Explanation on the

startup_TRB49.sh script

Sequence of commands and details on the required files

The script called to startup TRB3 and initialize ADC AddOns is **startup_TRB49.sh**[1]

[Local reference: /home/odroid/trbsoft/userscripts/trb49/]

The file can be downloaded from the repository (inside the directory /userscripts_trb49_on_control_unit_OdroidC2):
https://github.com/jsmarcos/TRB3-GSI-based-DAQ

Direct link: https://raw.githubusercontent.com/jsmarcos/TRB3-GSI-based-DAQ/master/userscripts_trb49_on_control_unit_OdroidC2/startup_TRB49.sh

It sends the configuration of several parameters in the FPGA and ADC AddOns of TRB3 board, throught the application 'trbcmd', which sends commands to TRB3.

→ **Four operations** executed by the **startup_TRB49.sh** script:

1) Check if there is **communication with TRB3 boar**d, reset the system and ask for the available "endpoints" (FPGAs). This is done through two Perl scripts: *check_ping.pl* and *trbreset_loop.pl* refered to below.

2) Set the name of the **available FPGAs** (confirm)

3) Set the **GbE settings** throught the Perl script **loadregisterdb.pl**, taking the required information from Perl database (text files). Example of a setting: IP addresses and ports of the machine that will recieve the .HLD files with the channels waveforms.

Below are the Perl command used in the **startup_TRB49.sh** script[2]:

```
loadregisterdb.pl db/register_configgbe.db    # must be changed accordingly (ID of the CTS_FPGA as defined above)
loadregisterdb.pl db/register_configgbe_ip.db          # must be changed accordingly (DAQ_MAC + IP + Dest Port (used in EventBuilder_TRB186.xml))

#echo "TDC settings"
loadregisterdb.pl db/register_configtdc.db
```

Note that in the beginning of the **startup_TRB49.sh** script the path for the Perl application loadregisterdb.pl is defined, through the commands:

```
DAQ_TOOLS_PATH=~/trbsoft/daqtools
export PATH=$PATH:$DAQ_TOOLS_PATH
export PATH=$PATH:$DAQ_TOOLS_PATH/tools
```

In the current software installation, the Perl databases are saved in a directory "db", inside the **startup_TRB49.sh:**

---

1 A copy of the startup_TRB49.sh script is available in this document.

2 The databases (.db files) passed as argument of the Perl scripts are copied in the end of this document.

/home/odroid/trbsoft/userscripts/trb49/db

4) **Setting up the ADCs AddOns**

The FPGAs that control the ADCs AddOns are initialized and configured with the parameters written in the script (passed as argument in ‘trbcmd’).

Examples of parameters are: number of samples, trigger thresholds, downsampling factor, number of the trigger channels , etc.

→ **NOTES:**

Note that when called remotely, the **startup_TRB49.sh** script does not launch the CTS web controller in the end. Is does that operation when it is called in the controller machine directly.

There is a specific script to be executed from remote: **startup_TRB49_remotely.sh**[3]**.**
The differences are only two in the remote case: 1) the path for the executable ‘trbcmd’ is explicitly writen and 2) the CTS web controller is not lauched in the end of the script. In remote case, the CTS controller is launched from other script called **launch_cts.sh**.

Code of the **script launch_cts.sh**

```
#!/bin/bash

export DAQOPSERVER=localhost:1 #Check if it is needed

echo "Launching cts_gui"

pkill -f "perl /home/odroid/trbsoft/daqtools/web/cts_gui"
cd /home/odroid/trbsoft/daqtools/web/
/home/odroid/trbsoft/daqtools/web/cts_gui # removed'&' in the end of the line. Jan 2019. JM
#/home/odroid/trbsoft/daqtools/web/httpi localhost 1234 &

#Load CTS configuration
#cd /home/odroid/trbsoft/userscripts/trb49/
#sh ./cts-dump-remotely.sh
```

Code of the main script **startup_TRB49.sh**

```
#!/bin/bash

#sleep 60
cd /home/odroid/trbsoft/userscripts/trb49

#to run the script from the crontab:
if [[ -x ~/.bashrc ]]; then
   chmod a+x ~/.bashrc
fi
PS1='$ '; source ~/.bashrc

DAQ_TOOLS_PATH=~/trbsoft/daqtools
export PATH=$PATH:$DAQ_TOOLS_PATH
export PATH=$PATH:$DAQ_TOOLS_PATH/tools

export DAQOPSERVER=localhost:1
```

3 This file is also available in the repository https://github.com/jsmarcos/TRB3-GSI-based-DAQ, inside the directory /userscripts_trb49_on_control_unit_OdroidC2

```
#export TRB3_SERVER=trb012:26000               #change to the respective host name (the same than
/etc/hosts)
export TRB3_SERVER=trb049:26000                #change to the respective host name (the same than
/etc/hosts)
export TRBNETDPID=$(pgrep -f "trbnetd -i 1")
echo "- trbnetd pid: $TRBNETDPID"

if [[ -z "$TRBNETDPID" ]]
then
    ~/trbsoft/trbnettools/bin/trbnetd -i 1      #starts trbnetd which allows connection with the TRB3; trbnetd
-i 1 opens a trbnetd with th$
fi

./check_ping.pl                                 #ping the TRB, the script must be changed accordingly
with the respective TRB name

echo "reset"
./trbreset_loop.pl                              #the # of endpoints must be changed accordingly (5
endpoints for 1 TRB)
sleep 1;

trbcmd i 0xffff
#trbcmd s 0x12000002e2d98d28 0x05 0xc001 #CTS TRB12
#trbcmd s 0xcf000003480d8a28 0x00 0xa001 #ADA1
#trbcmd s 0x99000002e30e5728 0x01 0xa002 #ADA2
#trbcmd s 0x4e000002e2e24c28 0x02 0xa003 #ADA3
#trbcmd s 0x6a000002e2e24328 0x03 0xa004 #not used
trbcmd s 0x6a000006e95d3528 0x00 0xC310 #ADC
trbcmd s 0xef000006e95d3228 0x01 0xC311 #ADC
trbcmd s 0xcf000006e95d1b28 0x02 0xC312 #not used
trbcmd s 0xb1000006e9545028 0x03 0xC313 #not used
trbcmd s 0x96000006e95d1828 0x05 0x8000 #CTS TRB49

trbcmd i 0xffff


echo "GbE settings"
loadregisterdb.pl db/register_configgbe.db      #must be changed accordingly (ID of the CTS_FPGA as
defined above)
loadregisterdb.pl db/register_configgbe_ip.db   #must be changed accordingly (DAQ_MAC + IP + Dest Port
(used in EventBuilder_TRB186.xml))

#echo "TDC settings"
loadregisterdb.pl db/register_configtdc.db      #must be changed accordingly (ID of the TDC_FPGAs as
defined above)
#echo "TDC settings end"

#echo "pulser"
# pulser #0 to 10 kHz
# trbcmd w 0xc001 0xa156 0x4e20       # 0x4268 = 17000 -> 170000 ns = 0.170ms -> 5.88 kHz (1ADC
+ 1 TDC -> 87MB/s, 8% of deadtime)

#echo "pulser enable"
# pulser enable
# trbcmd setbit 0xc001 0xa101 0x2

# JM commented the three commands below, from Alberto/João Saraiva group
#echo "start trigger"
#trbcmd w 0xc001 0xa101 0xffff4000  #enable ch14 (coincidence module 0) trg_channel_mask:
edge=1111 1111 1111 1111, mask=0000 0000 0010 0000

#echo "start coincidences with detector 1&3"
#trbcmd w 0xc001 0xa13e 0x30005              #trg_coin_config0: coin_mask=0000 0101,
inhibit_mask=0000 0000; window=3 = 30 ns

#echo "limit cts_throttle"
#trbcmd w 0xc001 0xa00c 0x00000401  # cts_throttle: enable=true, stop=false, threshold=1


#Threshold Settings - Below some settings of Alberto/João Saraiva group
```

```
#~/trbsoft/daqtools/tools/dac_program.pl ~/trbsoft/userscripts/trb12/thresholds/configFile_a001_40mv
#~/trbsoft/daqtools/tools/dac_program.pl ~/trbsoft/userscripts/trb12/thresholds/configFile_a002_40mv
#~/trbsoft/daqtools/tools/dac_program.pl ~/trbsoft/userscripts/trb12/thresholds/configFile_a003_40mv

#setup ADC addon
Samples=30
SamplesAfterTrigger=60
Threshold_Ch9=80
Threshold_Ch100=220

FPGA="0xC310"
~/trbsoft/daqtools/tools/adc.pl $FPGA init

trbcmd w $FPGA 0xa010 $Samples                  #Buffer depth
trbcmd w $FPGA 0xa011 $SamplesAfterTrigger      #Samples after trigger
trbcmd w $FPGA 0xa012 1                         #Process blocks
trbcmd w $FPGA 0xa013 $Threshold_Ch9        #Trigger offset, invert
trbcmd w $FPGA 0xa014 0          #Readout offset
trbcmd w $FPGA 0xa015 1          #Downsampling
trbcmd w $FPGA 0xa016 8          #Baseline
#trbcmd w $FPGA 0xa017 0x30000000  #Trigger Enable ch31-00
#trbcmd w $FPGA 0xa018 0x0000      #Trigger Enable ch47-32
trbcmd w $FPGA 0xa017 0x30000      #Trigger Enable ch31-00   JM: Hardware channels 16 and 17
trbcmd w $FPGA 0xa018 0x0       #Trigger Enable ch47-32
trbcmd w $FPGA 0xa01a 0x00000000  #Channel disable ch31-00, all channels except ch0
trbcmd w $FPGA 0xa01b 0x0000      #Channel disable ch47-32
trbcmd w $FPGA 0xa01c 0          #Processing mode 0=BlockMode, 1=PSA, 2=CFD
# ignore CFD stuff
#trbcmd w $FPGA 0xa01d 0x340       #CFD delay is 3, CFD window 64=0x40

trbcmd w $FPGA 0xa020 1          #Sum values
trbcmd w $FPGA 0xa021 1          #Sum values
trbcmd w $FPGA 0xa022 1          #Sum values
trbcmd w $FPGA 0xa023 1          #Sum values
trbcmd w $FPGA 0xa024 $Samples    #word count
trbcmd w $FPGA 0xa025 0          #word count
trbcmd w $FPGA 0xa026 0          #word count
trbcmd w $FPGA 0xa027 0          #word count

trbcmd w $FPGA 0xa000 0x100       #Reset Baseline

# Second ADC
FPGA1="0xC311"
~/trbsoft/daqtools/tools/adc.pl $FPGA1 init

trbcmd w $FPGA1 0xa010 $Samples                  #Buffer depth
trbcmd w $FPGA1 0xa011 $SamplesAfterTrigger      #Samples after trigger
trbcmd w $FPGA1 0xa012 1                         #Process blocks
trbcmd w $FPGA1 0xa013 $Threshold_Ch100              #Trigger offset, invert
trbcmd w $FPGA1 0xa014 0          #Readout offset
trbcmd w $FPGA1 0xa015 1          #Downsampling
trbcmd w $FPGA1 0xa016 8          #Baseline
#trbcmd w $FPGA1 0xa017 0x30000000  #Trigger Enable ch31-00
#trbcmd w $FPGA1 0xa018 0x0000      #Trigger Enable ch47-32
trbcmd w $FPGA1 0xa017 0x30000      #Trigger Enable ch31-00  JM: Hardware channels 64 and 65
trbcmd w $FPGA1 0xa018 0x0         #Trigger Enable ch47-32
trbcmd w $FPGA1 0xa01a 0x00000000  #Channel disable ch31-00, all channels except ch0
trbcmd w $FPGA1 0xa01b 0x0000      #Channel disable ch47-32
trbcmd w $FPGA1 0xa01c 0          #Processing mode 0=BlockMode, 1=PSA, 2=CFD
# ignore CFD stuff
#trbcmd w $FPGA 0xa01d 0x340       #CFD delay is 3, CFD window 64=0x40

trbcmd w $FPGA1 0xa020 1          #Sum values
trbcmd w $FPGA1 0xa021 1          #Sum values
trbcmd w $FPGA1 0xa022 1          #Sum values
trbcmd w $FPGA1 0xa023 1          #Sum values
trbcmd w $FPGA1 0xa024 $Samples    #word count
trbcmd w $FPGA1 0xa025 0          #word count
trbcmd w $FPGA1 0xa026 0          #word count
trbcmd w $FPGA1 0xa027 0          #word count
```

```
trbcmd w $FPGA1 0xa000 0x100       #Reset Baseline
# Second ADC Done


#CTS_GUI
echo "Launching cts_gui"
pkill -f "perl $HOME/trbsoft/daqtools/web/cts_gui"
cd $HOME/trbsoft/daqtools/web/
$HOME/trbsoft/daqtools/web/cts_gui &

   #echo "Launching cts_gui"
   #pkill -f "perl /home/odroid/trbsoft/daqtools/web/cts_gui"
   #cd /home/odroid/trbsoft/daqtools/web/
   #/home/odroid/trbsoft/daqtools/web/cts_gui &

#xterm -display :1 -hold -geometry 105x60 +sb -e '$HOME/trbsoft/daqtools/web/cts_gui --noopenxterm'

cd - #JM: to return to the directory in which the previous command was executed
#Load CTS configuration
sh ./cts-dump.sh
```

→ Perl scripts

/home/odroid/trbsoft/userscripts/trb49/

1. Script **check_ping.pl**

Checks the avaibility of the communication with TRB3 board (among other things)

2. Script **trbreset_loop.pl**

Starts by reseting the TRB3 board.

Inquires the name of the available 'endpoints' (FPGAs), including the central/controller one (usually called 0x8000).

→ Three Perl databases

/home/odroid/trbsoft/userscripts/trb49/**db**

1. register_configgbe_ip.db

```
################################################################################################
#########
#Eventbuilders:
# EB 0:   kp1pc105  eth1  00:1B:21:43:97:EA 192.168.0.2 ports 50000 - 50099

!Register table
#  Type  #   C0   #   C1   #   C2   #   C3   #   C4   #   C5   #   C6   #   C7   #   C8   #
################################################################################################
#########
#new memory locations
    0    0x8100  0x8101  0x8102  0x8103  0x8104  0x8105  0x8106  0x8107  0x8108
    1    0x8110  0x8111  0x8112  0x8113  0x8114  0x8115  0x8116  0x8117  0x8118
    2    0x8120  0x8121  0x8122  0x8123  0x8124  0x8125  0x8126  0x8127  0x8128
    3    0x8130  0x8131  0x8132  0x8133  0x8134  0x8135  0x8136  0x8137  0x8138
    4    0x8140  0x8141  0x8142  0x8143  0x8144  0x8145  0x8146  0x8147  0x8148
    5    0x8150  0x8151  0x8152  0x8153  0x8154  0x8155  0x8156  0x8157  0x8158
    6    0x8160  0x8161  0x8162  0x8163  0x8164  0x8165  0x8166  0x8167  0x8168
    7    0x8170  0x8171  0x8172  0x8173  0x8174  0x8175  0x8176  0x8177  0x8178
    8    0x8180  0x8181  0x8182  0x8183  0x8184  0x8185  0x8186  0x8187  0x8188
    9    0x8190  0x8191  0x8192  0x8193  0x8194  0x8195  0x8196  0x8197  0x8198
   10    0x81A0  0x81A1  0x81A2  0x81A3  0x81A4  0x81A5  0x81A6  0x81A7  0x81A8
   11    0x81B0  0x81B1  0x81B2  0x81B3  0x81B4  0x81B5  0x81B6  0x81B7  0x81B8
   12    0x81C0  0x81C1  0x81C2  0x81C3  0x81C4  0x81C5  0x81C6  0x81C7  0x81C8
   13    0x81D0  0x81D1  0x81D2  0x81D3  0x81D4  0x81D5  0x81D6  0x81D7  0x81D8
   14    0x81E0  0x81E1  0x81E2  0x81E3  0x81E4  0x81E5  0x81E6  0x81E7  0x81E8
   15    0x81F0  0x81F1  0x81F2  0x81F3  0x81F4  0x81F5  0x81F6  0x81F7  0x81F8


## Alberto/João Saraiva:
#TRB12:  DA7A 32E2D98D 10.0.1.135 ->87
#ODROID: 001E 0634FAB2 10.0.1.67  ->43

## Gamma cameras group - B16
```

```
#TRB49 (B16): 02:00:f8:00:be:18 10.0.0.49 -> 49 in Hexadecimal = 31
#ODROID B16: 00:1e:06:36:49:d0 10.0.0.1 -> 1 in Hexadecimal = 1
#JM Laptop: f0:79:59:2f:4e:ee 10.0.0.61 -> 61 in Hexadecimal = 3D
#B16 Linux PC: 14:cc:20:00:6c:61 10.0.0.65 -> 65 in exadecimal = 41

!Value table
#                Dest MAC    Dest MAC      Dest IP    Dest Port    Src MAC    Src MAC      Src IP       Src Port   Packet Size
# Hub   #  Type #    C0        #    C1       #    C2        #    C3      #    C4       #    C5      #    C6    #    C7    #    C8    #
##############################################################################################################################

#0xc001      0  0x0634fab2   0x001e    0x0a000143    50084    0x0634FAB2   0xda7a  0x0a000187    0xc350    0x0578
#0x8000      0  0x063649d0  0x001e    0x0a000001    50084    0xf800be18    0x0200  0x0a000031    0xc350    0x0578
#0x8000      0  0x592f4eee   0xf079    0x0a00003D    50084    0xf800be18    0x0200  0x0a000031    0xc350    0x0578   # JM
Laptop
 0x8000       0   0x20006c61    0x14cc      0x0a000041    50084     0xf800be18     0x0200   0x0a000031      0xc350
0x0578   # B16 Linux PC (delta)
```

2. register_configgbe.db

```
!Register table
#  Type  #   C0   #   C1   #   C2   #   C3   #   C4   #   C5   #   C6   #   C7   #   C8   #  C9   # C10
##############################################################################
#############
   0    0x8300  0x8305  0x8307  0x8308  0x830b
   1    0x8301  0x8302  0x8304  0x8309  0x830c  0x830e  0x830f  0x8310

!Value table
#
                                        Enable
#              SubEvtId     UseGbE     MultiQueue   Trig. Num.  InclTrgType
# Hub    #  Type #    C0    #    C1    #    C2    #    C3    #     C4
##############################################################################
#############
   0x8000     0      0x8000        1         0       0xffffff       1
#  0xc001     0      0xc001        1         0       0xffffff       1
#  0xc002     0      0x8000        1         0       0xffffff       1


#These values to not need to be written - for completeness only
#              SubEvtDec    QueDec     FrameSize   RX enable    SubEvtSize   Evt/Queue    QueueClose  MaxQueueSize
# Hub    #  Type #    C0    #    C1    #    C2   #    C3    #    C4    #    C5    #    C6    #    C7    #
##############################################################################
######################
# 0xff7f     1     0x00020001   0x00030062    0x578       1       59800       200      32000       60000
```

3. register_configtdc.db

```
# TDC / ADC config registers

!Register table
#  Type  #   C0   #   C1   #   C2   #   C3   #   C4   #   C5   #
##############################################################
   0    0xc800  0xc801  0xc802  0xc803  0xc804  0xc805

!Value table
#               Gnl Conf    Trg Window    Ch En 1-32    Ch En 33-64 RingBufSize    Invert
# TDC    #  Type #    C0    #    C1    #    C2    #    C3    #    C4    #    C5    #
##############################################################################
####################

#0xA001             0  0x00000000  0x80640064  0x3fffffff  0x00000000  0x0000007c  0x00000000     # TRISTAN
#0xA002      0  0x00000000  0x80640064  0x3fffffff  0x00000000  0x0000007c  0x00000000     # TRISTAN
#0xA003      0  0x00000000  0x80640064  0x3fffffff  0x00000000  0x0000007c  0x00000000     # TRISTAN

0xC310      0  0x00000000         0x80640064  0x3fffffff  0x00000000  0x0000007c  0x00000000     # ADC1
0xC311      0  0x00000000         0x80640064  0x3fffffff  0x00000000  0x0000007c  0x00000000     # ADC2
```