

MCMTESTS

20201029

Power on

1. 5to1 card: should show 5V(?) and 0.10 - 0.12A
2. MCM – control voltages and currents (current is normally ca 2.0A at this stage)
3. Press reset button on 5to1 card. THIS MUST DONE AFTER A POWER ON OF MCM (else will the I2C communication no work).
4. SRU. Press reset button on SRU (by experience is this sometimes needed)

The MCM should then draw ca 2.2A.

Program startup

From a terminal window type:

ilc-start.sh

This will open a number of windows/tabs and programs. The locations refer to where they are placed on the screen.

Upper left is a window in the preselected catalogue ilc/ilcdaq/ilcserver-v10.2/

This is where the readout program is started/stopped.

Start the readout program:

./ilcserver

To stop it:

Ctrl-C.

When the system hangs must the program be stopped and restarted.

Upper right is a window with two tabs, both tabs in the preselected catalogue ilc/ilcdaq/testt-v10.0/

This is where testprograms are executed. Use one tab for testing if there is a response from the MCM. The other tab for running the testscripts.

Lower left is a window with two tabs. One with the preselected catalogue ilc/ilcdaq/gui-v10.0/. This is the place where the program to initialize the DAQ is executed:

java DaqPanels

This must be done each time the **ilcserver** above has been started and before running any testscript.

In the GUI that appears do:

1. Power on
2. PCA load
3. DAQ start
4. Exit

If the STATUS box appear red after any of 1-3, then one need to check why, reset or restart the SRU. More about this below.

The other tab is with the preselected catalogue wdir/ilcdaq/config.v10.1/ (to bec checked) . This is used to edit DAQ comfiguration files. Normally not needed, expert use.

Lower right is a window with three tabs. Tab 1 is for running **root**, which is started automatically in this tab. Tab 2 is for editing root scripts (need rudimentary knowhow of root C++ to do that), tab 3 to create the directory for testresults (normally the current date and the subdirectory root in that directory) and to copy files to this directory. E.g.

```
mkdir 20201029
```

```
mkdir 20201029/root
```

Put log outputs and descriptions in the file readme.txt.

Copy pdf and root files with these commands:

```
*** COPY COMMANDS Examples ***
ls -l *.pdf | grep "okt 29" | awk '{print $9}' | xargs cp -a -t 20201029/
ls -l *.root | grep "okt 29" | awk '{print $9}' | xargs cp -a -t 20201029/root
```

Replace "okt 29" and 20201029 with appropriate date.

Putty window: Putty is used to communicate with the 5to1 card. Putty is automatically started.

(

If starting it by hand then do:

1. Select 5to1
 2. Load it
 3. Open it
-)

Before doing it must the 5to1 card be powered up and has been reset as described above.

Putty starts a window for writing commands to the 5to1 processor. The only command currently needed is to initialize/change the DAC settings on the MCM. This is only needed after each power on of the MCM.

Initilaize the DAC with default values (125=250mV refN, 375=750mV cmin/cmout, 625=1250 mV refP), at the prompt '>' type (if there is no prompt hit CR):

DIN

The reply should be:

```
S00009E0000  
S00009E0000  
S00009E0000  
S00009E0000
```

Due to the 100 ohm resistors are the default values not used, you need to change refN and refP:

refN:

```
DWR30075
```

refP:

```
DWR10675
```

The replies should be:

```
S00009E0000
```

The argument is a four character decimal value, each step corresponds to 2 mV of the DAC output:

MCMTESTS

When everything is powered up and initialized one can try to communicate with the SRU and MCM, startup of DAQ.

INIT SRU DATA PATH

In the window/tab for test communication with the SRU/MCM do:

```
./sru-init -init
```

This will tell if the SRU data transfer line is active. The output should be:

TO BE ADDED

Retry until correct reply. If no luck, try with resetting SRU. Should be done after each power on/reset av SRU.

TEST COMMUNICATION

In the upper left window start the ilcserver, if not already done. This is needed before any commands should sent to the MCM.

To test the communication with MCM do in the window/tab for test of communication:

```
./mcmdumpregs -fec 0
```

There should come a dump of the MCM registers if all is fine e.g.

TO BE ADDED

if there comes any error -3 as a reply, then try either of:

```
./srureset.sh
```

this partially resets the SRU and resyncs the data line with the MCM. Try again the mcmdumpregs. If there is a run going on then the ilcserver/daq needs to be restarted.

If this not helps:

stop the ilcserver (with Ctrl-C) and reset the SRU. Repeat sru-init and start the ilcserver/daq again.

SALTRO GLOBAL RESET

Before doing any access to the SALTRO after a power on do they need a global reset:

```
./grst.sh 0 1
```

(0 1 means PCA is not in shutdown and ADCs are running).

After a global reset always stop and restart the ilcserver. (MAY CHANGE THIS PROCEDURE).

The current should now be 3.2A

DAQ START

To allow for run and data taking start/exit the DAQ via DaqPanels as described in “Lower left is a window” section above. If status goes red in the GUI, you have to try the options in the section “TEST COMMUNICATION” until it all works.

TEST RUNS

Go to the upper right windowtab used for runs. One simple action is to take run for pedestals of one chip with 135 samples. So this is given here as the example. Other tests needs some more editing of testscripts to be used by nonexpert.

```
./mkpedestal.sh <mcm>00<chiponmcm> <gains> <shaper>
```

where the current mcm number is 2. Chips 0-7.

gain: 0 = 12mV/fC, 1=next highest, 2=next lowest, 3=27mV/fC

shaper: 0 = 120 ns, 4=90 ns, 6= 60 ns, 7 = 120 ns

(The bit ordering is due to reorder of bits between MCM and SALTRO, will do the reorder in a program when all bits is tested to work as expected).

Example:

```
./mkpedestal.sh 2001 0 0
```

will use chip 1 with lowest gain and longest shaping.

This script sets the number of samples in all chips to 135, activates the selected chip for readout in the CPLD, and takes 500 events if it all works.

The raw data file is converted to root ntuples by the test script.

In the window with the ilcserver running should one see a line for each event taken, the script will output a line for each hundred event. At the end there is crash dump of the program that converts the data to ntuples, ignore that dump. At the end is a line with a root command to execute for pedestal plots. This is to be executed in the windowtab where root is running.

E.g for the example above:

```
.x plotallped.cxx(2001,34643,"gain0-shaper0")  
(34643 is the runnumber, which is of course different for each run).
```

The output pdf-file will be:

ch-all-chip2001-34643-ped-gain0-shaper0.pdf

Add the root output text in the "readme.txt" file if it is to be stored, see as example:

<https://www.hep.lu.se/eudet/pictures/mcmtests/20201021/readme.txt>

To see the time sweep of all channels execute (again this example) the root command:

```
.x plotallchchip0x.C(10,2001,"34643-gain0-shaper0","onerun.txt")  
10 is the event number to do it for. The file onerun.txt must contain the root file to use i.e.
```

```
readout-<runnumber>_0.root
```

Currently to be edited by hand, may change it such that the script updates it.

The output pdf-file will be:

chip0x-all-chip2001-evt10-34643-gain0-shaper0.pdf

There is also a root script to make the normalized time sweep distributions:

```
.x plotallchchipxnorm.C( 2001,"34643-gain0-shaper0","onerun.txt")
```

Output pdf-file is: chipx-all-chip2001-norm-34643-gain0-shaper0.pdf

Next test:

```
./mkpedestals-shift.sh <mcm>00<chipnb> <gain> <shaper> <samples in decimal> <channels in hex>
```

This do the shift of sampling.

TO BE ADDED more text here.

PROBLEMS

It takes sometimes long before the first event is transferred. If more than 5 minutes, you have to restart things. (see section "TEST COMMUNICATION").

It sometimes just stops in the middle of a run, again see section “TEST COMMUNICATION”. If the communication works it might just be it has reached the event number 28560. Then it normally recovers by itself after a few minutes.