**Tutorial 1**

**SR1 Initial Set-up and Introduction to DAQSlice(Datataking)**

*December 2021*

SR1 is testing framework at the surface level at CERN that has replicas of the hardware present in the LHC detector (which is underground). It consists of a copy of a vertical slice of the detector. The SR1 lab enable developers to access the hardware and carry out testing at any time, regardless of the actual detector being on. The setup can be accessed and tests can be run remotely without physically being at CERN, which offers flexibility. There are both pixel and IBL style modules available. In this tutorial the process steps required to create an SR1 account, remotely compile and deploy the SR1 Infrastructure, starting the infrastructure and a short introduction to DAQSlice(datataking with emulator) and rodmon have been described.

1. **Account setup: SR1, Testbed, GitLab access and WorkPlanner**
2. The first step is to ensure that you have a CERN account
3. Request a Testbed account as described here <https://twiki.cern.ch/twiki/bin/viewauth/Atlas/Lab4Testbed#User_Accounts>
4. Request to get access to SR1 as described here

<https://codimd.web.cern.ch/LdZamtxVTbSprV8JB1jvaA#SR1-account>

1. For gitlab access subscribe to atlas-pixel-daq
2. Check if this page is visible within a few hours of being added

<https://gitlab.cern.ch/atlas-pixel/daq/atlaspixeldaq/>

A lot of the information in this tutorial can be found here

1. Ultimately you should be subscribed to **atlas-tdaq-testbed, atlas-pixel-daq-sr1-users** and **atlas-pixel-daq** e-groups**.**

Create an account in WorkPlanner

<https://atlas-sct-web.web.cern.ch/scripts/workplanner>

More information on how to select slots will be described in coming steps

1. Log in and increase quota in AFS to 10 GB to avoid quota exceeded errors during the deployment step.

<https://resources.web.cern.ch/resources/Manage/AFS/Settings.aspx>

1. **VNC Setup for remote access to SR1**

Follow instructions (in sections - **Starting a VNC server on a lxplus node** and **Connecting to a VNC server that is running on lxplus**) to set up a VNC

<https://gitlab.cern.ch/atlas-pixel/daq/atlaspixeldaq/-/wikis/VNC>

1. **Building (compiling and checking out) the software**

In this process, we come across 3 different machines, and it is helpful to know their uses before proceeding:

* lxplus – The LXPLUS service (Linux Public Login User Service) is the interactive logon service to Linux for all CERN users. This is where you will be required to log in first in the VNC server
* pix-sr1-01- SR1 – SR1 **development** machine
* pcpix-sr1-01 – SR1 **Testbed** machine

We can’t build the software in the testbed machines because of limited capacity, so we are first logging in to a SR1 development machine(pix-sr1-01) to build it, and we will deploy this to the SR1 testbed machine (pcpix)

1. Ssh into a lxplus machine and open the VNC (follow steps from the VNC webpage)
2. Go to pix-sr1 using ssh pix-sr1-01
3. Go to your directory in the home directory cd/home/<your username>
4. Clone git repo (as described here: <https://gitlab.cern.ch/atlas-pixel/daq/atlaspixeldaq/> )

git clone --recurse-submodules <https://:@gitlab.cern.ch:8443/atlas-pixel/daq/atlaspixeldaq.git>

This means that all submodules will also be downloaded. This is checking out the “Host code” and not the firmware.

1. Source command

source /cvmfs/atlas.cern.ch/repo/sw/tdaq/tools/cmake\_tdaq/bin/cm\_setup.sh tdaq-09-04-00

1. Go into the created directory cd atlaspixeltdaq
2. cmake commands

cmake\_config $CMTCONFIG

cd $CMTCONFIG

make -j12 && make install -j12

1. This might take a while- the software would be checked out and compiled if done successfully.
2. **Deployment and starting the infrastructure**
3. Copy the installed directory in atlaspixeldaq to a directory named daq in the public folder (this is the public folder each user gets in AFS)

cp -r ../installed/ ~/public/daq/

1. Still in pix-sr1-01, create directory in /tbed/user/<your username>

cd /tbed/user/<your username>

mkdir repos

mkdir repos/oks

1. Go back to the atlaspixeldaq directory, then go to DeployTools

cd ../DeployTools/

1. Run the python file

python3 deploy\_oks.py

1. Make sure the input paths are correct. If they are not automatically correct, enter it manually

VERSION- SR1

TARGET- pcpix-sr1-01:/tbed/user/<username>/repos/oks/

DAQDIR - ~/public/daq

1. Check the oks directory and ensure it looks like this



1. After this is successful exit pix-sr1-01 and go to the testbed machine (pcpix-sr1-01)

ssh -XY pcpix-sr1-01

(ensures trusted x-forwarding)

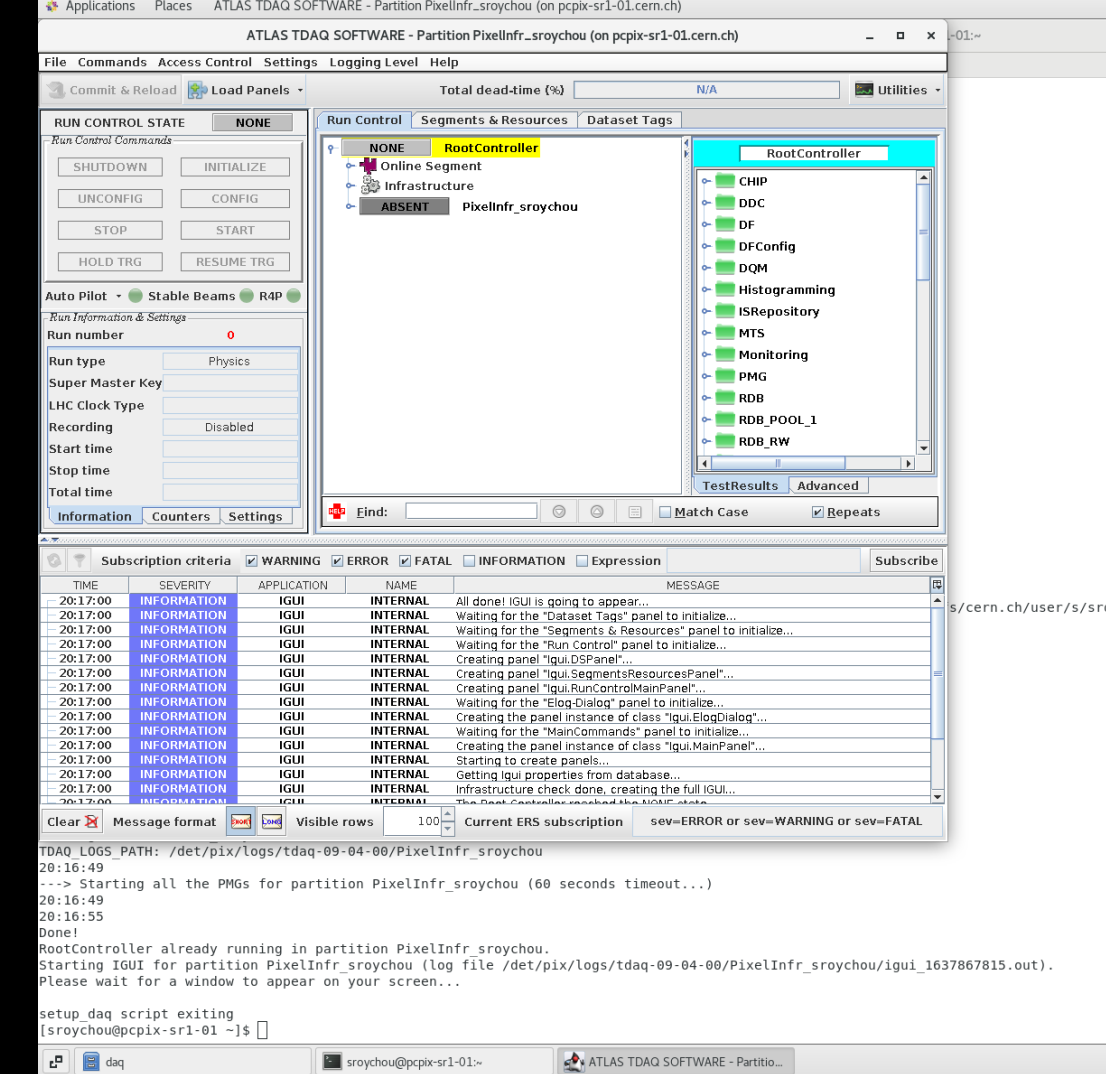
1. Source the file generated in previous step

source /tbed/user/<username>/repos/oks/setup\_everything.sh

1. Start the SR1 infrastructure partition

start\_infr

1. The following GUI window should pop up-

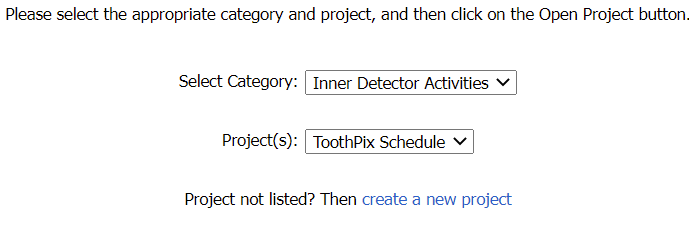


1. **Navigating the infrastructure, WorkPlanner and TagManager**
2. Go to access control and select control.
3. The initialize button should get enabled- press initialize

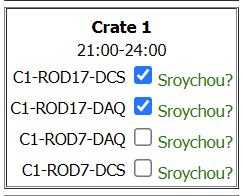
Might see backup error if this is the first time running the software

1. Select a slot in WorkPlanner that corresponds with the module you want to use. <https://atlas-sct-web.web.cern.ch/scripts/workplanner>

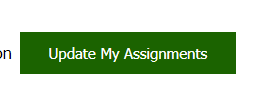
Select category and project as shown here:



Go to my assignments, scroll down to the date and time needed and select the module you will use (here C1-ROD17 has been used) and time slot. Select both boxes(DCS and DAQ) corresponding to the required module, and also the corresponding LTP box.



Click on the Update My assignments button on the top of the page



Note that the times are according to the time zone at CERN (CET). This selection doesn’t actually reflect anything in the infrastructure but is just a bookkeeping mechanism to ensure no clashes between users wanting to use the system and is good practice.

1. In the terminal type the command to open **TagManager**

start\_TagManager

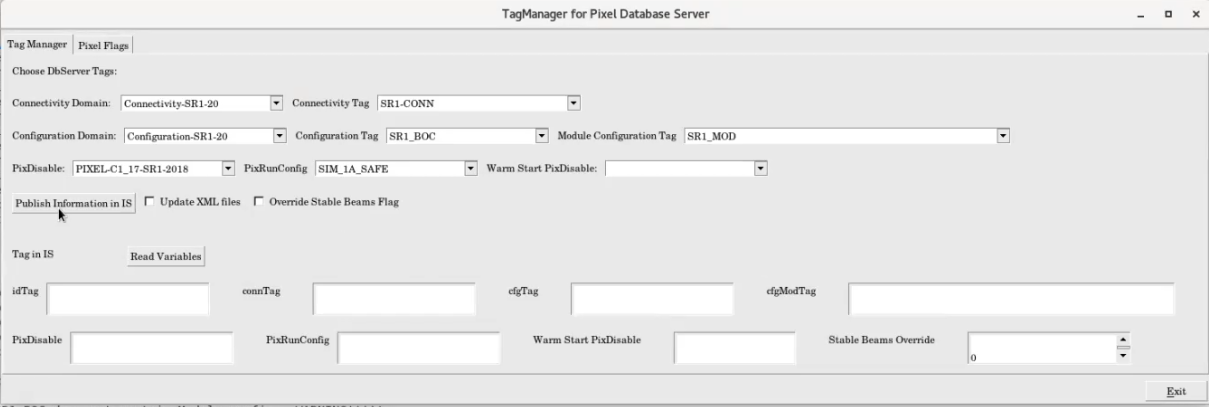
1. Here we have to select the tags that will be set as global tags in the infrastructure to reflect what we want to do. The first 6 fields have only 1 option.

The important ones are –

**PixDisable**- disables all other modules and keeps the one you want to use enabled. This should match with your selection on WorkPlanner.

**PixRunConfig** – selects what kind of run you want and what kind of data you want collected. Select SIM\_1A\_SAFE which is using an emulator, with no hits.

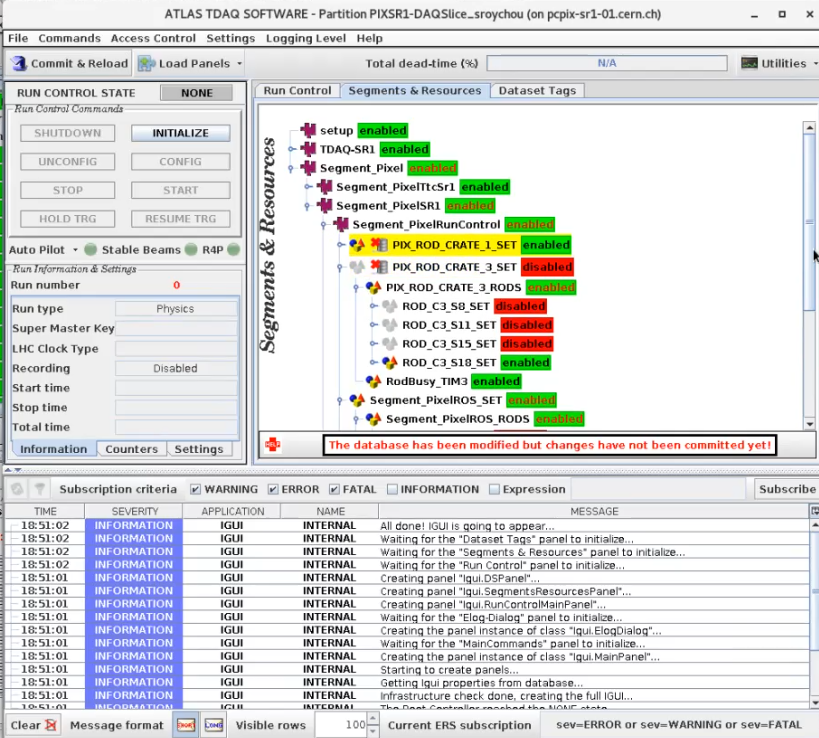
Information about other modes may be covered in future tutorials.



1. After making all the selections click Publish information in IS
2. Back in SR1 infr click config -> start
3. We can then proceed to open daqslice
4. In the end stop and exit the partition: stop->unconfig->shutdown
5. Go to Files-> Close IGUI and exit partition
6. **Intro to DAQSlice**

Here a basic set up and run using the emulator (the BOC emulates data coming in from modules)

1. Keep the SR1 infr running, in terminal run start\_daqslice
2. A new window will open up – looks very similar to Infr so be careful about which one is being used.
3. In the Segments and resources tab



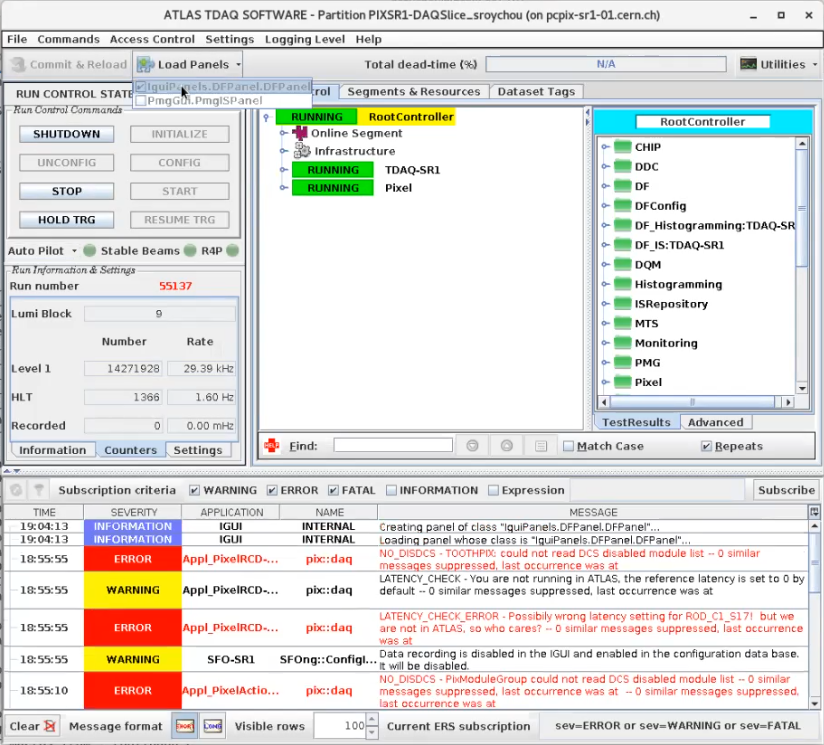
Right click PIX\_ROD\_CRATE\_1\_SET and enable it

1. Click commit and reload on top left
2. Click yes in the pop-up confirmation box
3. Select Reload all
4. Go back and check enables – only the ones we are performing data taking on should be enabled
5. If the correct ones are enabled, go to run control, INITIALIZE -> CONFIG -> START (similar to infr)

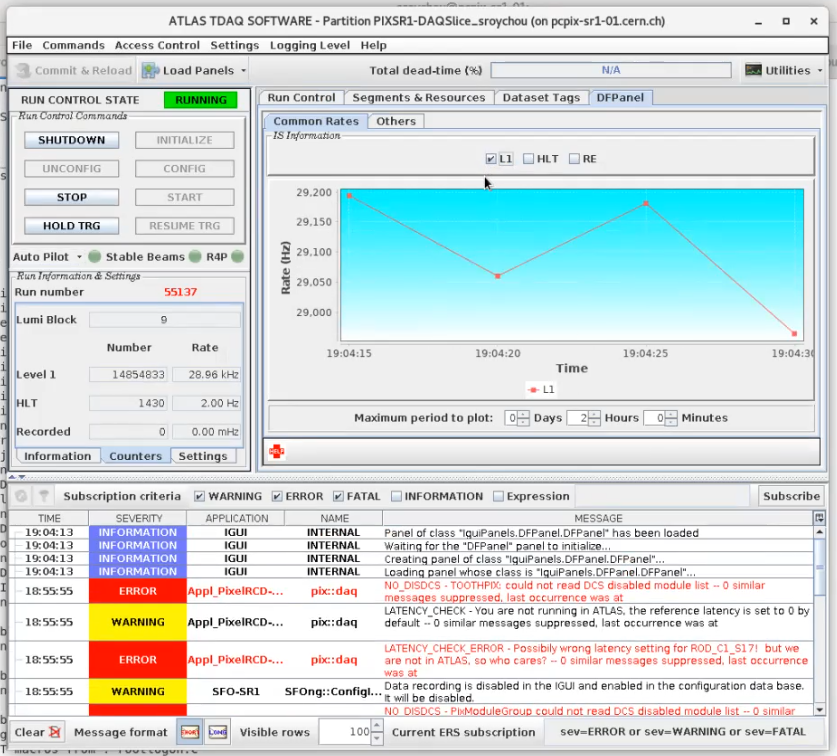
Here triggers are sent to the ROD-BOC and the data is being read back at 30 kHz

We can use this mechanism to test firmware flashed on one of the RODs through this software. Firmware is flashed using other tools which will be covered in other tutorials.

1. Select Load Panels -> iguiPanels.DFPanel.DFPanel



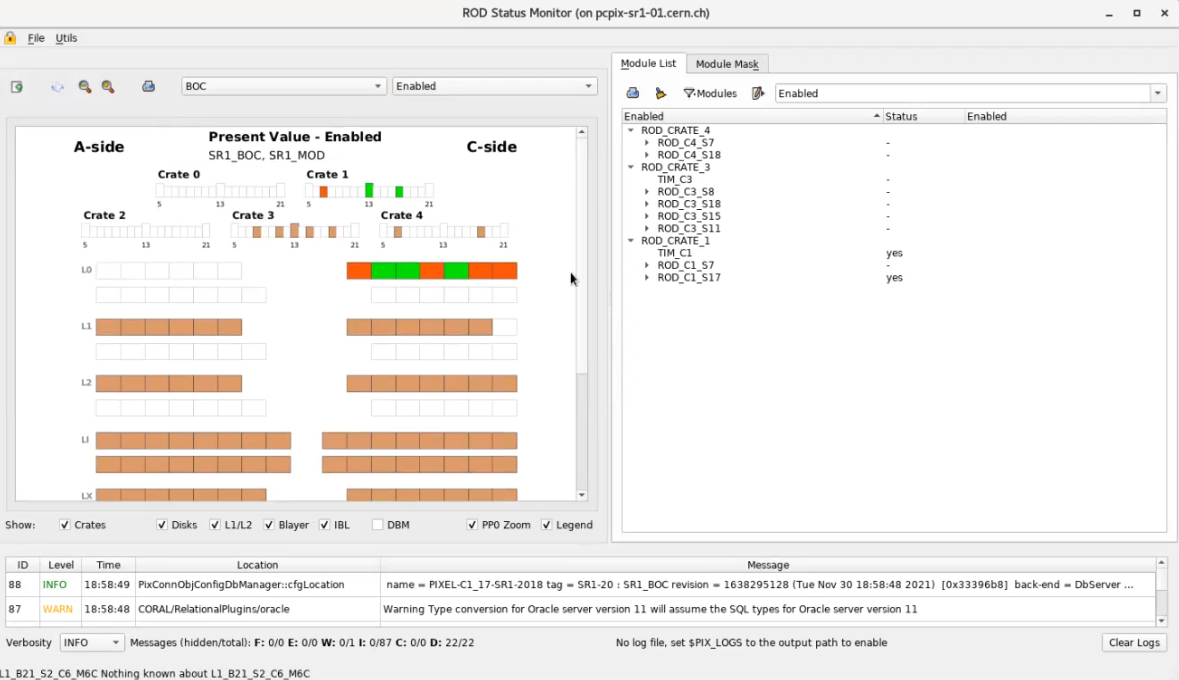
1. Select L1
2. The graph of trigger rate should be visible



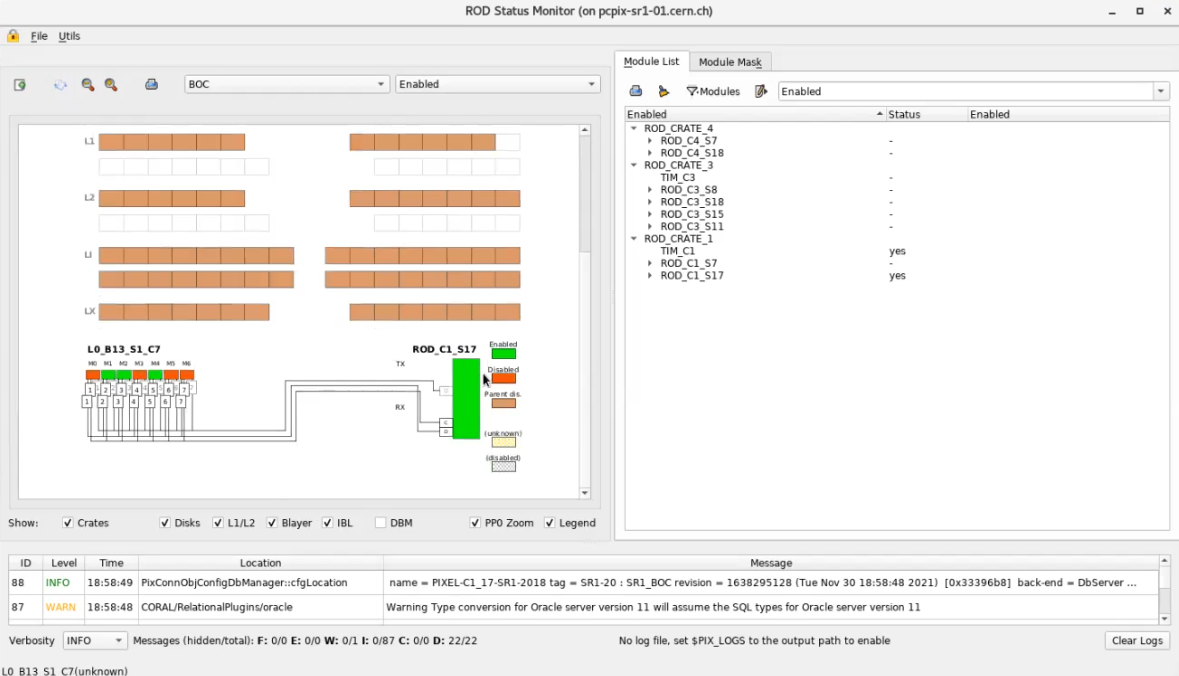
1. Click Hold TRIG – stops trigger hardware from sending triggers, registers/data can be analyzed at this point if required when testing. Observe the graph changing to show no triggers now.
2. Click STOP ->UNCONFIG-> SHUTDOWN in order
3. **Always stop DAQSlice** once you are done and ensure that it is shut down and exit the partition properly each time. While using it you are blocking other users from using it at the same time.
4. Go to Files-> Close IGUI and exit partition
5. **Viewing some data using Rodmon**

Rodmon or ROD Status Monitor can be used to view the status information about modules. This is a brief introduction on how to open and view information here. Things may be described in more detail in future tutorials.

1. While DAQSlice is running, in the terminal run start\_rodmon, the window should pop up



1. Select BOC and Enabled in the two dropdown boxes in the top left to see a schematic view of enabled modules



1. Change the drop-down box selections to see different statuses

**IMPORTANT NOTES**

1. If at all during the compiling and deployment process there is an interruption, or if the terminal was closed and reopened, run the source command before starting again

source /cvmfs/atlas.cern.ch/repo/sw/tdaq/tools/cmake\_tdaq/bin/cm\_setup.sh tdaq-09-04-00

1. To open the PixelInfr again after closing

ssh -XY pcpix

source /tbed/user/<username>/repos/oks/setup\_everything.sh

1. Always close DAQSlice properly. It blocks other people from using it if left running. This is not the case with the infrastructure but it is still better to exit properly. Also, since the two GUIs look very similar it is important to be careful about this.
2. If in contact with CERN, request to be added to the SR1 Hotline if not already. People there can help resolve issues and there could be announcements abo.ut shutdowns or errors from their end.
3. Make sure to update WorkPlanner before using the system.