



# Hands-on session getting started with PCIe40 Test Bench

Harsh Purwar

December 16th, 2021  
**Revised: September 16th, 2024**

# Outline

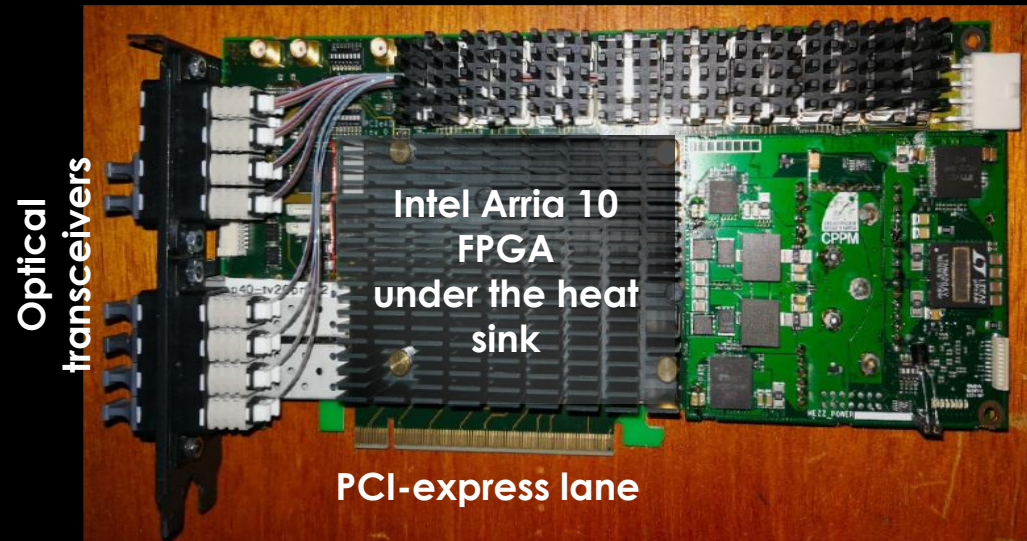
## What all we will cover today

- PCIe40 Test Bench at ID-Lab
- Setting up your account on [daqupsvr.phys.hawaii.edu](http://daqupsvr.phys.hawaii.edu)
  - Remote access: SSH, VNC, etc.
- PCIe40 software/GUI for masking/unmasking channels
- Available tools/scripts for TOP/KLM test bench configurations
- Switching between readout boards (PCIe40/COPPER)
- Introduction to the local FTSW network
- Other operational tools: **trigft**, **statft**, **ttaddr**, **ttaddr\_cpr**
- Setting up & using **basf2** to read & unpack TOP/KLM PCIe40 data

# PCle40 Test Bench

Where is it? I don't see it in the lab!

- There is no PCIe40 Test “**Bench**”, no bench at all...
- All we have in lab is a PCIe40 card (or board) and a high-end server that hosts this board.
- It is mounted inside the PCIe40 host server, located under the Big Optical Table in the lab.



# PCle40 Test Bench

## Where is it? I don't see it in the lab!

- There is no PCle40 Test Bench, no bench at all...
- All we have in lab is a PCle40 card (or board) and a high-end server that hosts this board.
- It is mounted inside the PCle40 host server, located under the Big Optical Table in the lab.
- This is a readout board (just like COPPER boards), which will be used to read data & configure (almost) all Belle II sub-detectors. CERN also plans on using this same RO board for LHCb & ALICE experiments.
- At Belle II, we have already upgraded all the sub-detectors to use this new PCle40 board.
- PCle40 was manufactured at CPPM, Marseille, France. The FW and SW libraries are provided to us by the Belle II collaboration (dev. by the DAQ Upgrade group).

# PCle40 Test Bench

Confluence: <https://confluence.desy.de/pages/viewpage.action?pageId=188791723>

- On the outside, we do see optical fiber cables (**cyan** or **orange**) that connect the PCIe40 board to an actual FEE (TOP/KLM electronics).
- COPPER boards can be connected to at max 4 FEEs, however PCIe40 can be connected to 48 FEEs.
- In the lab we have currently few FEEs connected to PCIe40 that can be used for testing.

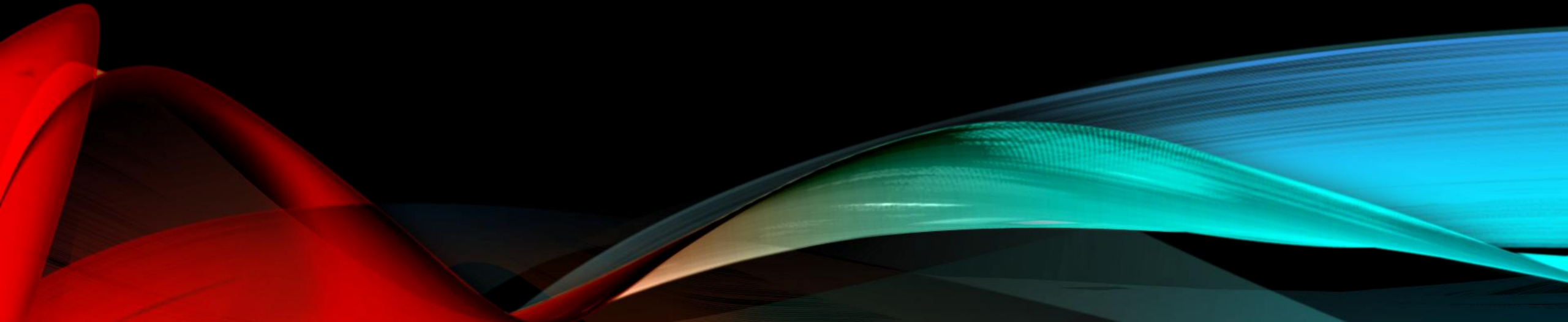
|                |                       |           |                     |    |    |          |
|----------------|-----------------------|-----------|---------------------|----|----|----------|
| PCIe40 Channel | 0                     | 1         | 2                   | 3  | 4  | 5        |
| Connected to   |                       | KLM HSLBa |                     |    |    | KLM DC   |
|                |                       |           |                     |    |    |          |
| PCIe40 Channel | 6                     | 7         | 8                   | 9  | 10 | 11       |
| Connected to   | TOP BS#8              | TOP BS#3  | <del>TOP BS#7</del> |    |    | TOP BS#5 |
|                | Issue programming BS. |           |                     |    |    |          |
| PCIe40 Channel | 12                    | 13        | 14                  | 15 | 16 | 17       |
| Connected to   |                       |           |                     |    |    |          |
|                |                       |           |                     |    |    |          |
| PCIe40 Channel | 18                    | 19        | 20                  | 21 | 22 | 23       |
| Connected to   |                       |           |                     |    |    |          |



December 16th, 2021

# Let's open that Terminal!

Setting up & securing your account on **daqupsvr**



# Setting up your account

[daqupsvr.phys.hawaii.edu](http://daqupsvr.phys.hawaii.edu)  
(192.168.153.17)

- For a new account please contact any of the existing admins:  
purwar, cketter, bessner, varner, kohani, tripathi, kurtisn
- SSH and VNC access is already setup for all users.

## Typical SSH Configuration:

```
Host idlab2
  User harsh
  Hostname idlab2.phys.hawaii.edu
  Port 24601

Host daqupsvr
  User purwar
  HostName daqupsvr.phys.hawaii.edu
  Port 22
  ProxyJump idlab2
  ForwardX11Trusted yes
  ForwardX11 yes
  LocalForward 5901 localhost:5901
```

| User     | Allocated VNC port |
|----------|--------------------|
| purwar   | 5901               |
| kohani   | 5902               |
| cketter  | 5903               |
| bessner  | 5904               |
| tripathi | 5905               |
| varner   | 5906               |
| shebalin | 5907               |
| mza      | 5908               |

Connect your VNC client at  
localhost:5901

December 16th, 2021

# SSH & VNC Passwords

- Please, please! do change your user account passwords after you login.
- Use `passwd` to change user account (login) password.
- Use `vncpasswd` to change VNC password.

## Keep your account secure!

- It is extremely important to keep your user account safe, since it is an admin account. **Your negligence may hinder other people's work.**
- Also, for what we are going to do next!



# Password-free access to TOP and KLM Test Benches

- Generate a **ssh key** pair:

```
ssh-keygen
```

- **No need to set a passphrase** (not ideal but helps with the new tools). Default filename is fine!
- You may have more than 1 key-pair (w and wo passphrase). Use unsecure key-pair only for local access (within the lab).

```
Host top
  User TOP PocketDAQ Username
```

```
Host klmvme
  User belle2
  ProxyJump klm
```

```
Host klm
  User belle2
```

```
ssh-copy-id klm
ssh-copy-id top
ssh top
ssh klm
ssh klmvme
```

December 16th, 2021

# PCle40 Firmware & Software

Thanks to Belle II Collaboration & DAQ Upgrade Team



# PCle40 Software Repository

- The source code for PCle40 software libraries is here:

PCle40 SW: [https://gitlab.desy.de/belle2/daq/pcie40\\_software](https://gitlab.desy.de/belle2/daq/pcie40_software)

- To setup PCle40 software on **daqupsvr**:

Copy your public SSH key (`~/.ssh/id_rsa.pub` from the key pair you just created on <http://gitlab.desy.de>)

```
git clone git@gitlab.desy.de:belle2/daq/pcie40_software.git
cd software/; mkdir build/; cd build/
cmake ..
Make
```

**OR**

- `ln -s /shared/software ~/`  
`source ~/software/Scripts/setup.sh` in your `~/.bashrc`

```
cd ~/software/Scripts; python3 pll_status_small.py 0
```

# Masking/Unmasking links

Command window Running on : p40\_fv21pr002  
Programming PLLs (source = Oscillator)  
..... ready .....

PLL SI54345\_1  
Input 1 : Loss of signal LOS seen  
PLL status: Loss of lock LOL seen

PLL SI54345\_2  
Input 1 : Loss of signal LOS seen  
PLL status: Loss of lock LOL seen

LOL/LOS seen is OK.

PLL SI54344  
Input 0 : Input clock present No LOS since 42.9 min.  
PLL status: Locked No LOL since 42.9 min.

## Voltages and currents

```
VCCIN      (0.9V) 0.91V 6.75A
VCCR       (1.02V) 1.06V 4.55AA
VCCT       (1.02V) 1.04V 1.05A
V1.8V      (1.8V) 1.82V 5.48A6A
A10_VCC_PT (1.8V) 1.81V 0.94A.76A
A10_V1.8    (1.8V) 1.81V 3.01A

V2.5V      (2.5V) 2.51V 3.32A.97A
V3.3V      (3.3V) 3.32V 1.89A48A

12V         (12V) 12.03V 2.2AA
12V_ATX     (12V) 0.01V -0.01A
```

Temperatures  
31.6 30.1 39.0 34.6 34.1 39.1 C

|    | RX | TX | B2L | B2U |    | RX | TX | B2L | B2U |
|----|----|----|-----|-----|----|----|----|-----|-----|
| 0  | NO | NO | NO  | NO  | 24 | NO | NO | NO  | NO  |
| 1  | NO | NO | NO  | NO  | 25 | NO | NO | NO  | NO  |
| 2  | NO | NO | NO  | NO  | 26 | NO | NO | NO  | NO  |
| 3  | NO | NO | NO  | NO  | 27 | NO | NO | NO  | NO  |
| 4  | NO | NO | NO  | NO  | 28 | NO | NO | NO  | NO  |
| 5  | OK | OK | OK  | OK  | 29 | NO | NO | NO  | NO  |
| 6  | NO | NO | NO  | NO  | 30 | NO | NO | NO  | NO  |
| 7  | NO | NO | NO  | NO  | 31 | NO | NO | NO  | NO  |
| 8  | NO | NO | NO  | NO  | 32 | NO | NO | NO  | NO  |
| 9  | NO | NO | NO  | NO  | 33 | NO | NO | NO  | NO  |
| 10 | NO | NO | NO  | NO  | 34 | NO | NO | NO  | NO  |
| 11 | NO | NO | NO  | NO  | 35 | NO | NO | NO  | NO  |
| 12 | NO | NO | NO  | NO  | 36 | NO | NO | NO  | NO  |
| 13 | NO | NO | NO  | NO  | 37 | NO | NO | NO  | NO  |
| 14 | NO | NO | NO  | NO  | 38 | NO | NO | NO  | NO  |
| 15 | NO | NO | NO  | NO  | 39 | NO | NO | NO  | NO  |
| 16 | NO | NO | NO  | NO  | 40 | NO | NO | NO  | NO  |
| 17 | NO | NO | NO  | NO  | 41 | NO | NO | NO  | NO  |
| 18 | NO | NO | NO  | NO  | 42 | NO | NO | NO  | NO  |
| 19 | NO | NO | NO  | NO  | 43 | NO | NO | NO  | NO  |
| 20 | NO | NO | NO  | NO  | 44 | NO | NO | NO  | NO  |
| 21 | NO | NO | NO  | NO  | 45 | NO | NO | NO  | NO  |
| 22 | NO | NO | NO  | NO  | 46 | NO | NO | NO  | NO  |
| 23 | NO | NO | NO  | NO  | 47 | NO | NO | NO  | NO  |

Clock FTSW  
Face plate clock : 127214530 Hz  
Run number : 5 Trigger tag : 0  
Clock Up : OK TTD Up : OK Trigger type : 15  
Trigger counter : 3701

Clock status from FTSW

F2: Program PLLs (source = FTSW), F3: Program PLLs (source = Osci.)  
F4: Clear LOS/OOF/LOL Flag, F5: Hard reset, F6: Soft reset  
F7: Reset trigger counter, r: Resynchronize one b2link  
a: Activate all channels, d: Deactivate all channels  
m: (De)activate one channel, g: (De)activate group of channels

Operational controls

Status of links  
from FEE

September 16th, 2024

# PCle40 Firmware Repository

- The source code for both PCIe40 software and firmware is on stash.

PCle40 FW: [https://gitlab.desy.de/belle2/daq/pcie40\\_firmware](https://gitlab.desy.de/belle2/daq/pcie40_firmware)

- PCIe40 FW can be automatically installed on every reboot (right now broken for some reason). Please follow the procedure to load PCIe40 firmware and driver.

## Reloading PCIe40 Firmware:

- Add `/shared/intelFPGA_pro/21.3/quartus/bin` to the PATH (in `~/.bash_profile`)
- `cd software/Scripts/`  
`./pcie40_program /usr/local/firmware/pcie40/latest.sof`
- `pcie40_reload` – to load the PCIe40 driver, required after FW programming/reboot
- In `python2 pll_status_small.py 0` GUI, press `F3`, wait, mask/unmask links.

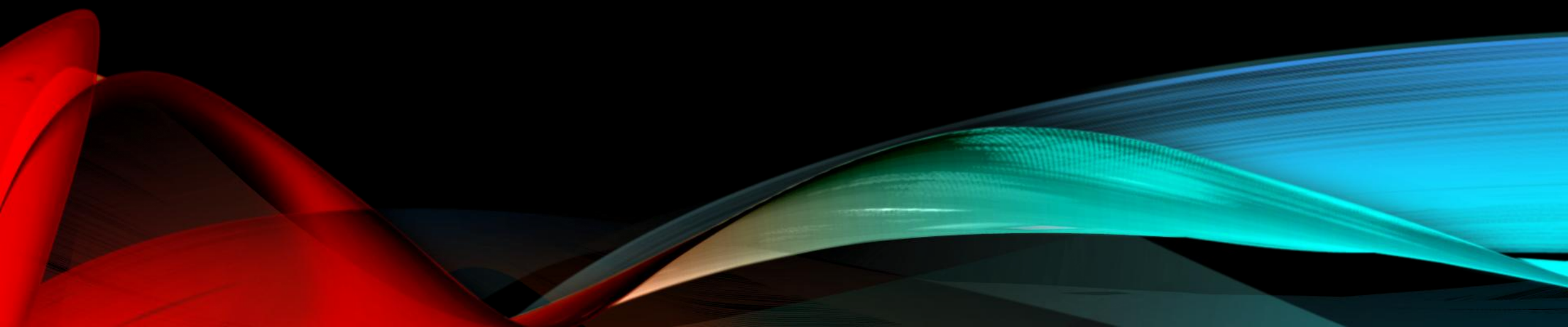
If for some reason this doesn't work the first time, please retry 1 more time after rebooting the daqpsvr, otherwise contact me (message on RC, if you have it, otherwise email!)



December 16th, 2021

# Detector configuration with PCIe40 Switching between RO boards

TOP/KLM



# Basic slow-control commands

(for PCIe40)

- reghs → `pcie40_regconfig` – Read/write FEE or PCIe40 registers  
`pcie40_regconfig --ch {0..47} --fee32 {-r addr, -w addr val}`
- staths → `pcie40_statlink` – Get status of a link  
`pcie40_statlink --ch {0..47} --fee`

```
[purwar@daqupsvr ~]$ pcie40_statlink --ch 5 --fee
statlink version 3 (20210107) / PCIe40 firmware version 14.9
memory: OK | ttd: UP | ttd clk: UP | run=: 0 | trg: 0 | trg type: 15
PLLs:LOCKED | B2L:READY (rx:111 tx:11) | DMA:FREE ( 0.0kB)
KLM serial 16 version 4
(05) b2l=UP (gbt=UP rx=UP tx=UP rxsta=READY txsta=READY mask=UNMASK)
(05) event=0 total= 0.0kB
(05) full=0 feecrcerr=202 check=NG rxrcrcerr=24446, check=0
(05) no b2link error
```

# Detector configuration before RO TOP/KLM

- Modified KLM scripts for TB configuration (now managed by Chris):  
[https://stash.desy.de/users/shebalin/repos/klm\\_scripts/browse](https://stash.desy.de/users/shebalin/repos/klm_scripts/browse)
- For TOP the scripts are located here:  
<https://gitlab.desy.de/belle2/detector/top/topConfig>

There is also a PyQt-5/6 GUI that can be used for Power-cycling & configuration of TOP BS in the lab:

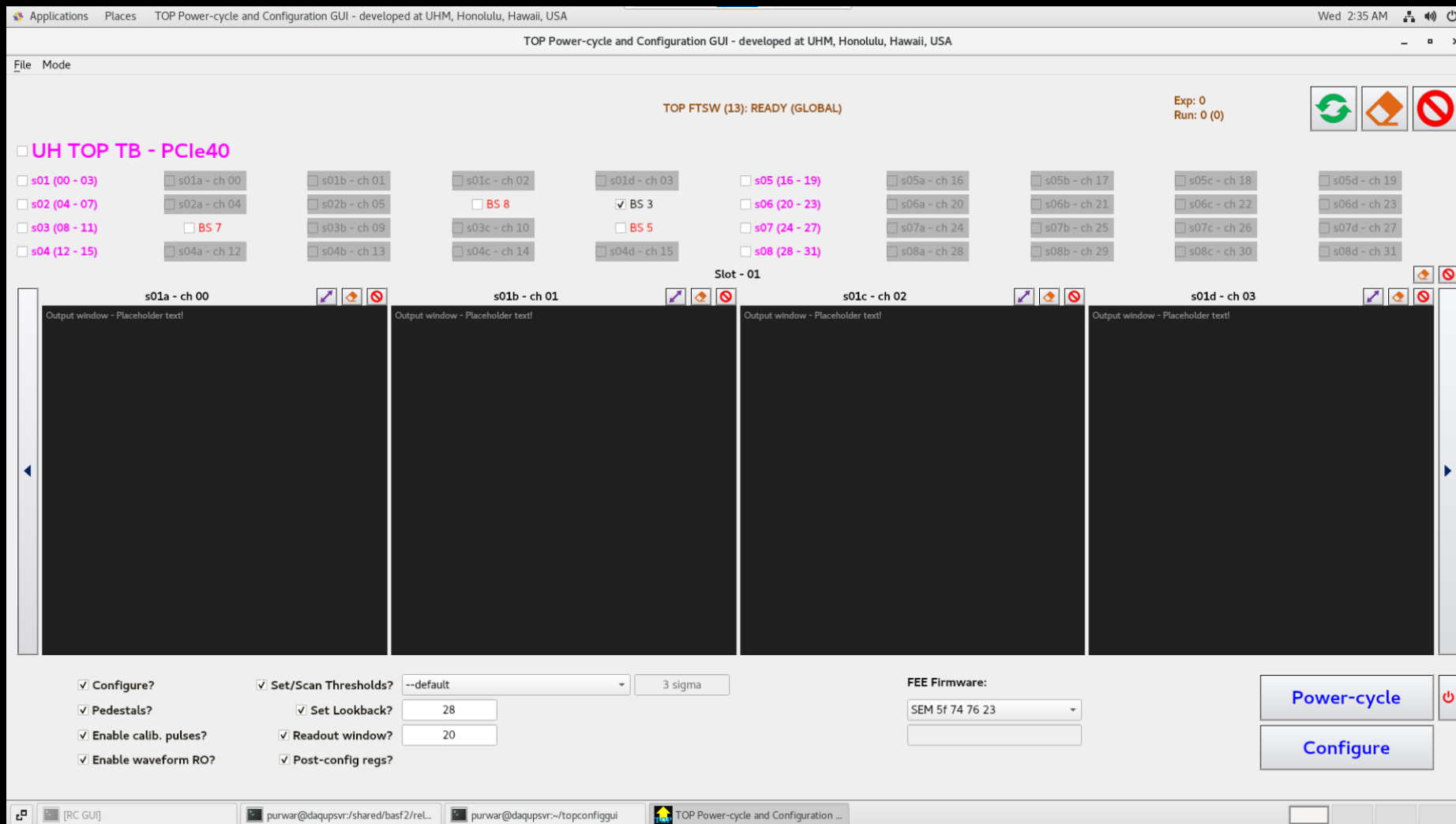
<https://gitlab.desy.de/belle2/detector/top/topconfiggui> - uhm branch

# TOP Power-cycle & Configuration GUI

```
git clone git@gitlab.desy.de:belle2/detector/top/topconfiggui.git; git checkout uhm
```

```
python3 uhGUI.py
```

If you plan on using this, please run it over VNC!



December 16th, 2021

**Note added recently:**

TOP BS at UH are now directly connected to PCIe40.  
All TOP COPPER & HSLB boards have been dismantled.

# Switching between RO boards

## PCIe40 / COPPER

- Switch FEE from PCIe40 to COPPER:
  - Turn off optical transceivers on PCIe40
  - Turn on optical transceivers on HSLB
  - Modify masking on FTSW

**switch top BS3 copper**

**switch klm copper**

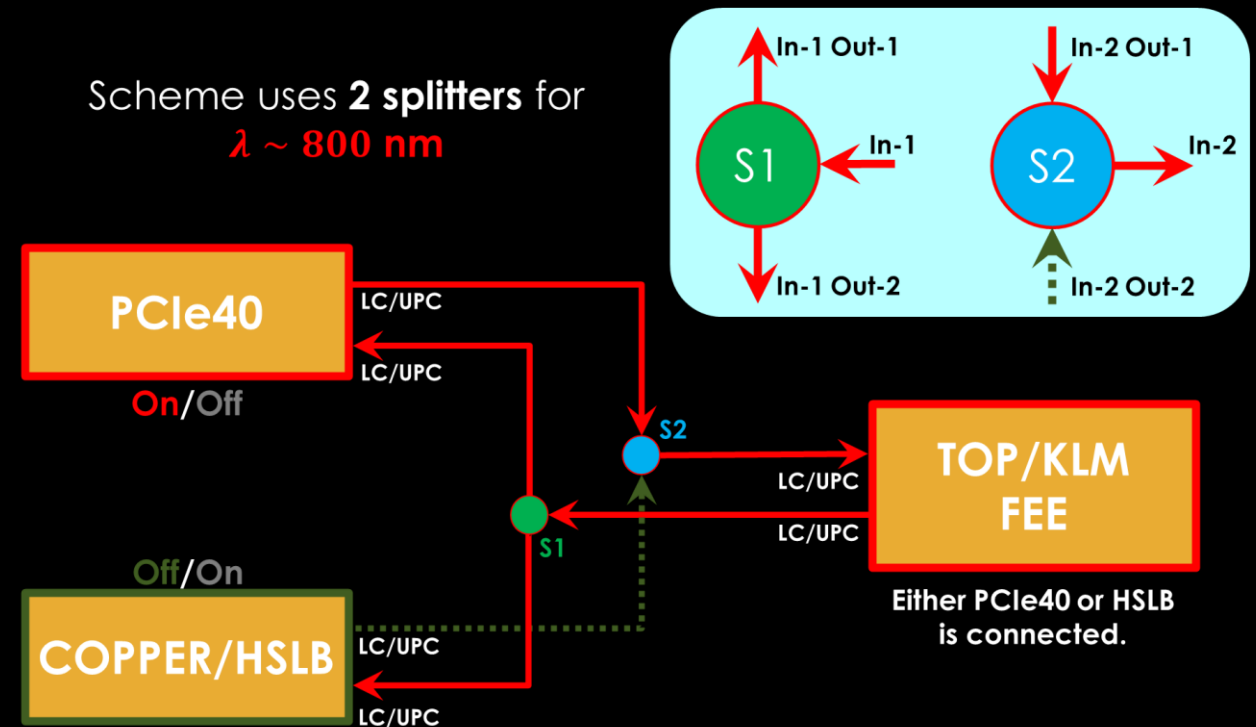
- Switch FEE from COPPER to PCIe40:

**switch top BS3 pcie40**

**switch top 7 pcie40**

**switch klm pcie40**

**switch** assumes default mapping  
PCIe40 channel  $\leftrightarrow$  FEE

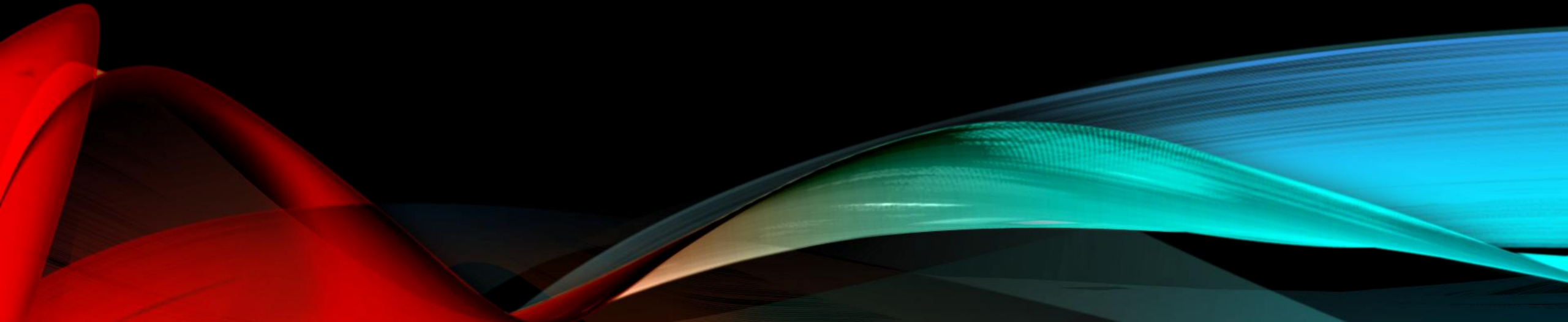




December 16th, 2021

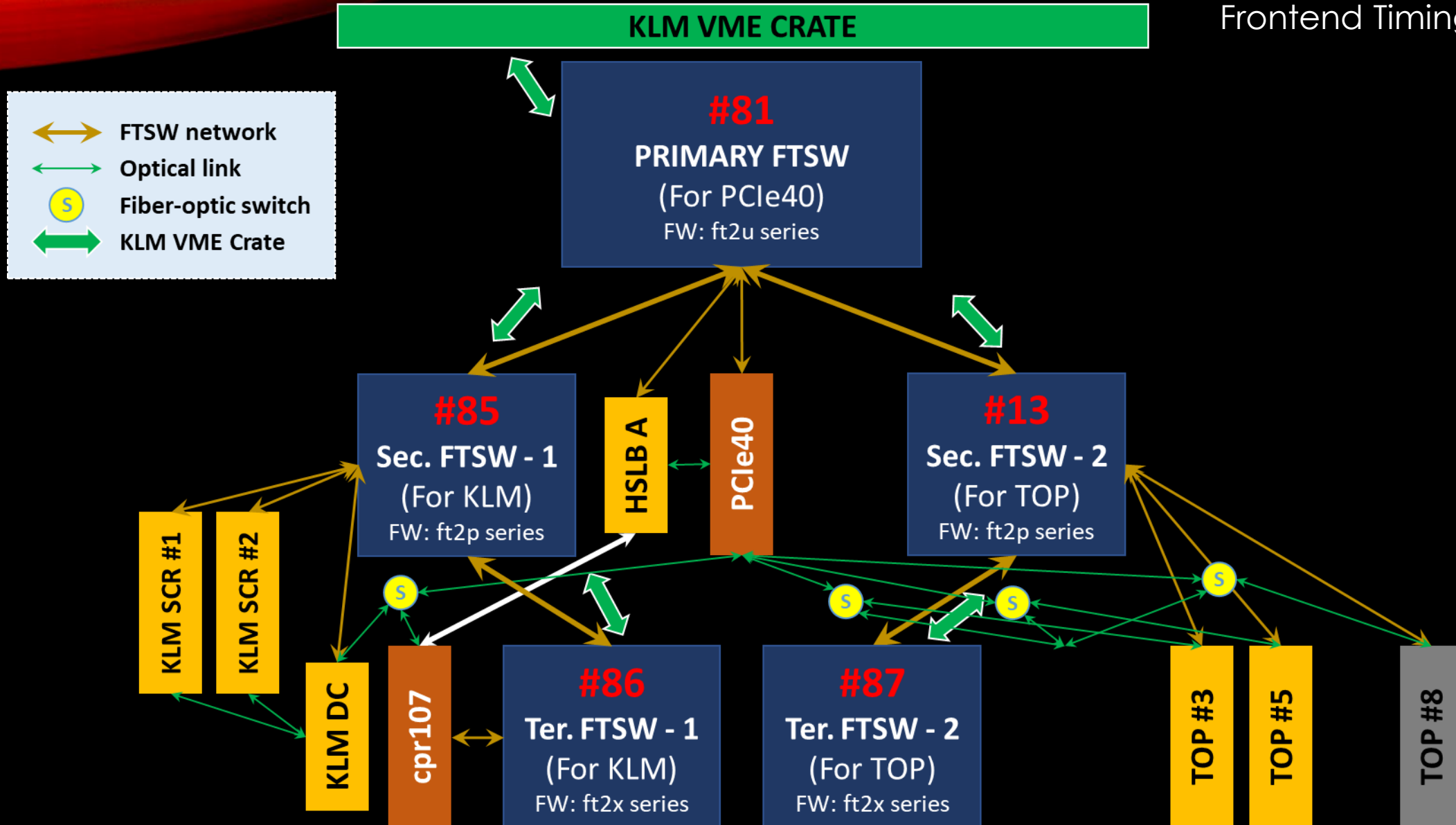
# Introduction to the FTSW Network

Upgraded in 2020, changed slightly again in May 2023.



# ID-Lab FTSW Network

Frontend Timing Switch



# Basic FTSW Commands

SSH to **klmvme** to issue the following commands:

- Status:

```
statft -{ID} -c1
```

- Reset FTSW:

```
resetft -{ID}
```

- Issue triggers:

```
trigft -{ID} --exp=0 --run=0 pulse { $f_{\text{Hz}}$ } { $N_{\text{max}}$ }
```

- Program FTSW:

```
bootft -{ID} {file.bit}
```

A very useful **tmux**-based utility is also available for all users: `/shared/HawaiiTBM.sh`

# HawaiiTBM.sh

```
purwar@daqupsvr:~  
Command window          Running on : p40_fv21pr002  
Firmware version : not implemented, for server : N/A, and number of links : N/A  
Temperatures  
33.7 30.1 39.7 34.4 36.2 38.8 C  
[purwar@daqupsvr ~]$  
  
purwar@daqupsvr:~  
statft version 20210921 FTSW #081 / ft2u090j - 2021.12.15 01:35:01.196  
16 exprun=00000000 exp 0 run 0 sub 0  
17 omask=00000001 s3q=0 selx/o=0/0 x/o=0/01 xor=00  
1f9f jpll=dc008000 clk=xtal GOOD-CLOCK  
28292c trg=00021034 pulse 1000.590 Hz 528e3 limit 0 <-> last 0  
2a2b27 cnt 38437308 > 0 > 0 > 0 (1000.6 > 0.0 > 0.0Hz)  
2d stafifo=10000000 empty trg-DISABLED  
20 reset=c0000000 01.11-20:46:25.436(reset) no-FIFO auto-reset  
31 err=80000000 01.11-20:46:25.436(error)  
25/30 e/bs=0f000000 01f00000 current-limiter GDL-notready  
393a3b me=08100004 0f000000 00000000  
405468 o0!08500000 0f000000 10800700 - LOCAL-mode d=0.00%  
42566a 02=01300000 0a000000 00000600 ready tag=0 d=0.00%  
44586c 04=03040000 0a000000 00000000 ready tag=0 d=0.00%  
9f limiter=0c00b000 maxtrig=12 maxtime=351.44us  
a0-a7 dead 0.00% (t=0.00% c=0.00% p=0.00% f=0.00% r=0.00% v=0.00% i=0.00%)  
  
statft-20210921 FTSW #085 / ft2p092a 2021.12.03-08:53:31 -> 12.15 01:35:00  
16 exprun=00000000 exp 0 run 0 sub 0  
17 omask=00008000 s3q=0 clk=00 lmask=0000 LOCAL  
1f9f jpll=cc008000 clk=in GOOD-CLOCK  
28292c trg=00000000 none limit 0 <-> last 0  
2a2b27 cnt 0 > 0 > 0 > 0 (0.0 > 0.0 > 0.0Hz)  
2d stafifo=10000000 empty trg-DISABLED  
20 reset=80000000 01.11-20:46:24.968(reset) no-FIFO  
31 err=e0000000 01.11-20:46:24.968(error)  
25/30 e/bs=0f000000 01000000  
393a3b me=08500004 0f000000 10800700 min=a98  
485c70 X8=08680000 0a000000 00abcdef ready tag=0 d=0.00%  
495d71 09=0850a000 0a000000 0a000000 ready tag=0 d=0.00%  
4a5e72 010=00000000 0a000000 0a000000 ready tag=0 d=0.00%  
9f limiter=0c00b000 maxtrig=12 maxtime=351.44us  
a0-a7 dead 0.00% (t=0.00% c=0.00% p=0.00% f=0.00% r=0.00% v=0.00% i=0.00%)  
  
statft-20210921 FTSW #013 / ft2p094a 2021.11.30-05:28:12 -> 12.15 02:51:38  
16 exprun=00000000 exp 0 run 0 sub 0  
17 omask=000019ff s3q=0 clk=00 gmask=01ff GLOBAL  
1f9f jpll=cc008000 clk=in GOOD-CLOCK  
28292c trg=00000000 none limit 0 <-> last 0  
2a2b27 cnt 0 > 0 > 0 > 0 (0.0 > 0.0 > 0.0Hz)  
2d stafifo=10000000 empty trg-enabled  
20 reset=80000000 01.11-20:46:26.436(reset) no-FIFO  
31 err=60000000 01.11-20:46:26.436(error)  
25/30 e/bs=0a000000 00000000 tag=0  
393a3b me=01300000 0a000000 00000600 ready tag=0  
485c70 x8!08780000 0a000000 00abcdef - ready tag=0 d=0.00%  
9f limiter=0c00b000 maxtrig=12 maxtime=351.44us  
a0-a7 dead 0.00% (t=0.00% c=0.00% p=0.00% f=0.00% r=0.00% v=0.00% i=0.00%)  
  
[belle2@vme ~]$  
[harsh@topvme ~]$  
  
02:25 15-Dec-21  
[HawaiiTBM@ssh* 1:python2 2:bash- "harsh@topvme:~" 02:25 15-Dec-21]
```

FTSW 86 – dismounted in 2024

FTSW 87 – dismounted in 2024

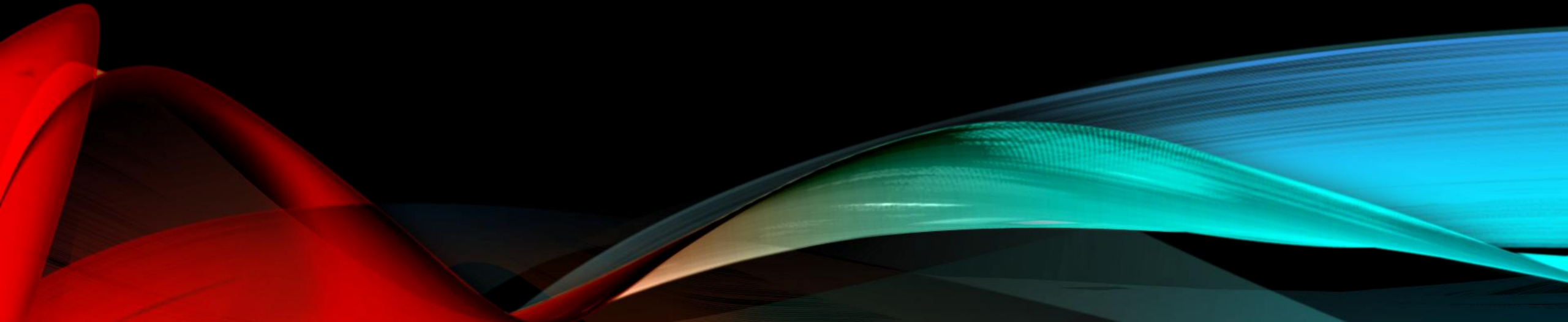
# HawaiiTBM.sh

The image shows a multi-panel terminal window with several active sessions. The top-left panel shows a 'Command window' with system information like 'Running on : p40\_fv21pr002' and 'Firmware version : not implemented'. Below this, a 'statft version 20' command is executed. The top-right panel displays 'Temperatures' for various components, ranging from 33.7 to 38.8 C. The middle-left panel shows the status of PLL SI54345\_1 and SI54345\_2, both indicating they are 'Locked'. The middle-right panel lists 'Voltages and currents' for various power rails, including VCCIN, VCCR, VCCCT, V1.8V, A10\_VCC\_PT, A10\_V1.8, V2.5V, V3.3V, 12V, and 12V\_ATX. The bottom-left panel shows 'Clock FTSW' information, including 'Face plate clock : 127214646 Hz' and 'Run number : 0'. The bottom-right panel shows a list of commands for activating/deactivating channels and groups. The bottom status bar shows the terminal is running on 'Hawaii1BM0' with the command '1:python2 2:python2' and the time '02:25 15-Dec-21'.



# Other available tools

`ttaddr`, `ttaddr_cpr`



# ttaddr/ttaddr\_cpr

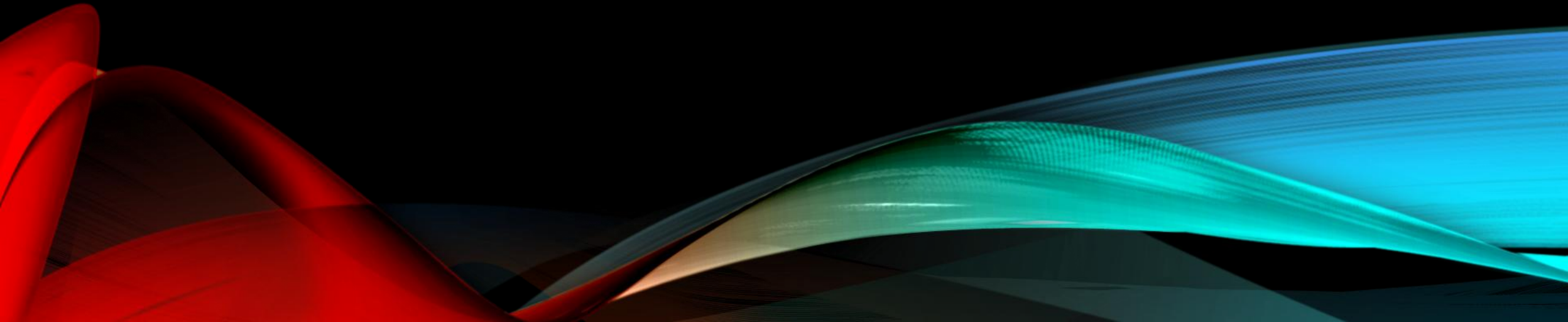
Implemented last week

- ttaddr - used for masking/unmasking FEE/PCIe40 on the FTSW side
- ttaddr\_cpr - used for masking/unmasking FEE/COPPER on the FTSW
- Usage for both commands is the same
- List of important commands (examples):
  - `ttaddr -81 -p` , `ttaddr -85 -p` , `ttaddr -13 -p` -- Print mapping with addresses
  - `ttaddr -81 -c; ttaddr -81 -a;` -- Clear & assign addresses
  - `ttaddr -85 -c; ttaddr -85 -a;`
  - `ttaddr -13 -c; ttaddr -13 -a;`
  - `ttaddr -81 -m top` , `ttaddr -85 -m s01` , `ttaddr -13 -m bs3` -- Mask FEE or TB
  - `ttaddr -85 -{l/g}` -- Switch KLM FTSW to local/global mode (**not for TOP FTSW**)
  - `ttaddr -13 -l` -- Switch FTSW 13 (TOP) to local
  - `ttaddr -13 -g` -- Switch FTSW 13 (TOP) to global

December 16th, 2021

# Data RO with **basf2**

Belle II Analysis Software Framework



# Reading PCIe40 data with basf2

- **Before reading data, ensure correct masking & address assignments on the FTSWs.**
- PCIe40 data (raw) can be readout even without basf2 – `pcie40_dmahighrate`
- For reading formatted/packed data we need to use basf2, which requires a local installation – already done in: `/shared/basf2`
- I prefer to add an alias in `~/.bashrc` then simply type `bs2` to use basf2:  

```
alias bs2='source /shared/basf2/tools/b2setup_old.sh;  
          cd /shared/basf2/release_daq/;  
          b2setup;'
```
- Now open 2 terminals on `daqupsvr` (or ssh to `daqupsvr`)
- Use of `tmux` is highly recommended!

# On Terminal 1

- Run `des_ser_PCIE40_main_{top/klm}`
- `cd /shared/basf2/release_daq/daq/pcie40/  
source Pcie40Software/Scripts/setup.sh`

```
pcie40_ulreset ; ./des_ser_PCIE40_main_{top/klm} {id}
```

```
Processing time in Seconds      : 0
Processing time in Nano seconds : 786008
[DEBUG] (hostname daqupsvr, nodeid 0x03000001 ) concides with stored info.(  daqupsvr 0x03000001 )
[DEBUG] Initializing PCIE40 readout...
SUCCESS: Device opened for ECS 0
SUCCESS: Device opened for ECS 2
SUCCESS: Device opened for DMA
[DEBUG] # of used channels = 1
[DEBUG] PCIE40 readout was initialized.
des_ser_PCIE40_main: Reading the 1st event from a PCIE40 board...
[DEBUG] Accepting... : port 31001
```

| Sub-detector | id         |
|--------------|------------|
| top          | 0x03000001 |
| klm          | 0x07000001 |



# On Terminal 2

- We run basf2 to receive data and write to an output file (important to setup basf2).

- `cd /shared/basf2/release_daq/daq/rawdata/examples/  
basf2 Recv2Root.py -o ~/dataFile.sroot`

```
[INFO] Steering file: Recv2Root.py  
[INFO] DeSerializerPC: Constructor done.  
[INFO] Starting event processing, random seed is set to  
'e1ab05f816f9c1027ee7b9993d0b02a44a3ff15a95c734c8202c73597688a81c'  
[INFO] DeSerializerPC: initialize() started.  
[INFO] DeSerializerPC: initialize() done.  
[DEBUG] Connecting to daqupsvr port 31001 ...  
[DEBUG] Done  
[DEBUG] Initialization finished  
[INFO] DeSerializerPC: Reading the 1st packet from eb0...
```

- On Terminal 1 (if connected):

```
[DEBUG] Accepted.  
Connection(port 31001) accepted  
buff1 = 0x7fcd2037a010
```

If not connected (**not to worry**):

Failed to connect. Retrying...: Connection refused

```
statft version 20210921 FTSW #081 / ft2u090j - 2021.12.15 15:02:56.759
-- RUNNING (about 1000.1Hz since 1970.01.12 20:54:34 for 6s) -----
16 exprun=00000000 exp 0 run 0 sub 0
17 omask=00001fee s3q=0 selx/o=0/15 x/o=1/ee xor=00
1f9f jpll=dc008000 clk=xtal GOOD-CLOCK
28292c trg=00021034 pulse 1000.590 Hz 528e3 limit -1 <-> last 0
2a2b27 cnt 6026 > 0 > 6023 > 0 (1004.3 > 0.0 > 1003.8Hz)
2d stafifo=00000000 some data trg-enabled
20 reset=c0000000 01.12-20:54:34.465(start) no-FIFO auto-reset
31 err=90000000 01.12-20:54:34.462(error) RUNNING
25/30 e/bs=0f000000 00700000 GDL-notready
393a3b me=08100004 0f800000 00000000 mask=none
405468 00=08500000 0a000000 00000000 ready tag=0 d=0.00%
42566a 02!01300000 0f000000 10800100 - LOCAL-mode d=0.00%
44586c 04=03040000 0a000000 00000000 ready tag=0 d=0.00%
9f limiter=0c00b000 maxtrig=12 maxtime=351.44us
a0-a7 dead 0.00% (t=0.00% c=0.00% p=0.00% f=0.00% r=0.00% v=0.00% i=0.00%)
)
```

MASTER

```
statft-20210921 FTSW #085 / ft2p092a 2021.12.03-08:53:31 -> 12.15 15:02:56
-- RUNNING (about 1000.1Hz since 1970.01.12 20:54:35 for 6s) -----
16 exprun=00000000 exp 0 run 0 sub 0
17 omask=000011ff s3q=0 clk=00 gmask=07ff GLOBAL
1f9f jpll=cc008000 clk=in GOOD-CLOCK
28292c trg=00000000 none limit 0 <-> last 0
2a2b27 cnt 0 > 0 > 6025 > 0 (0.0 > 0.0 > 1004.2Hz)
2d stafifo=10000000 empty trg-enabled
20 reset=80000000 01.12-20:54:35.465(start) no-FIFO
31 err=70000000 01.12-20:54:35.462(error) RUNNING
25/30 e/bs=0a000000 00000000 tag=0
393a3b me=08500000 0a000000 00000000 ready tag=0
485c70 x8!08680000 0a001789 00abcdef - ready tag=6025 d=0.00%
495d71 09!0850a000 0a001789 0a001789 - ready tag=6025 d=0.00%
4a5e72 01!0850b000 0a000000 0a000000 - ready tag=0 d=0.00%
9f limiter=0c00b000 maxtrig=12 maxtime=351.44us
a0-a7 dead 0.00% (t=0.00% c=0.00% p=0.00% f=0.00% r=0.00% v=0.00% i=0.00%)
)
```

KLM

```
statft-20210921 FTSW #013 / ft2p094a 2021.11.30-05:28:12 -> 12.15 16:19:31
-- READY (1970.01.12 16:19:35 -> 1970.01.12 16:19:35 for 0s) -----
16 exprun=00000000 exp 0 run 0 sub 0
17 omask=00008000 s3q=0 clk=00 lmask=0000 LOCAL
1f9f jpll=cc008000 clk=in GOOD-CLOCK
28292c trg=00000000 none limit 0 <-> last 0
2a2b27 cnt 0 > 0 > 0 > 0 (0.0 > 0.0 > 0.0Hz)
2d stafifo=10000000 empty trg-DISABLED
20 reset=80000000 01.12-16:57:15.969(reset) no-FIFO
31 err=e0000000 01.12-16:57:15.969(error)
25/30 e/bs=0a000000 01000000 tag=0
393a3b me=01300004 0f000000 10800100 min=8
485c70 X8=08780000 0a000000 00abcdef ready tag=0 d=0.00%
9f limiter=0c00b000 maxtrig=12 maxtime=351.44us
a0-a7 dead 0.00% (t=0.00% c=0.00% p=0.00% f=0.00% r=0.00% v=0.00% i=0.00%)
)
```

TOP

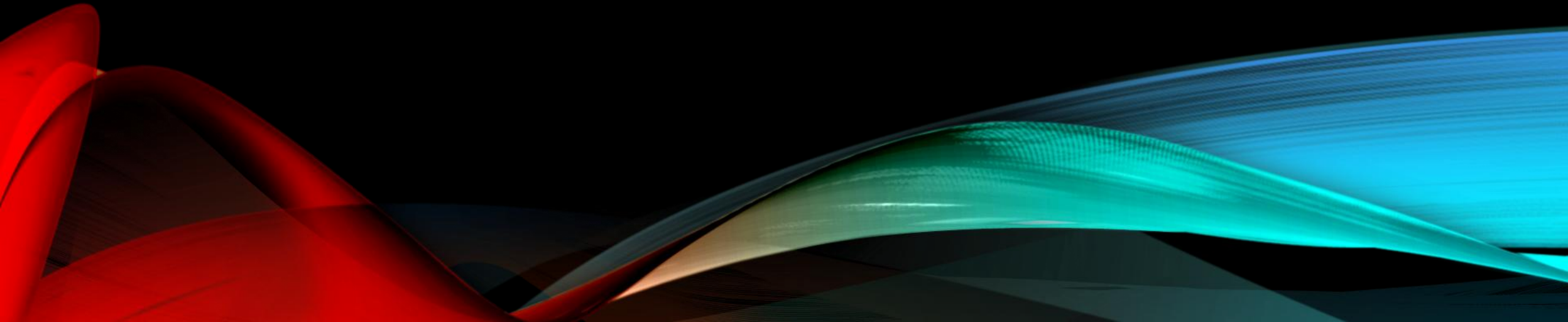
# Now start issuing triggers...

```
trigft -81 --exp=0 --run=0 pulse 500 {N}
```

December 16th, 2021

# Unpacking PCIe40 TOP/KLM Data

Sequential root → root



# Unpacking TOP/KLM Data with basf2

- We need a steering file: a sample is already provided:  
`/shared/basf2/unpackTOPdigi_pcie40.py` and `unpackTOPraw_pcie40.py`
- `unpackTOPdigi_pcie40.py` → Unpacks TOP data to TOPDigits using a fake electronics map for the 2 BS (or SCRODs) we have in the lab (BS3 and BS5).
- `unpackTOPraw_pcie40.py` → Unpacks TOP data to TOPRawDigits – Does not assume or need any electronic map for the SCRODs.
- `basf2 /shared/basf2/unpackTOPdigi_pcie40.py -i ~/dataFile.sroot -o ~/dataFile.root`
- You may open the unpacked `.root` file in `root -l ~/dataFile.root` and use `TBrowser t` over VNC to view the data histograms!

December 16th, 2021

Thank you very much for your time.

- Harsh PURWAR

