



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 board.

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 provided
- 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 ' 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 now 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 ctrl PC yum update. This is a way to implement a hardware test function
→ To provide a hardware test function.

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)

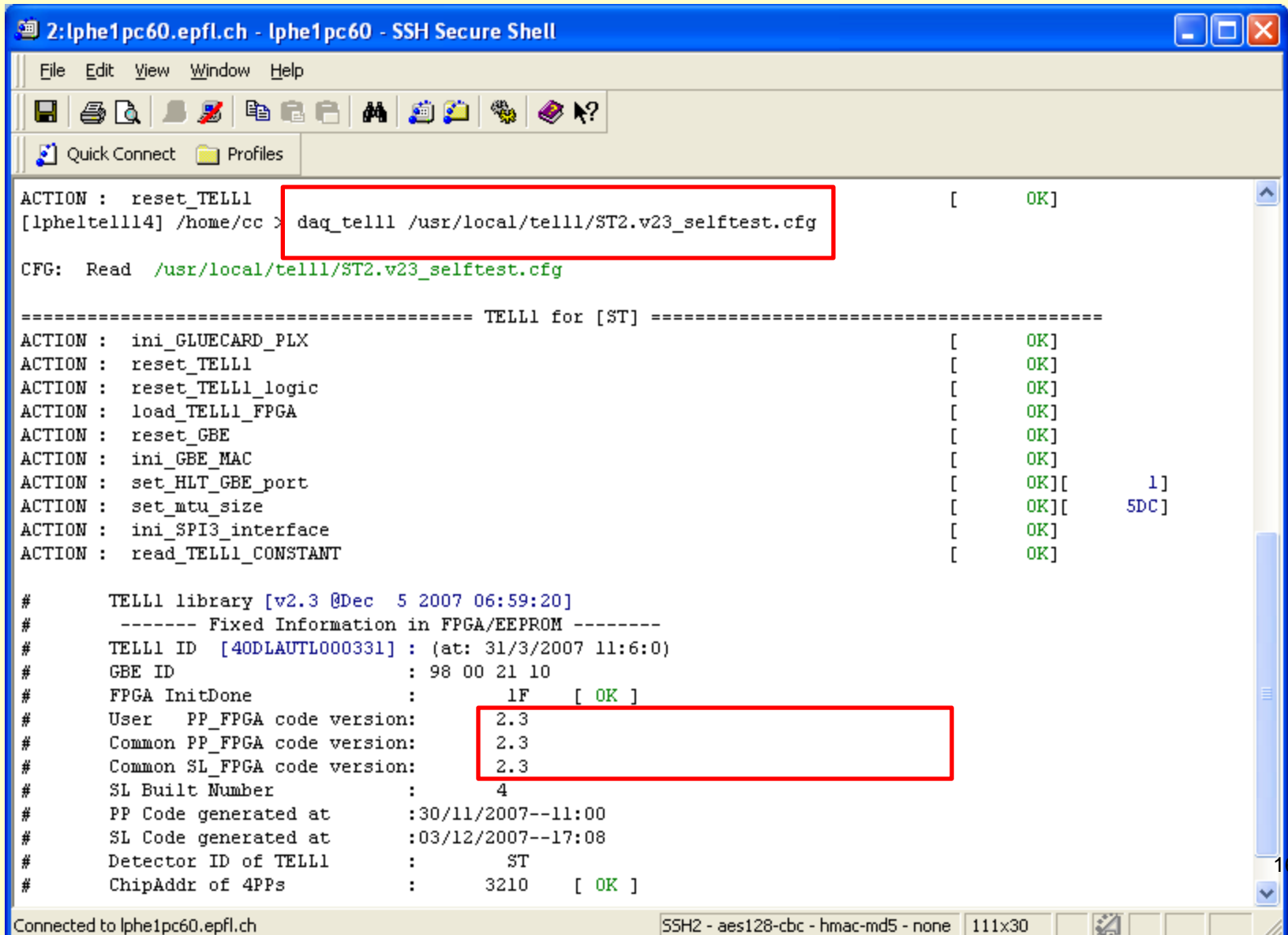
1. Make an upgrade of the tell1lib installed on your ctrlPC eg. itdaq01 for IT
2. Program the FPGAs with the required version, a copy of the .pof files is always distributed and placed in:
[/usr/local/tell1/](#)
3. Run the selftest using the .cfg files also available in [/usr/local/tell1/](#)

Running the selftest (version check)

```
# TELL1 library [v2.3 @Dec 5 2007 06:59:20]
# ----- Fixed Information in FPGA/EEPROM -----
# TELL1 ID [40DLAUTL000331] : (at: 31/3/2007 11:6:0)
# GBE ID : 98 00 21 10
# FPGA InitDone : 1F [ OK ]
# User PP_FPGA code version: 2.3
# Common PP_FPGA code version: 2.3
# Common SL_FPGA code version: 2.3
# SL Built Number : 4
# PP Code generated at :30/11/2007--11:00
# SL Code generated at :03/12/2007--17:08
# Detector ID of TELL1 : ST
# ChipAddr of 4PPs : 3210 [ OK ]

[lphe1tell14] /home/cc >
```

Running the selftest (running the test)



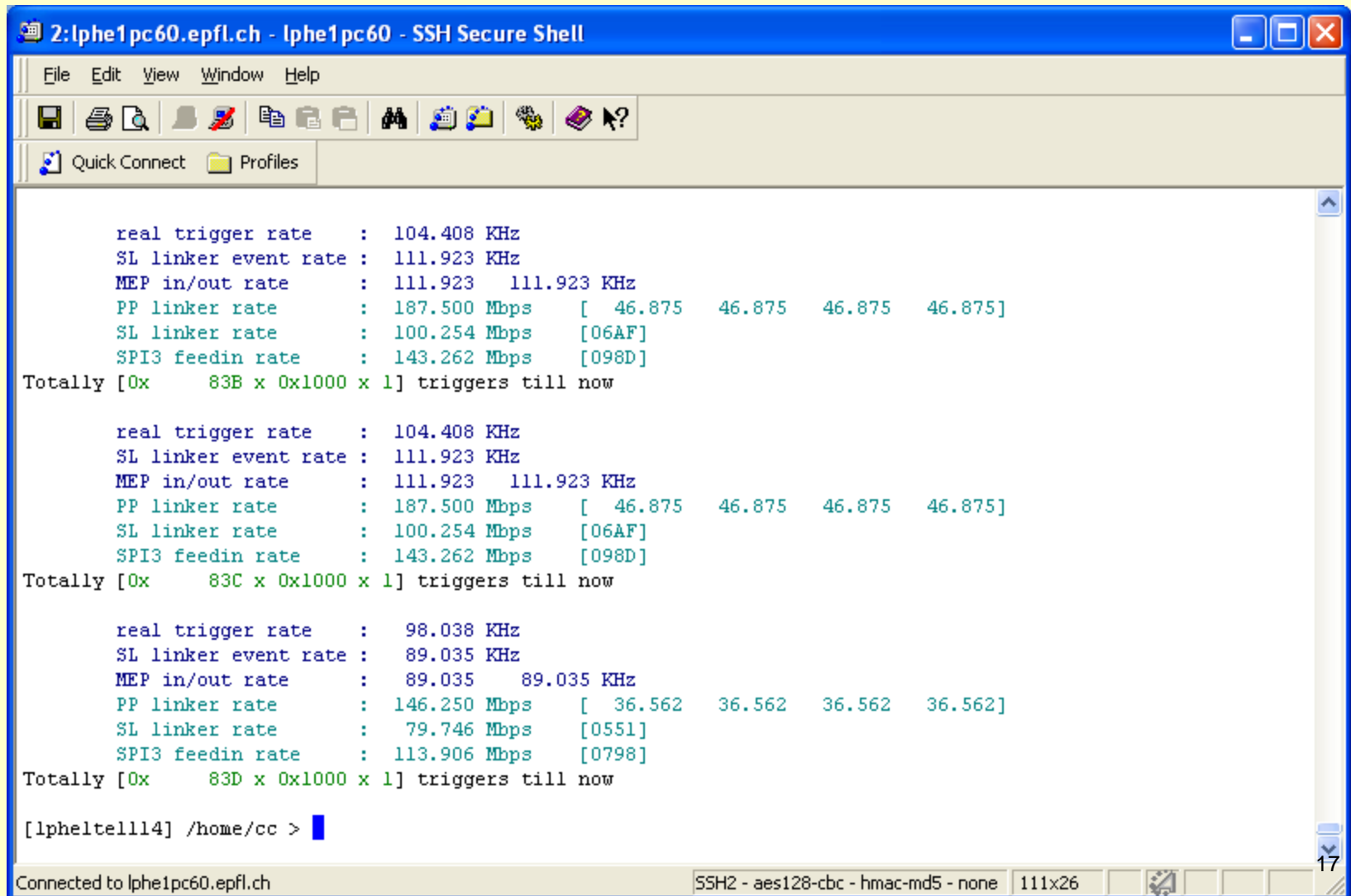
```
2:lphe1pc60.epfl.ch - lphe1pc60 - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
ACTION : reset_TELL1 [ OK ]
[lpheltell14] /home/cc > daq_tell1 /usr/local/tell1/ST2.v23_selftest.cfg
CFG: Read /usr/local/tell1/ST2.v23_selftest.cfg

===== TELL1 for [ST] =====
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 : set_HLT_GBE_port [ OK][ 1]
ACTION : set_mtu_size [ OK][ SDC]
ACTION : ini_SPI3_interface [ OK ]
ACTION : read_TELL1_CONSTANT [ OK ]

# TELL1 library [v2.3 @Dec 5 2007 06:59:20]
# ----- Fixed Information in FPGA/EEPROM -----
# TELL1 ID [40DLAULT000331] : (at: 31/3/2007 11:6:0)
# GBE ID : 98 00 21 10
# FPGA InitDone : 1F [ OK ]
# User PP_FPGA code version: 2.3
# Common PP_FPGA code version: 2.3
# Common SL_FPGA code version: 2.3
# SL Built Number : 4
# PP Code generated at :30/11/2007--11:00
# SL Code generated at :03/12/2007--17:08
# Detector ID of TELL1 : ST
# ChipAddr of 4PPs : 3210 [ OK ]

Connected to lphe1pc60.epfl.ch SSH2 - aes128-cbc - hmac-md5 - none 111x30
```


Running the selftest (results)



The screenshot shows an SSH terminal window titled "2:lphe1pc60.epfl.ch - lphe1pc60 - SSH Secure Shell". The terminal displays the output of a selftest command, showing performance metrics for three different triggers. The window includes a menu bar (File, Edit, View, Window, Help), a toolbar with various icons, and a status bar at the bottom indicating the connection details and window size.

```
real trigger rate      : 104.408 KHz
SL linker event rate  : 111.923 KHz
MEP in/out rate       : 111.923 111.923 KHz
PP linker rate        : 187.500 Mbps [ 46.875 46.875 46.875 46.875]
SL linker rate        : 100.254 Mbps [06AF]
SPI3 feedin rate     : 143.262 Mbps [098D]
Totally [0x 83B x 0x1000 x 1] triggers till now

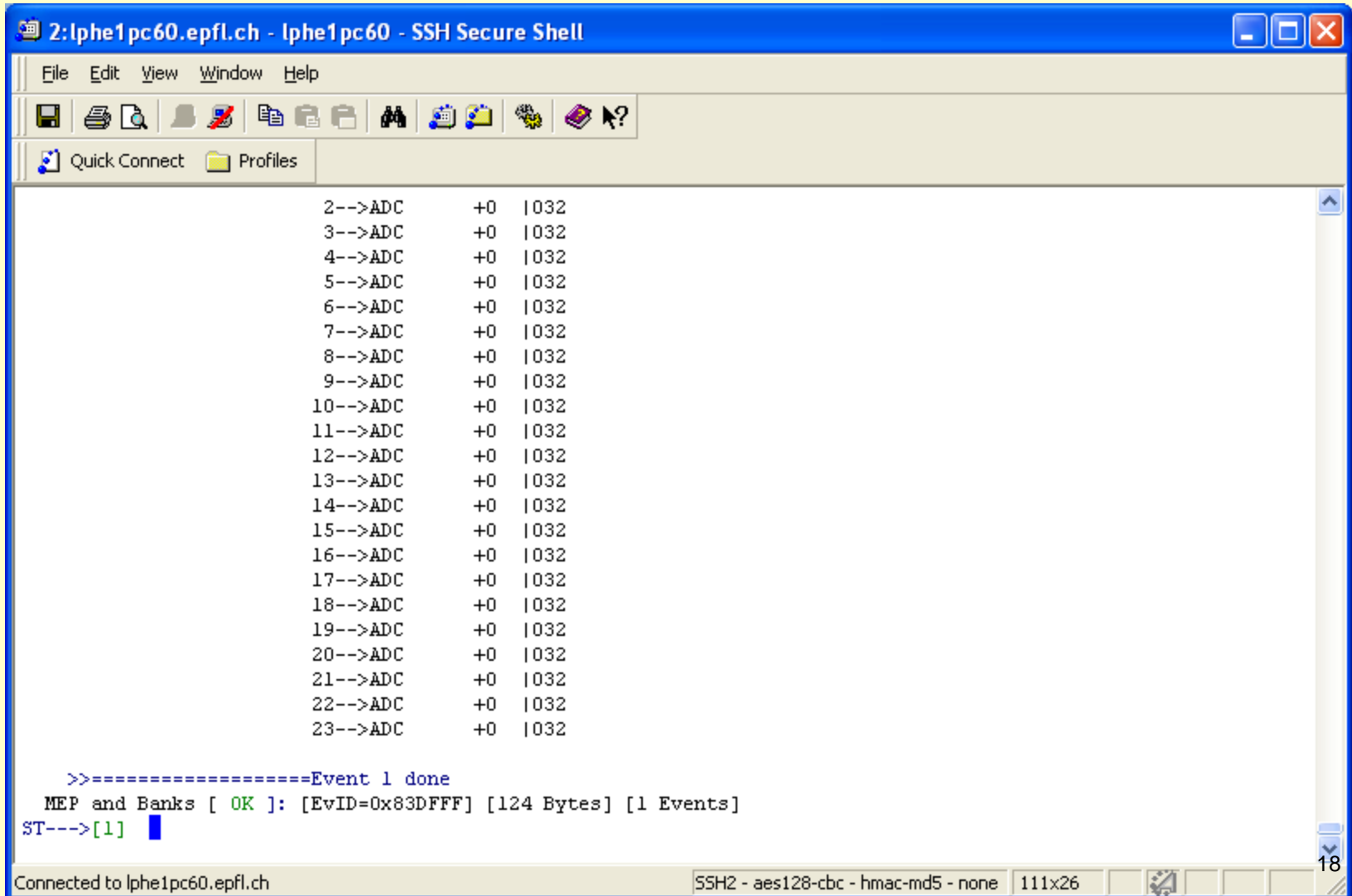
real trigger rate      : 104.408 KHz
SL linker event rate  : 111.923 KHz
MEP in/out rate       : 111.923 111.923 KHz
PP linker rate        : 187.500 Mbps [ 46.875 46.875 46.875 46.875]
SL linker rate        : 100.254 Mbps [06AF]
SPI3 feedin rate     : 143.262 Mbps [098D]
Totally [0x 83C x 0x1000 x 1] triggers till now

real trigger rate      : 98.038 KHz
SL linker event rate  : 89.035 KHz
MEP in/out rate       : 89.035 89.035 KHz
PP linker rate        : 146.250 Mbps [ 36.562 36.562 36.562 36.562]
SL linker rate        : 79.746 Mbps [0551]
SPI3 feedin rate     : 113.906 Mbps [0798]
Totally [0x 83D x 0x1000 x 1] triggers till now

[lpheltell14] /home/cc >
```

Connected to lphe1pc60.epfl.ch SSH2 - aes128-cbc - hmac-md5 - none 111x26

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



The screenshot shows an SSH terminal window titled "2:lphe1pc60.epfl.ch - lphe1pc60 - SSH Secure Shell". The terminal displays the output of a selftest for 23 ADC channels. Each channel (numbered 2 to 23) shows a result of "+0 |032". Below the list, the terminal indicates "Event 1 done" and provides details for the event: "MEP and Banks [OK]: [EvID=0x83DFFF] [124 Bytes] [1 Events]". The prompt "ST-->[1]" is followed by a cursor. The terminal status bar at the bottom shows "Connected to lphe1pc60.epfl.ch", "SSH2 - aes128-cbc - hmac-md5 - none", and "111x26".

```
2-->ADC      +0 |032
3-->ADC      +0 |032
4-->ADC      +0 |032
5-->ADC      +0 |032
6-->ADC      +0 |032
7-->ADC      +0 |032
8-->ADC      +0 |032
9-->ADC      +0 |032
10-->ADC     +0 |032
11-->ADC     +0 |032
12-->ADC     +0 |032
13-->ADC     +0 |032
14-->ADC     +0 |032
15-->ADC     +0 |032
16-->ADC     +0 |032
17-->ADC     +0 |032
18-->ADC     +0 |032
19-->ADC     +0 |032
20-->ADC     +0 |032
21-->ADC     +0 |032
22-->ADC     +0 |032
23-->ADC     +0 |032

>>=====Event 1 done
MEP and Banks [ OK ]: [EvID=0x83DFFF] [124 Bytes] [1 Events]
ST-->[1] █
```

Connected to lphe1pc60.epfl.ch SSH2 - aes128-cbc - hmac-md5 - none 111x26

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 LOEVID : 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

The screenshot displays the TELL1 CONTROL UNIT web interface. At the top, the system is identified as 'it_cu_tell1' and is in a 'READY' state. A table lists eight sub-systems, all of which are also in a 'READY' state. The main configuration area is titled 'Configuration (start up)' and includes options for assigning a Device Unit, POF File, CCPC, and CFG File. The 'Assign CFG File' field is highlighted with a red circle, and the text 'Use selftest cfg file' is written below it. The interface also features a CERN logo, a date and time stamp, and a 'Messages' section at the bottom.

Sub-System	State
ittell_01	READY
ittell_02	READY
ittell_03	READY
ittell_04	READY
ittell_05	READY
ittell_06	READY
ittell_07	READY
ittell_08	READY

Configuration (start up) Action on all enabled boards

Device Unit: ittell_07

Assign POF File: :3/tell1_ST_v2.3.1.pof Assign

Assign POF to ALL

Edit recipes

Export (copy) recipes

Assign CCPC: ittell07 Assign

subscribe unsubscribe

subscribe All unsubscribe All

Assign CFG File: :cpc_daq/ST2.v23.cfg Assign

Assign Cfg to ALL

Configure this TELL1 via cfg

Configure all TELL1s via cfg

Messages

Close

Use selftest cfg file

Sent ECS triggers to perform selftest

The screenshot shows a control interface for the 'itell_03' device. At the top, the device name 'itell_03' and its state 'READY' are displayed. A navigation bar includes tabs for Overview, Board Status, ORx, GBE, TTC & FlowCtrl Mon., Proc. Mon., Data Mon., Buffer Mon., Run Ctrl, BER Test, and User Specific Mon. The 'Run Ctrl' tab is active.

The 'DAQ Process' section indicates 'No DAQ Is running'. The 'Processing Configuration' section has several dropdown menus set to 'ECS', which are circled in red. The 'ECS Configuration' section includes fields for Trigger number per DAQ loop (65536), Consecutive trigger number (1), Wait cycles between triggers (360), MEP Factor (1), and Trigger Type (Physics). A 'Send ECS triggers' button is circled in red.

Handwritten text 'Use ECS triggers' is on the left, and 'Send ECS triggers' is at the bottom. A 'Hexadecimal' checkbox is also visible.

Device: itell_03
State: READY
Date/Time: Wed 05/12/2007 09:09:25

Navigation: Overview | Board Status | ORx | GBE | TTC & FlowCtrl Mon. | Proc. Mon. | Data Mon. | Buffer Mon. | Run Ctrl | BER Test | User Specific Mon.

DAQ Process: No DAQ Is running

Processing Configuration:
Info sent by: ECS triggers and all info sent by ECS ✓
Dest IP sent by: ECS triggers and all info sent by ECS ✓
Trigger Type sent by: ECS triggers and all info sent by ECS ✓
Data Gen Enable: enable ✓
Bank Class: err 4 ped 30 nZS 2 ZS 1 ✓
Error bank: enable ✓

ECS Configuration:
Trigger number per DAQ loop: 65536 ✓
Consecutive trigger number: 1 ✓
Wait cycles between triggers (25ns): 360 ✓
MEP Factor: 1 ✓
Trigger Type: Physics ✓

Buttons: Send ECS triggers, Send TTC resets, Configure Registers used in Panel

Handwritten text: Use ECS triggers, Send ECS triggers

Hexadecimal:

Monitor counters

itell_03: TOP (on itdaq01) Wed 05/12/2007 09:09:14

Device itell_03 **State** READY ✔ ⚠

Overview Board Status ORx GBE TTC & FlowCtrl Mon. Proc. Mon. Data Mon. Buffer Mon. Run Ctrl BER Test User Specific Mon.

Tell1 board is of type: Silicon Tracker

Counters Hexadecimal:

Events: 65536

TE resets: 1

EvCnt resets: 0

NCnt: 151

Ps transm.: 65536

P factor: No MEPs sent currently ...

TELL1 overall status

Temperature PP: 22 °C

Temperature GBE: 17 °C

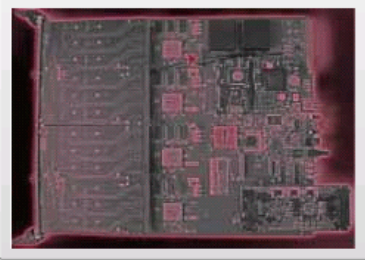
CCPC ● Process mon OK

TTC ●

ORx ● Throttles NO

GBE ●

CFG ● Memory mon OK



Tell1 ID: 313

GBE Tx Status GBE ID: 33 00 21 10

	Tx OK	Tx Error	Link Status	Dest. MAC	Dest. IP
port 0	0 Mbit/s	0 Mbit	UP!	00:11:22:33:44:55	192.168.XX.XX
port 1	0 Mbit/s	0 Mbit	DOWN!	00:11:22:33:44:55	192.168.XX.XX
port 2	0 Mbit/s	0 Mbit	DOWN!	00:11:22:33:44:55	192.168.XX.XX
port 3	0 Mbit/s	0 Mbit	DOWN!	00:11:22:33:44:55	192.168.XX.XX

More Counters ...

ions

ur Version: USR: 2.3 PP: 2.3 SL: 2.3

C-code Version: 2.3

Server Version: 2-2-8

No TTC triggers, but LO-Events are counted ECS+TT C

Monitoring the selftest via PVSS

ittell_03: TOP (on itdaq01)

Wed 05/12/20

Device
ittell_03

State
READY

Overview | Board Status | ORx | GBE | **TTC & FlowCtrl Mon.** | Proc. Mon. | Data Mon. | Buffer Mon. | Run Ctrl | BER Test | User Speci

TTC Counters

SPI3 TX packet size	16 bit	40	number of TTC dest IP	32 bit	0
number of SPI3 TX SOP	32 bit	65536	last dest IP sent by TTC	IP	0.1
number of SPI3 TX EOP	32 bit	65536	LSB L0Event		3
number of TTC triggers	32 bit	0	number of TTC L0-EVID Error	16 bit	0
number of TTC trigger types	32 bit	0	number of TTC MEP flush signals	8 bit	0
			number of TTC L0FE reset signal	8 bit	1
			number of TTC BCNT reset signal	8 bit	6
			number of TTC EVCNT reset signal	8 bit	0

snapshot sent by ECS

Flow Control information

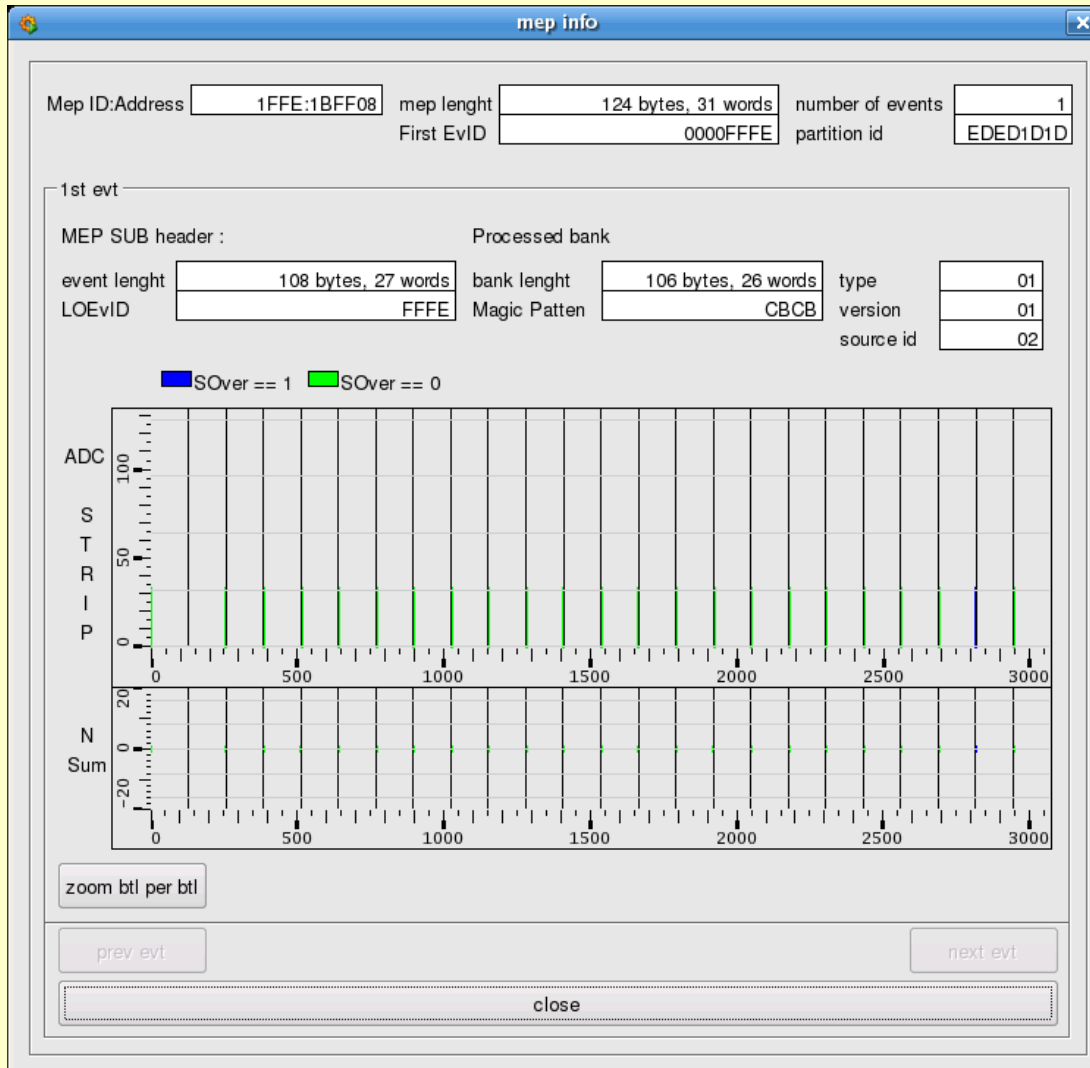
Throttle counters

Current log

23

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

```
cpotterat@vetella05:~/jelllib_v2.3.2/ccpc_daq
File Edit View Terminal Tabs Help
MESSAGE: When write the ECS address [0100001C] error : write into[AAAAAAAA]->read back[00000000]
MESSAGE: ECS write with check [0100001C] error [ read_FPGA_InitDone] : [./common/tell1_common_ccpc.c:444]
MESSAGE: When write the ECS address [0100001C] error : write into[55555555]->read back[00000000]
MESSAGE: ECS write with check [0100001C] error [ read_FPGA_InitDone] : [./common/tell1_common_ccpc.c:445]
MESSAGE: When write the ECS address [04000010] error : write into[AAAAAAAA]->read back[00000010]
MESSAGE: ECS write with check [04000010] error [ read_FPGA_InitDone] : [./common/tell1_common_ccpc.c:446]
MESSAGE: When write the ECS address [04000010] error : write into[55555555]->read back[00000010]
MESSAGE: ECS write with check [04000010] error [ read_FPGA_InitDone] : [./common/tell1_common_ccpc.c:447]
MESSAGE: When write the ECS address [05000010] error : write into[AAAAAAAA]->read back[00000011]
MESSAGE: ECS write with check [05000010] error [ read_FPGA_InitDone] : [./common/tell1_common_ccpc.c:448]
MESSAGE: When write the ECS address [05000010] error : write into[55555555]->read back[00000011]
MESSAGE: ECS write with check [05000010] error [ read_FPGA_InitDone] : [./common/tell1_common_ccpc.c:449]
MESSAGE: When write the ECS address [06000010] error : write into[AAAAAAAA]->read back[00000012]
MESSAGE: ECS write with check [06000010] error [ read_FPGA_InitDone] : [./common/tell1_common_ccpc.c:450]
MESSAGE: When write the ECS address [06000010] error : write into[55555555]->read back[00000012]
MESSAGE: ECS write with check [06000010] error [ read_FPGA_InitDone] : [./common/tell1_common_ccpc.c:451]
MESSAGE: When write the ECS address [07000010] error : write into[AAAAAAAA]->read back[00000013]
MESSAGE: ECS write with check [07000010] error [ read_FPGA_InitDone] : [./common/tell1_common_ccpc.c:452]
MESSAGE: When write the ECS address [07000010] error : write into[55555555]->read back[00000013]
MESSAGE: ECS write with check [07000010] error [ read_FPGA_InitDone] : [./common/tell1_common_ccpc.c:453]

>>>> Simple ECS write/read/check access failed

>>>> Make sure your firmware is v2.3 or higher
MESSAGE: [read_FPGA_InitDone] return error: [ File: ./common/tell1_common_ccpc.c Line: 461]
ACTION : read_TELL1_CONSTANT [ FAIL]
MESSAGE: [read_TELL1_CONSTANT] return error: [ File: ./common/tell1_common_ccpc.c Line: 401]

*****
* 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--> [?]
```

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 0xFFFFFFFF) in no time (0.1s) so, if you have saturated counters it's a bad link!