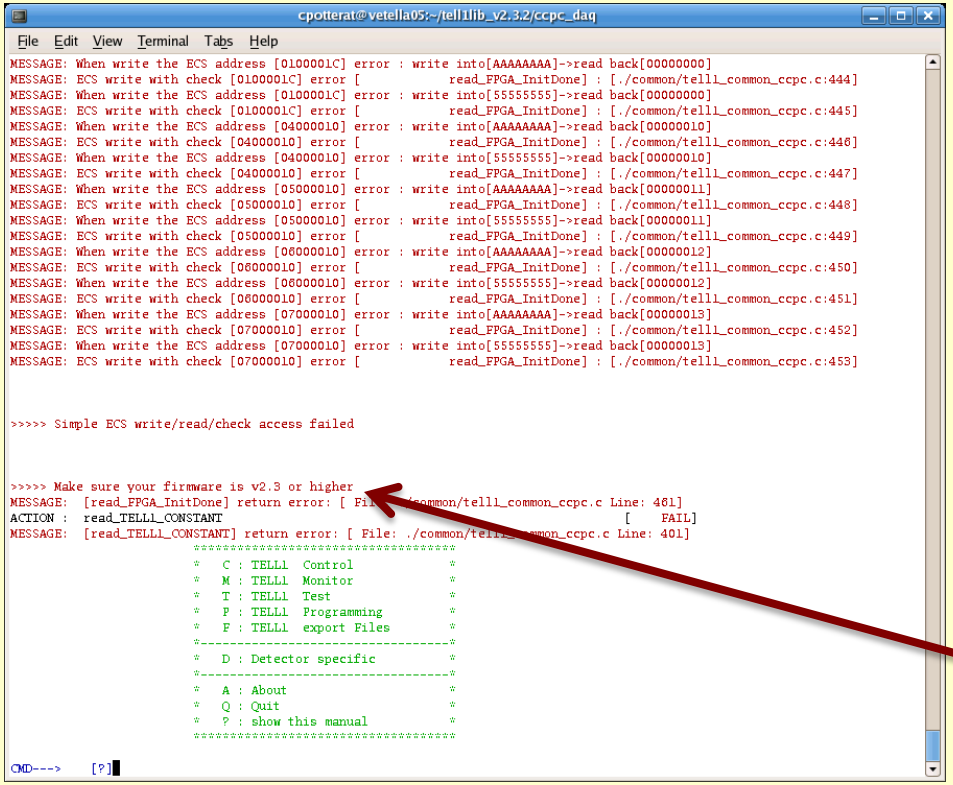
No TTC triggers,

Monitoring the event data



- Monitoring the event data is a way to parse the MEPs stored on the TELL1 in the MEP bugger

Using old firmware (smaller v2.3) with new software doesn't work!



- The error messages occur because a register is missing and therefore unsuccessfully write read checked
- The error doesn't quit the program as it was in previous versions
- I give a comment about v2.3 and higher

Bit error rate test with O-Rx cards

- There are two options to use the bit error rate test (BER) test, the command line tool and the PVSS
  - Command line tool, use

    daq_tell1 cfg_file -b

    use the console to reset the BER counters, C (control), option e
  - PVSS use the panel BER test
    - Reset counters will clean the error counters and the data counters
    - The error counters saturate (get to 0xFFFFFF) in no time (0.1s) so, if you have saturated counters it's a bad link!