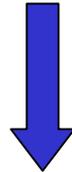


COMMON GUI FOR BOARD DIAGNOSTIC

- Remind: we want a graphical interface to run test programs (quick in-situ test of SVT boards, no board or cable reshuffling)
The GUI should be also configurable to test boards in the test stand.
- Since my last talk at the Tuesday SVT meeting: specs for board test programs (thanks to Stefano and Simone for their suggestions)
- I'll use HB board as example (of course this is preliminary and to be discussed among all of us)

- 2 test programs:
 - quick test: VME registers and few more things (execution time ~ seconds)
 - extensive test: RAMs, spy buffers, injects data and reads back results (execution time ~ minutes)
- Test programs like random-test (long execution time to be meaningful) probably are not suitable for this package (comments?)

- The GUI will allow to run the **quick, extensive** or **quick+extensive** tests
- Test programs will be self-contained, separate C functions to be included in SVTVME library
- Test programs should generate internally the data to be sent to the board under test, the lookup tables and also the data to check the results.
- The GUI will be a Linux executable, but it would be quicker to run test programs directly on the crate CPU. SVTVME library provides both Linux and VxWorks support, so no problem for the test programs, but we need a transport mechanism (corba/robin ?)



I suggest a first version where all test programs run from the host and then a second version with test programs running from the crate CPU

- Test programs will dump a logfile: I suggest the format defined by Luciano for AMboard:

```

0100 11/00/07/06/15/30/34 am_test_data Version 1.09 logfiles/amd11000706153034.log
0105 11/00/07/06/15/30/34 CPU = <vmesvt2> slots = mrg:03 ams:08 am:10
0115 11/00/07/06/15/30/34 serial numbers = mrg:01 ams:01 am:11
0110 11/00/07/06/15/30/34 Amboard ID = <0031Amboard V2.0>
0860 11/00/07/06/15/30/43 Plug 00:478 all tests OK
0860 11/00/07/06/15/30/43 Plug 01:480 all tests OK
0860 11/00/07/06/15/30/43 Plug 02:473 all tests OK
0860 11/00/07/06/15/30/43 Plug 03:481 all tests OK
0860 11/00/07/06/15/30/43 Plug 04:474 all tests OK
0860 11/00/07/06/15/30/43 Plug 05:484 all tests OK
0860 11/00/07/06/15/30/43 Plug 06:472 all tests OK
0860 11/00/07/06/15/30/43 Plug 07:485 all tests OK
0860 11/00/07/06/15/30/43 Plug 08:477 all tests OK
0860 11/00/07/06/15/30/43 Plug 09:483 all tests OK
0860 11/00/07/06/15/30/43 Plug 10:476 all tests OK
0860 11/00/07/06/15/30/43 Plug 11:482 all tests OK
0860 11/00/07/06/15/30/43 Plug 12:479 all tests OK
0860 11/00/07/06/15/30/43 Plug 13:591 all tests OK
0860 11/00/07/06/15/30/43 Plug 14:475 all tests OK
0860 11/00/07/06/15/30/43 Plug 15:592 all tests OK
0900 11/00/07/06/15/30/43 ----- TEST PASSED OK -----

```

Logfile file name:

hb<slot>_<crate>_<date><time>.log
 e.g. :
 hb12_b0svt01_02052001134510.log



Test program function prototype:

```
return_value          => 0 = all tests OK, >0 = error, same number of logfile testID (first error)
= hb_test1(           => test function name, hb_test2 hb_test3 etc if more test programs
char *<cpu>           => cpu crate for the board under test
,int <slot>           => slot for the board under test
,int <conf>           => configuration mode : 0 standard SVT, 1 read configuration from conf_file (1)
,char* <conf_file>    => configuration file (2)
,char*<error_string>  => error string from test program, shown in the GUI, the same from the log file
,int <error_string_len>) => max length for error string
```

- (1) Test programs will have the standard SVT board configuration hardwired in the code.
e.g. hb will know that his standard slot is 12 (or 21), that will receive test data from merger in slot 7 (or 16) and from ams in slot 8 (or 17) and that data should be read back from TF in crate b0svt06, slot <n> depending from the hb phi sector
- (2) The same information should be readable from a file, for testing custom configurations or boards in the test stand. The GUI will provide a window to edit this kind of files. The format of this file has to be discussed with test programs providers. Do we have to think a file format suitable for all the boards?

Who writes test programs:

- Spy control: Franco
- Hit buffer: Stefano
- XTFA&B: Simone
- Merger + XTFC: Annamaria
- HF: Bill
- TF: Un-Ki
- AMS: ...
- AM: