

TTD Miscellaneous

Mikihiko Nakao (KEK)

`mikihiko.nakao@kek.jp`

2019.8.27

Trigger/DAQ workshop at Yonsei

Overflow items

- Run control related things
- Software management
- Data format
- TTD for DAQ-upgrade

PAUSE handling

- **CDC HV trip → PAUSE (only) TTD**
 - RC should be updated to send PAUSE to TTD (then RESUME)
 - pocket_ttd is able to PAUSE and RESUME
 - ttctrl is not able to handle yet
- **No change in run condition expected**
 - No change in run number, subrun number
- **subrun number?**
 - PAUSE request may have an option to increment subrun number
 - Otherwise currently subrun number can be updated only by regft

Error handling

- **Error outside of TTD**

- TTD should receive STOP request
- Maybe with a different option from normal STOP by shifter

- **Error found by TTD: checking one error bit**

- if the bit is high, one can run “ttaddr” function for more detailed report

- **Busy: checking the staying busy bit**

- if the bit is high for long time, pocket_ttd can locally pause the trigger, then run “ttaddr” function to identify the busy source

- **pocket_ttd**

- Running external script for monitor right now
- Should be done by pocket_ttd, and all other external scripts should be avoided

Run end report

- **TTD should generate run-end report**
 - A natural extension of the error reporting
 - Should also do at normal run stop

TTD software structure

- **ttctrld** — daq_slc style program to talk to RC
- **pocket_ttd** — main readout program
originally for beam-test with PocketDAQ setup
- **ftprogs** — ftsw library and command line programs
- **statft** — one of ftsprogs, widely used to monitor FTSW
- **ttaddr** — ttd connection setting and status collection
- (ttinlv — process to receive injection veto from GDL)
- **scripts** — jtag programming, monitoring FTSW, ...
- no GUI for TTD ...

git repository

● **daq_slc repo**

- ttctrld is the only program in daq_slc for TTD
- ttctrld does FTSW register access, which should be avoided
- ttctrld was out of daq_slc sync till July
- now all changes are committed and pushed to git

● **ttd repo**

- Rest of the programs are in ttd repo (recently separated from ftsw repo)
- Currently writeable only by me
- pocket_ttd is in ttd11:~b2daq/ttd/, rest are in /usr/local/bin
- Some of the scripts are also added to git, some are not yet

● **daq_restart repo**

- start-ttd11.sh to start missing ttd11 process

run-time environment

● account on ttd11

- NSM2 processes are running under b2daq account
- Other programs can be run by anybody on ttd11 (**dangerous**)

● daq_slc

- cvmfs is mounted on ttd11, now run time is using cvmfs
- a bit of struggle: cvmfs needs a few run-time writeable directories

● online parameters

- No place (file/DB) to remember the status of include/exclude, dummy trigger
- No particular parameters, except those given in command line option
- Crucial initial values are hardcoded in program / firmware
- Every connection configuration is hardcoded in ttaddr

Code update 1

- **pocket_ttd reusing ftprogs**

- regft, trigft, statft, ttaddr codes are called
- messages into char string (instead of stdout) and then to log file

- **statft code management**

- ftsw register map has been updated several times
- statft checks the firmware version and show differently
- now we can forget older versions and keep only latest ft3m and ft2u
- code rearrangement is needed, yet to be done

- **pocket_ttd update**

- first priority: need to properly react against errors
- first thing to do: clean up statft dependeces
- next thing to do: make statft and other variables

Code update 2

● ttctrld update

- ERROR reporting are not properly implemented
- PAUSE handling is needed to pause trigger during CDC HV trip
- ttctrld should not do VME access
- extra/ftpargs2 to be removed from daq_slc
- possible only after pocket_ttd update

Data format

Information sent from TTD to datastream

- (header)

word 0: bit[31:0] nword=22

word 1: bit[31:0] nword_in_header=8

word 2: bit[31:16] nevent=1; bit[15:0] nboard=1

word 3: bit[31:22] exp.number; bit[21:8] run.number; bit[7:0] sub-run.number

word 4: bit[31:0] event.number

word 5: bit[31:0] unassigned=0

word 6: bit[31:8] nodeid=0x545444 (ascii code for "TTD"); bit[7:0] version=0x20 (as of 2019.3.25, next version will be 0x31, ascii code '1') **(updated 2019.3.25)**

word 7: bit[31:0] unassigned=0

- (data)

word 8: bit[31] 0; bit[30:4] ctime; bit[3:0] trgtyp;

word 9: bit[31:0] utime;

word 10: bit[31:0] event.number;

word 11: bit[31:0] frame-count;

word 12: bit[31:0] time since previous trigger

word 13: bit[31] her-inj(1) or ler-inj(0); bit[30:0] time since last inj

word 14: bit[31:11] unused=0; bit[10:0] bunch number (yet to be implemented)

word 15: bit[31:0] unused=0

word 16: bit[31:0] unused=0

word 17: bit[31:0] unused=0

word 18: bit[31:0] unused=0

word 19: bit[31:0] unused=0

- (trailer)

word 20: bit[31:0] reserved=0x54544420

word 21: bit[31:0] magic-number=0x7fff0000

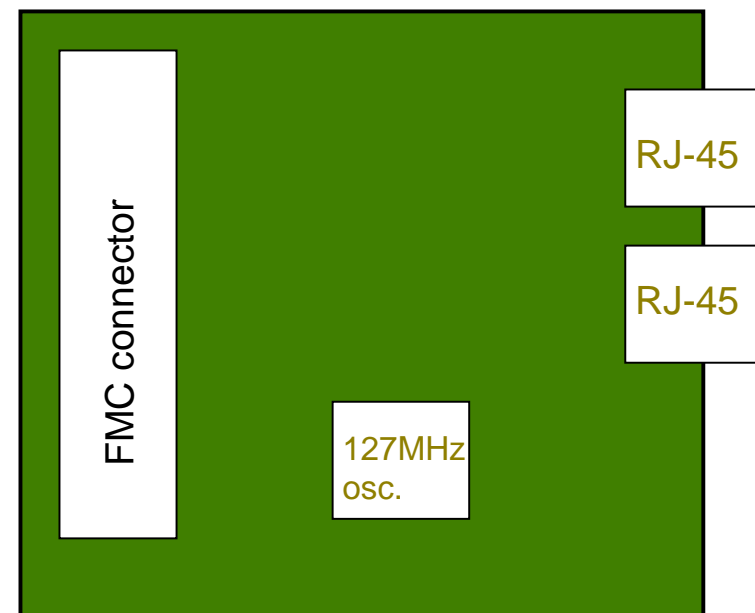
Time is in system clock (127 MHz) unit, for 32-bit, unless otherwise specified.

[from DAQ/TTD confluence]

- Version number is still 0x20 now
- RawFTSW code was recently modified to accept other versions
- NTP time (sec, μ sec) to be added, if it does not degrade the readout speed
- If it works, new version number will be used

Pocket-FTSW?

- **PocketDAQ / Pocket-DAQ-upgrade**
does not need full FTSW spec
- **1-port FTSW function** can be implemented on Xilinx eval board (e.g., sp605)
- **Missing on eval board:**
127 MHz clock, RJ-45 connector
- **FMC card**
 - 1 port FTSW can be easily generated
 - Second RJ-45 to test second RJ-45 or AUX input
- **No software**
 - No need to prepare VME CPU
 - Chipscope based control for limited ftprogs function



Anybody willing to work on this?

Work priority

- 1. NSM2 update
- (to discuss tomorrow)
- 2. ftprogs (statft) clean-up
- 3. pocket_ttd clean-up
- 4. ttctrld clean-up
- 5. daq_slc clean-up
- Implementation of PAUSE, error report, etc are in the course of the clean-up procedure above