

ITk Pixel DAQ using Optoboard-FELIX at LBNL

(Qualification project)

Angira Rastogi, Simone Pagan Griso, Timon Heim

ITk Pixel Read-out meeting
Dec 9th, 2022

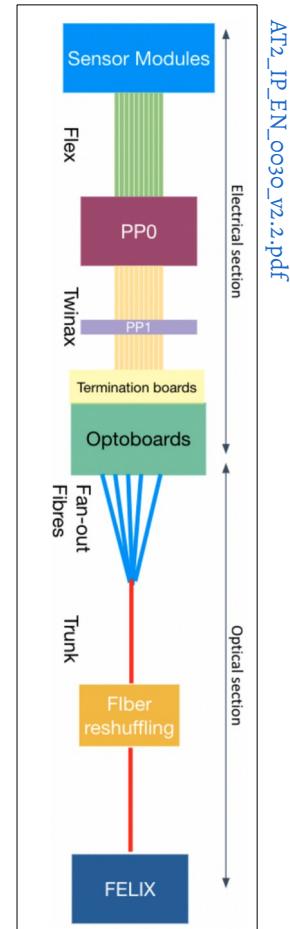


Outline

- Introduction
- Hardware specifications
- Driver, Firmware & Software
- Establishing communication
- Next steps

Introduction

- The HL-LHC is designed to operate at higher collision energy as well as instantaneous luminosity $\sim 5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$.
- To keep up with such rate of collisions, the ATLAS inner detector will be upgraded to be read out using the Front-End Link eXchange (FELIX) system.
- FELIX is a “data router”, receiving packets from detector front-end (FE) electronics and sending them to programmable peers on a high bandwidth network.
- Thus, FELIX improves both the performance as well as maintenance of the full data acquisition (DAQ) chain.



Aim of the project

To set up a ITk Pixel Optoboard-FELIX test stand at LBNL

- For ATLAS TDAQ software-related developments.
- For a more realistic system testing or system-test like infrastructure e.g. for the Pixel Luminosity Ring (PLR).
- Validation of the base framework will be done with the conjunction of this set up with the FELIX test stand for Strips at LBNL.
- Using the optoboard-FELIX interface, reading out the full 8-quad module serial power (SP) chain.
- To test the scalability of the current setup like a realistic detector.

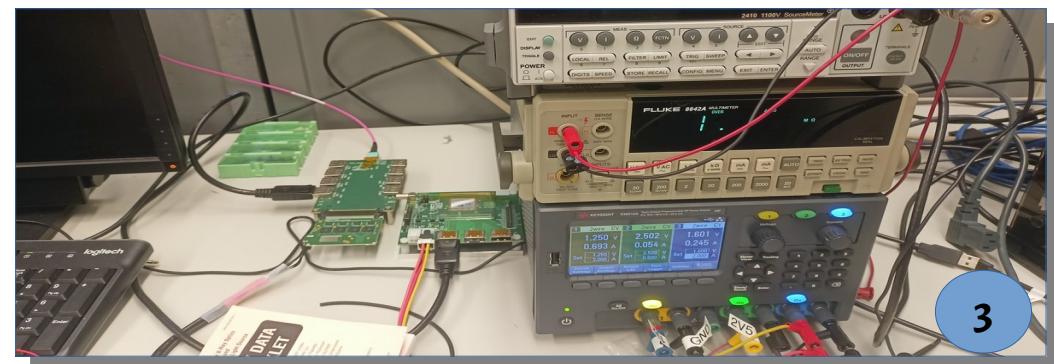
Approach

- Have a working set up using the vanilla recipes for the FELIX firmware and software, and for the optoboard.
- Resources:
 - RD53A readout with FELIX and VLDB+, Marco Trovato et al [CDS link](#)
 - Initial setup instructions for RD53A readout with FELIX and Optoboard, Ismet Siral [CERNbox](#)
 - More recently, ITkPix v1.0 readout with Optoboard (different breakout & adapter board) and FELIX, Daniele Dal Santo [Slides](#)
 - EDMS documents, for more detailed specifications.
- For technical issues, using the thread on JIRA ticket [FLX-2032](#) and also the mattermost “BERN-Optoboard” channel.

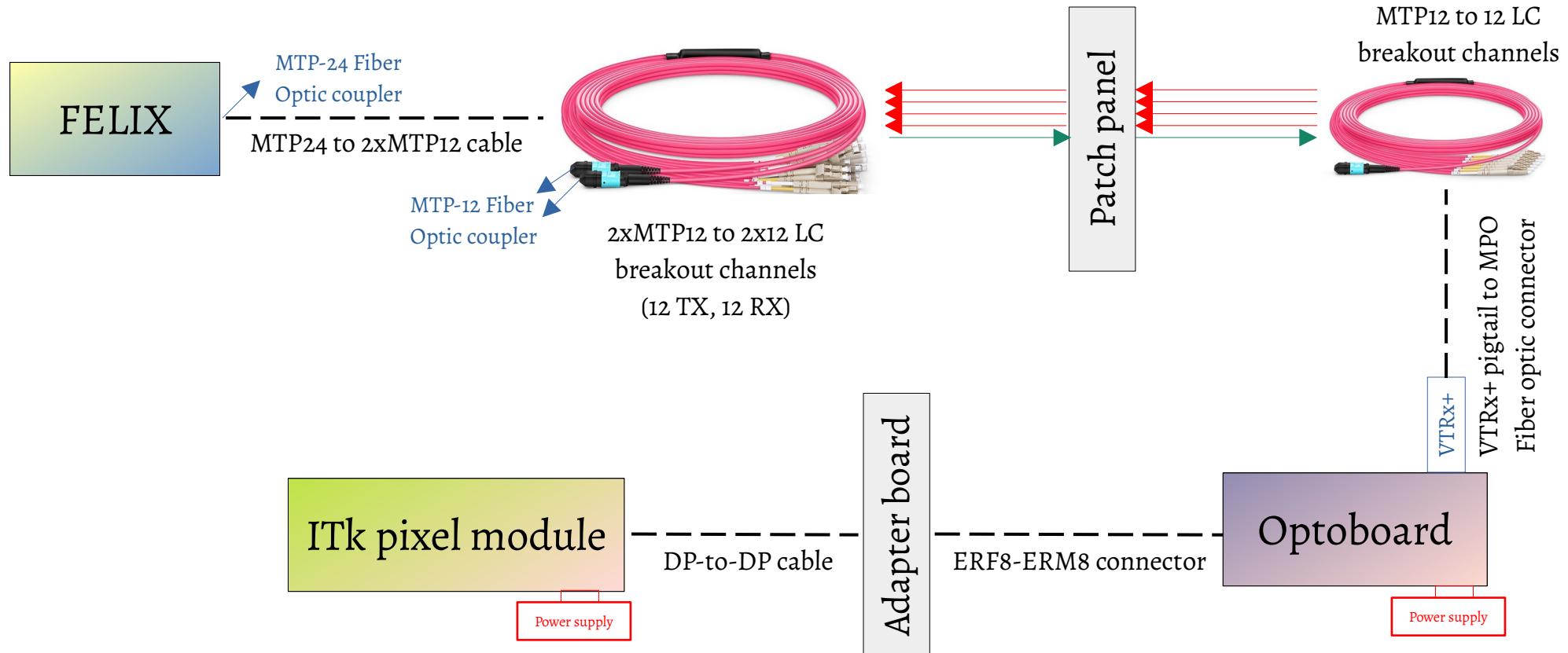
Pixel DAQ setup



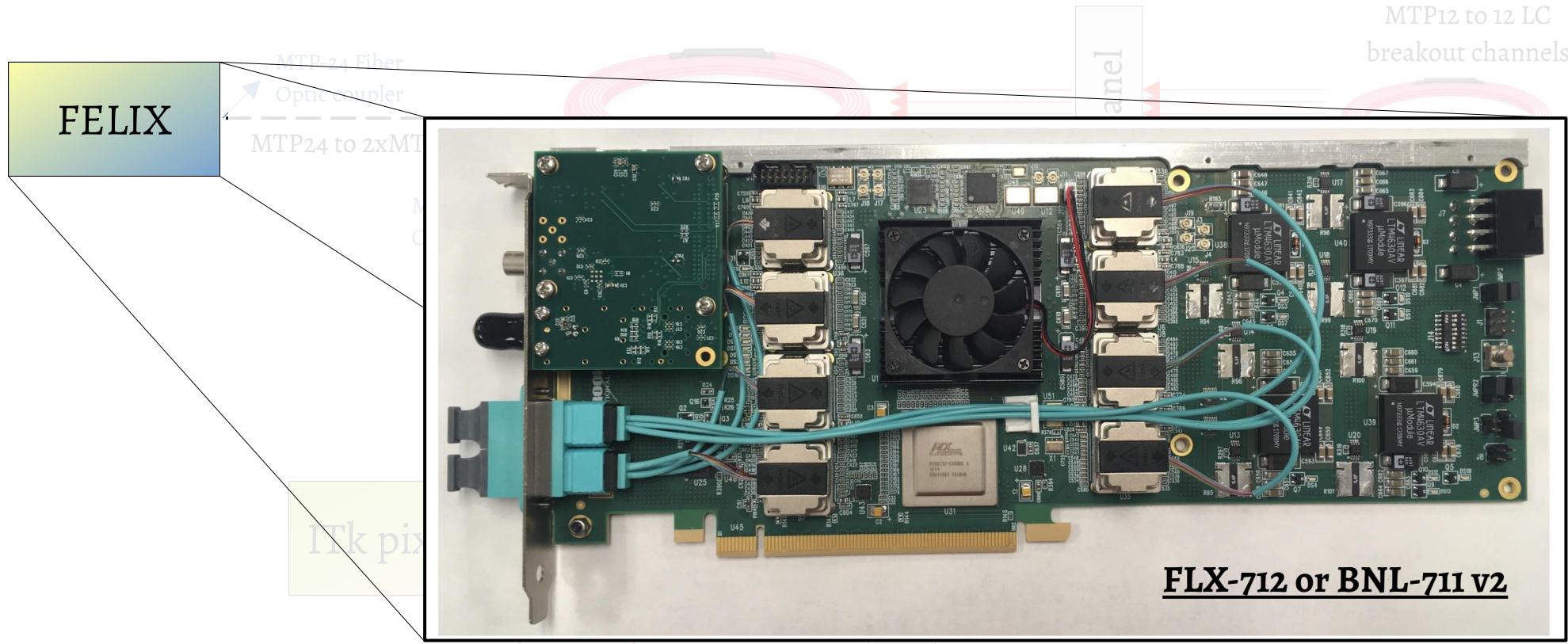
- 1 FELIX card
- 2 Optical fibers
- 3 LpGBT – pixel chip module



Schematic

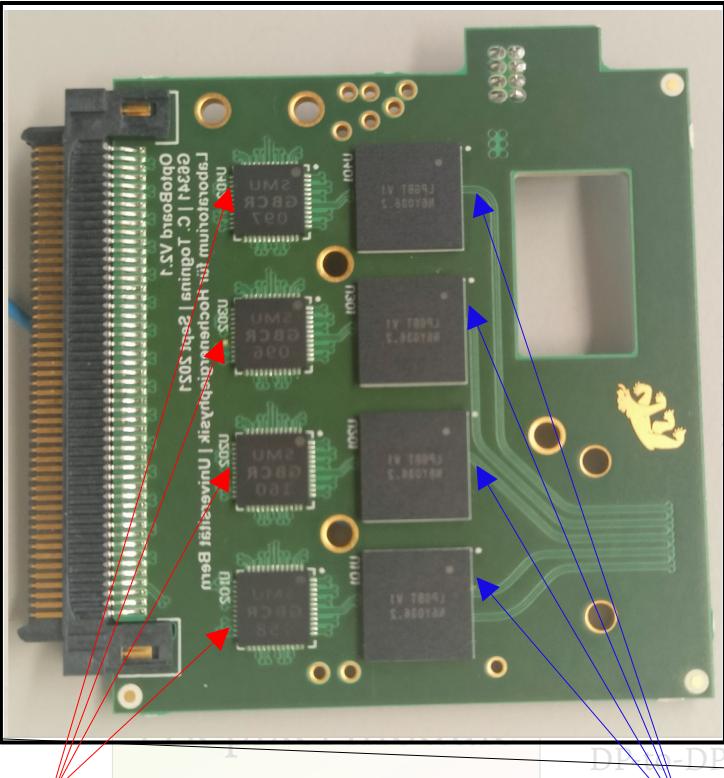


Hardware specifications



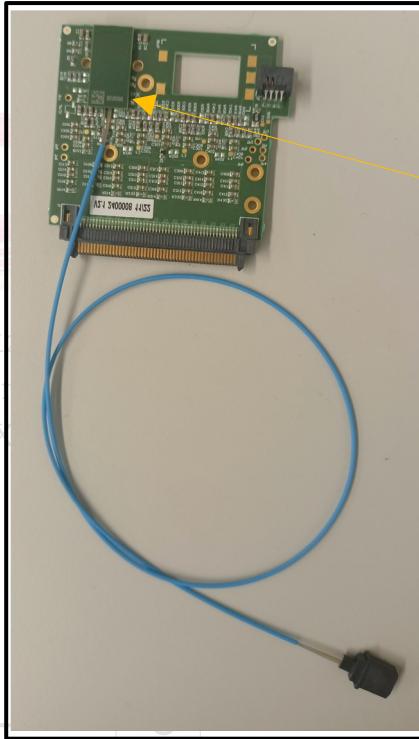
Hardware specifications

FELIX



GBCR v2

LpGBT v1



Optoboard v2.1

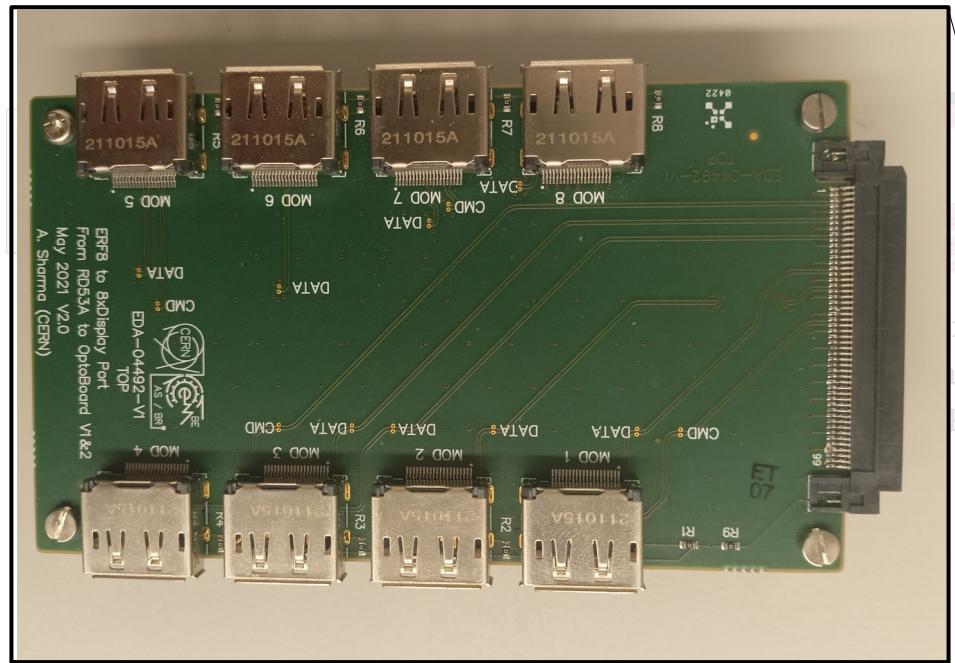
(Doesn't have the BPOL2V5
DC-DC converter chip)

VTRx+ v1.3



Power supply

Hardware specifications



ITk pixel module

Power supply

DP-to-DP cable

Adapter board

ERF8-ERM8 connector

Optoboard

Power supply

ERF8 to DP adapter board v2.0

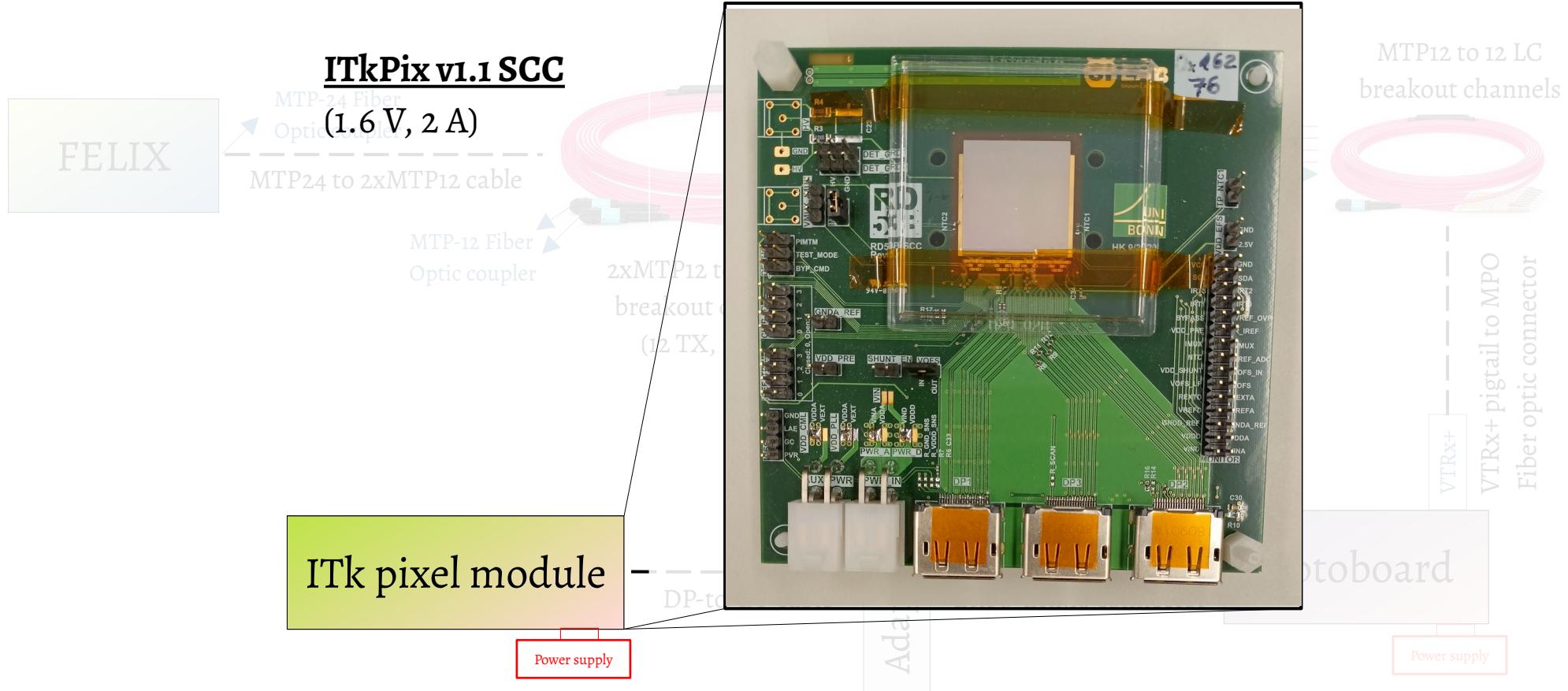
A. Sharma (CERN)

MTP12 to 12 LC
breakout channels



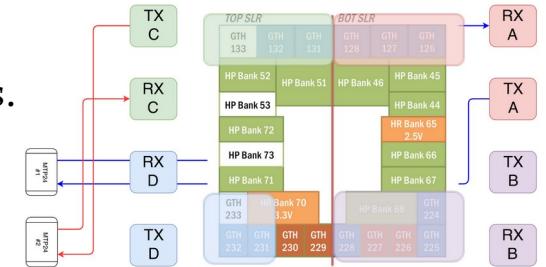
VTRx+
VTRx+ pigtail to MPO
Fiber optic connector

Hardware specifications



FLX-712 card

- 16-lane Gen-3 PCIe card with 48 TX and RX optical links, with a FPGA on-board.
- Has 4 MiniPOD transmitters (TX) and 4 MiniPOD (RX) receivers, each with 12-channels.
 - TX channels have light, upon powering while RX channels receive light from optoboard.
- Two configurations possible: 24-channel (MTP24) and 48-channel (MTP48).
- **At LBNL, we are using the 24-channel configuration via the MTP24 coupler.**
 - TX channels: 1-12, RX channels: 13-24.

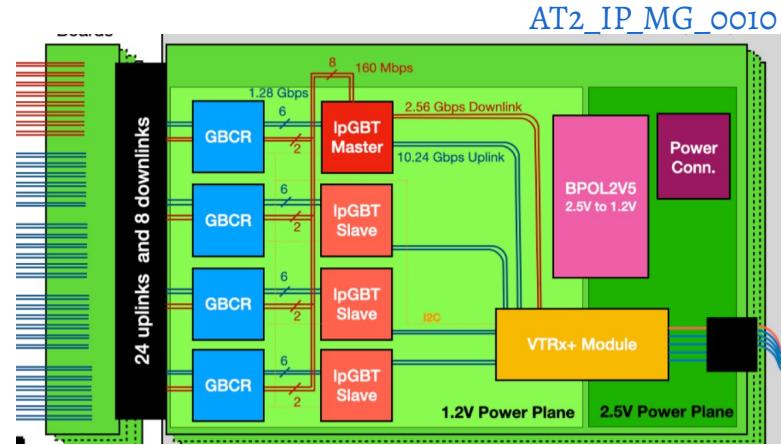


Optoboard v2.1

- Hosts the ASICs that perform the (de-)multiplexing and the electrical-optical (optical-electrical) conversion.
- The main components are:
 - **Four Low power Gigabit Transceivers** (LpGBT) chips, multiplexes the uplink data signals. The main LpGBT chip receives and de-multiplexes the downlink clock and commands and sends back to the detector. LpGBT [manual](#).
 - **Four GBCR chips**, used to recover the uplink signal after the transmission over the twinax cables. GBCR [manual](#).
 - A **VTRx+ quad laser driver** which converts the electrical uplink signal into an optical signal, and vice versa for the downlink signal. Specifications [here](#).

Table 8: VTRx⁺ module optical interface pinout.

VTRx ⁺ Function	Fibre Number
RX	7
TX1	6
TX2	5
TX3	4
TX4	3

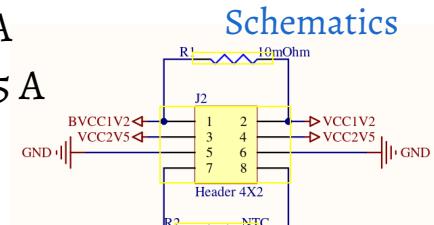


At LBNL, we are setting directly from DC power supply -

Pins 1,2: 1.25 V, 3 A

Pins 3,4: 2.5 V, 0.5 A

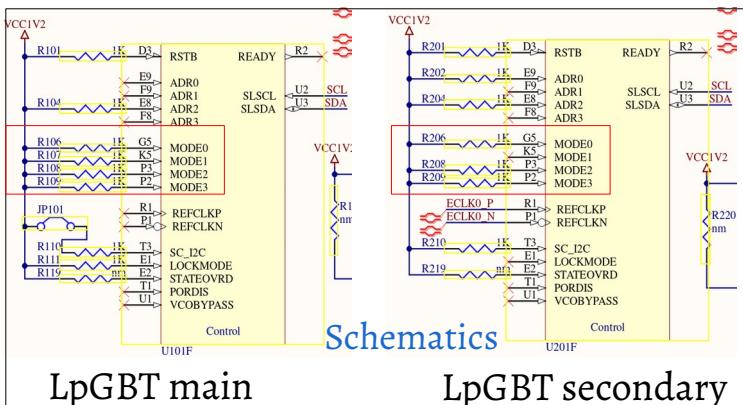
Pins 5,6: GND



BPOL2V5 chip not present.

LpGBT

- High speed bidirectional optical links, 2.56 Gb/s downlink (command, trigger) speed and 5.12 or 10.24 Gb/s uplink (data) speed.
- Connection of Mode[3:0] pins on the LpGBT chips decides the mode of operation of the LpGBT, and also the encoding format along with uplink speed.



Manual

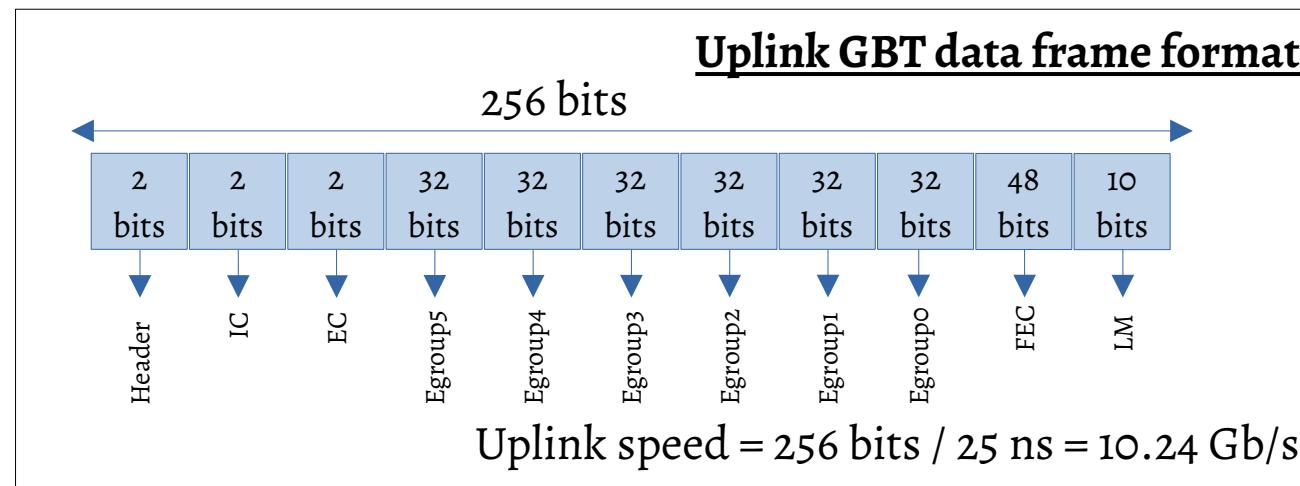


Table 3.1: MODE pins decoding.

MODE [3:0]	Tx Data Rate	Tx Encoding	IpGBT Mode
4'b0000	5 Gbps	FEC5	Off
4'b0001	5 Gbps	FEC5	Simplex TX
4'b0010	5 Gbps	FEC5	Simplex RX
4'b0011	5 Gbps	FEC5	Transceiver
4'b0100	5 Gbps	FEC12	Off
4'b0101	5 Gbps	FEC12	Simplex TX
4'b0110	5 Gbps	FEC12	Simplex RX
4'b0111	5 Gbps	FEC12	Transceiver
4'b1000	10 Gbps	FEC5	Off
4'b1001	10 Gbps	FEC5	Simplex TX
4'b1010	10 Gbps	FEC5	Simplex RX
4'b1011	10 Gbps	FEC5	Transceiver
4'b1100	10 Gbps	FEC12	Off
4'b1101	10 Gbps	FEC12	Simplex TX
4'b1110	10 Gbps	FEC12	Simplex RX
4'b1111	10 Gbps	FEC12	Transceiver

LpGBT...cont'd

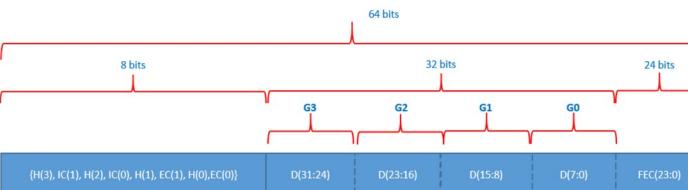
Manual

- Each egroup can have up to four epaths (a.k.a. channels), depending on what data speed is selected.

Registers EPRX[0,1,2,3,4,5] DataRate = 3 (1.28 Gb/s)

- Downlink GBT data frame format: Fixed at 64 bits.

Registers EPTX[0,1,2,3] DataRate = 2 (160 Mb/s)



Data length = 32 bits
Egroups = 4 (max. 8 bits)

Table 7.1: ePortRx (uplink) data rates					
TxRate Gb/s	Data Select	Data Rate Mb/s	Links per group	Active Channels	Frame-channel mapping
5.12	2'b00	0	0	none	{16'b0}
5.12	2'b01	160	4	0, 1, 2, 3	{chn3[3:0], chn2[3:0], chn1[3:0], chn0[3:0]}
5.12	2'b10	320	2	0, 2	{chn2[7:0], chn0[7:0]}
5.12	2'b11	640	1	0	{chn0[15:0]}
10.24	2'b00	0	0	none	{32'b0}
10.24	2'b01	320	4	0, 1, 2, 3	{chn3[7:0], chn2[7:0], chn1[7:0], chn0[7:0]}
10.24	2'b10	640	2	0, 2	{chn2[15:0], chn0[15:0]}
10.24	2'b11	1280	1	0	{chn0[31:0]}

Table 7.2: ePortTx (downlink) data rates

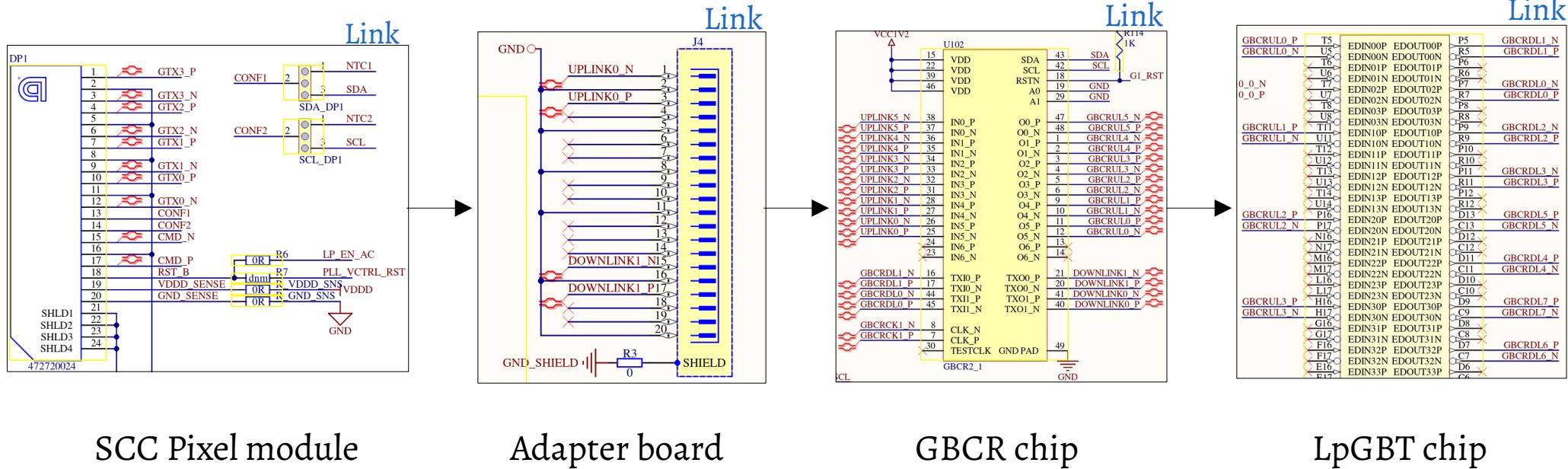
Data Rate Select	Data Rate Mb/s	Links per group	Active Channels	Frame-channel mapping
2'b00	0	0	none	{8'bx}
2'b01	80	4	0, 1, 2, 3	{chn3[1:0], chn2[1:0], chn1[1:0], chn0[1:0]}
2'b10	160	2	0, 2	{chn2[3:0], chn0[3:0]}
2'b11	320	1	0	{chn0[7:0]}

To summarize, at LBNL test stand:

- LpGBT is operated in FEC12 encoding mode, at uplink speed of 10.24 Gb/s.
- Encoding to be enabled for epath=0,2 for all egroups, and epath width = 4 bits (160 Mb/s downlink).
- Decoding to be enabled for epath=0 for all egroups, and epath width = 32 bits (1.28 Gb/s uplink).

Pin-mapping for polarity

Let's try to trace UPLINK0 and DOWNLINK1...



SCC Pixel module

Adapter board

GBCR chip

LpGBT chip

Pin-mapping for polarity

Let's try to trace UPLINK0 and DOWNLINK1...

FELIX Link	Pixel chip (pin #)	Adapter board (pin #)	GBCR pin	LpGBT pin
Up eLink (N)	GTX3_P (1)	Uplink0_N (1)	ULo_P	EDINooP
Up eLink (P)	GTX3_N (3)	Uplink0_P (3)	ULo_N	EDINooN
Down eLink (N)	CMD_N (15)	Downlink1_N (15)	DL1_N	EDOUTooP
Down eLink (P)	CMD_P (17)	Downlink1_P (17)	DL1_P	EDOUTooN

- Polarity of uplink data from pixel chip is inverted in the FELIX firmware.
- Hence, at LBNL, we set EPRX[G][C]Invert to zero in the optoboard configuration (due to this adapter board), where G=egroup number and C=epath number. Finally, the uplink data polarity is inverted in the FELIX firmware.
- The downlink polarity from FELIX is inverted twice before reaching the pixel chip. Therefore, no change is required.

Driver, Firmware & Software

- **FELIX driver**

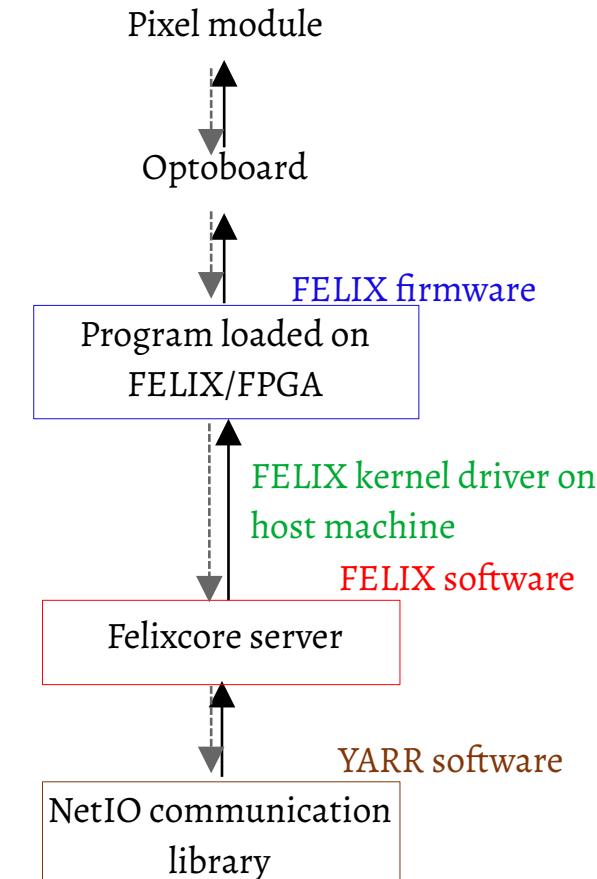
- For ATLAS PhaseII upgrade, current recommended driver version is 4.9.1
- Using: **tdaq_sw_for_Flx-4.9.0-2dkms.noarch**
- List of available drivers - [here](#).

- **FELIX firmware**

- Firmware bit files can be found - [here](#).
- Using: **FLX712_PIXEL_4CH_CLKSELECT_GIT_phase2-master_FLX-2038_LPGBTalignmentFEC12_rm-5.0_2749_221101_11_29.bit**
- Using Vivado 2022.1 to upload the firmware and program the memory (as described [here](#)).

- **FELIX software**

- ATLAS TDAQ FELIX software, master branch with the git patch FLX-1613 (needs cvmfs, if not using precompiled binary software version.)
- Git repository: [here](#), instructions in the README.



Driver, Firmware & Software

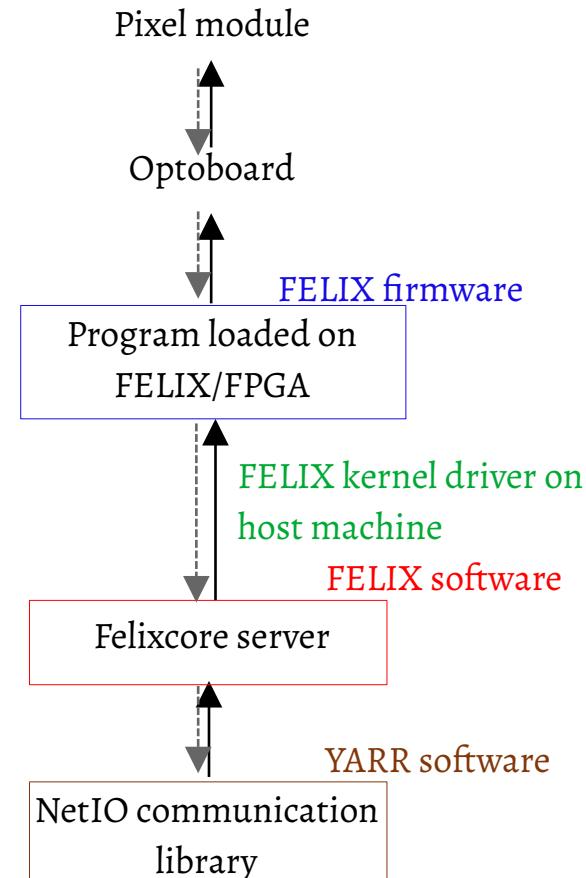
Additionally,

- **YARR software**

- For performing readout from the chip module using NetIO library.
- Git branch: “devel_itkpix_felixNetio” [here](#)
- Changes w.r.t. “devel” branch: bypass register reading, slightly different reset procedure, and firmware-based trigger generation.

- **Optoboard software**

- For configuring optoboard via the Felixcore server.
- Git branch: “reorganization” [here](#). Instructions in the README file.
- Extra [ICHandler library](#) (from Ismet) to be added to FELIX software and recompile to be able to communicate with optoboard using Felixcore.



Initial setup

- Once the FELIX driver is successfully installed and is running, it would be able to see the connected FELIX cards (or devices).
- The correct version of the firmware flashed via Vivado tool can also be checked at the same time.

- Once the FELIX software is also installed, we can proceed to configuring the FELIX card to establish the link alignment with the main GBT from Optoboard.

```
(python_env) -bash-4.2$ cat /proc/flx
FLX driver for FELIX release 4.9.0 (compatible with RM4 and RM5 F/W). Based on tag ROSRCDdrivers-00-01-04, flx.c revision 4.9.0

Debug = 0
Number of devices detected = 4

Locked resources
device | global_locks
=====|=====
0 | 0x00000000
1 | 0x00000000
2 | 0x00000000
3 | 0x00000000

Locked resources
device | resource bit | PID | tag
=====|=====|=====|=====
Device 0: (BAR0 = 0xfaaa00000)
Card type : FLX-712
Device type : 0x0427
FPGA_DNA : 0x013a6f281d412105
Reg Map Version : 5.0
GIT tag : rm-5.0
BUILD Date and time : 1-11-2022 at 11h29
GIT commit number : 2749
GIT hash : 0x85f6366c
F/W partition (jumpers) : 3
Firmware mode : ITK PIXEL
Number of descriptors : 3
Number of interrupts : 8
Interrupt count | 3065867 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
Interrupt flag | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
Interrupt mask | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
MSI-X PBA 00000000
```

FELIX configuration

- Setting up the FELIX software environment (reg map 5.0, LCG 101 from cvmfs, python env).
- Starting the GBT links on the FELIX firmware to receive light signal from the optoboard.
- Enabling the proper egroup & epath structure, and also setting their encoding and decoding.
- Inverting the polarity of each GBT link (or fiber channels) from the LpGBT.
- Setting up the FEC12 encoding to select the data rate speed.

After this, the alignment should be present between FELIX and main LpGBT chip of the optoboard.

```
-----+---+  
(python_env) bash-4.2$ flx-info -c 0 LINK  
Card type : FLX-712  
Firmw type: PIXEL  
  
Link alignment status  
-----+---+  
Channel | 0 1  
-----+---+  
Aligned | YES NO  
  
Channel | 2 3  
-----+---+  
Aligned | NO NO
```

At this stage, optoboard channel
1.25 V draws ~0.69 A of current.

```
source setup.sh  
flx-init -c 0  
source encoding_decoding.sh 0  
fgpolarity -c 0 -r set  
flx-config set LPGBT_FEC=0xF
```

Optoboard configuration

- Once the alignment is present with the main LpGBT, we can configure all the other LpGBT chips and GBCR chips and the VTRx+ (will have light on all four TX fibers).

Start the Felixcore server:

```
./x86_64-centos7-gcc11-opt/felixcore/felixcore -d o --data-interface lo --toflx-tlp 64 -v -p 12350 -r 12340 -P 12330 -w 8080
```

Configure the optoboard: [config](#)

```
python -i InitOpto.py -config_path "configs/optoboard_lpgbtv1_gbcr2_vtrxv1_3_default.json" -optoboard_serial "oooooooo" -vtrx_v "1.3" -FELIXFOLDER "/path/to/software/" -YARRFOLDER "/path/to/YARR" -configure 1
```

Optoboard configuration

- Once the alignment is present with the main LpGBT, we can configure all the other LpGBT chips and GBCR chips and the VTRx+ (will have light on all four TX fibers).

```
... ...
(python_env) bash-4.2$ flx-info -c 0 LINK
Card type : FLX-712
Firmw type: PIXEL

Link alignment status
-----
Channel | 0   1
-----  

Aligned | YES YES

(python_env) bash-4.2$ flx-info -c 0 POD
Card type : FLX-712
Firmw type: PIXEL

MiniPODs status
-----
| 1st TX | 1st RX | 2nd TX | 2nd RX | 3rd TX | 3rd RX | 4th TX | 4th RX |
|=====|=====|=====|=====|=====|=====|=====|=====|
Temperature [C] | 49 | 50 | 43 | 41 | ----- | ----- | ----- | ----- |
3.3 VCC [V] | 3.25 | 3.26 | 3.26 | 3.26 | ----- | ----- | ----- | ----- |
2.5 VCC [V] | 2.44 | 2.41 | 2.46 | 2.43 | ----- | ----- | ----- | ----- |

Optical power (Receive=RX or Transmit=TX) of channel in uW:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|
1st TX | 1036.10 | 1027.30 | 964.40 | 996.60 | 984.00 | 980.30 | 1015.90 | 980.30 | 986.80 | 970.30 | 962.70 | 881.10
1st RX | 832.10 | 812.00 | 728.60 | 771.70 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00
2nd TX | 1286.20 | 1270.60 | 1236.30 | 1196.90 | 1276.20 | 1265.80 | 1276.30 | 1123.70 | 1263.00 | 1173.30 | 1154.70 | 1188.10
2nd RX | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00
3rd TX | <not detected>
3rd RX | <not detected>
4th TX | <not detected>
4th RX | <not detected>

How to read the table below:
* = FLX link endpoint OK (no LOS)
- = FLX link endpoint not OK (LOS)
First letter: Current channel status
Second letter: Latched channel status
Example: *(-) means channel had lost the signal in the past but the signal is present now.

Latched / current link status of channel:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|
1st TX | *(*) | *(*) | *(*) | *(*) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-)
1st RX | *(*) | *(*) | *(*) | *(*) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-)
2nd TX | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-)
2nd RX | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-) | -(-)
```

All four GBT links (TX) are aligned.

FELIX MiniPOD status... receives light on four RX channels from Optoboard.

```
2022-12-08 10:08:07,198 - INFO - Initialising config from configs/optoboard_lpgbtv1_gbcrt2_vtrv1_3_default.json
2022-12-08 10:08:07,199 - INFO - Comm_wrapped object initialised!
2022-12-08 10:08:07,199 - INFO - Logger level at INFO
2022-12-08 10:08:07,199 - INFO - component: OrderedDict([('optoboard_v', 2), ('lpgbt_v', 1), ('lpgbt_master_addr', 'gbcrt2'), 1), ('gbcrt3', 1), ('gbcrt4', 1), ('efused', 0), ('i2c_master', 0)])
2022-12-08 10:08:07,200 - INFO - lpgbt1 object initialised!
2022-12-08 10:08:07,200 - INFO - lpgbt2 object initialised!
2022-12-08 10:08:07,200 - INFO - lpgbt3 object initialised!
2022-12-08 10:08:07,200 - INFO - lpgbt4 object initialised!
2022-12-08 10:08:07,200 - INFO - gbcrt1 object initialised!
2022-12-08 10:08:07,200 - INFO - gbcrt2 object initialised!
2022-12-08 10:08:07,201 - INFO - gbcrt3 object initialised!
2022-12-08 10:08:07,201 - INFO - gbcrt4 object initialised!
2022-12-08 10:08:07,201 - INFO - vtrx object initialised!
2022-12-08 10:08:07,201 - INFO - Optoboard object initialised!
2022-12-08 10:08:07,201 - INFO - The optoboard object 'opto' is now available!
2022-12-08 10:08:07,201 - INFO - The felix object 'Felix' is now available!
2022-12-08 10:08:07,208 - INFO - The felix object 'Yarr' is now available!
2022-12-08 10:08:07,208 - INFO - Starting configuration of the board
/bin/sh: ./setup.sh: No such file or directory
2022-12-08 10:08:08,210 - INFO - Configuring lpgbt1..
2022-12-08 10:08:09,354 - INFO - Configuring I2C controller settings on I2C master 0 and reset..
2022-12-08 10:08:09,369 - INFO - Reset the I2C master: generating a pulse 0->1->0 on bit RSTI2CM0
2022-12-08 10:08:09,419 - INFO - I2C controller 0 settings done
2022-12-08 10:08:09,423 - INFO - lpgbt1 status is READY
2022-12-08 10:08:09,423 - INFO - Configuring lpgbt2..
2022-12-08 10:08:19,393 - INFO - lpgbt2 status is READY
2022-12-08 10:08:19,394 - INFO - Configuring lpgbt3..
2022-12-08 10:08:29,334 - INFO - lpgbt3 status is READY
2022-12-08 10:08:29,335 - INFO - Configuring lpgbt4..
2022-12-08 10:08:39,241 - INFO - lpgbt4 status is READY
2022-12-08 10:08:39,242 - INFO - Configuring gbcrt1..
2022-12-08 10:08:40,084 - INFO - Configuring gbcrt2..
2022-12-08 10:08:40,918 - INFO - Configuring gbcrt3..
2022-12-08 10:08:41,747 - INFO - Configuring gbcrt4..
2022-12-08 10:08:42,582 - INFO - Configuring vtrx..
2022-12-08 10:08:42,582 - INFO - Using VTRx+ quad laser driver v1.3, enabling all TX fibre channels..
```

Optoboard fully configured.

We are all set to talk with the pixel chip...

For the optoboard channel 1.25 V, current changes to ~1.34 A.

Pixel chip configuration

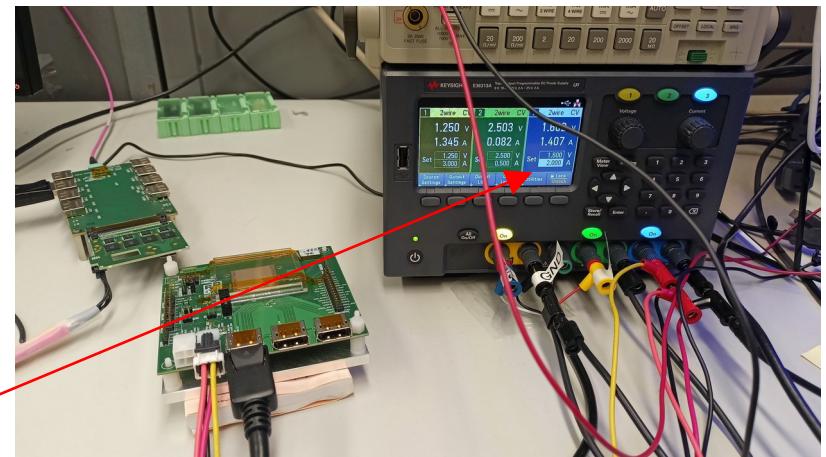
- Tx and Rx port numbers correspond to the egroup-epath structure of the GBT link (fiber channels), and can be found out using the felink command.
 - For ex, “felink -d 0 -G 0 -g 0 -p 0” returns the correct eLink number (HEX, convert to decimal) to be used in the connectivity config.
- At LBNL, using port J5 on adapter board (i.e. UPLINK0, DOWNLINK1), Tx = 0 and Rx = 0.

CdrClkSel = 0 (to select clock speed of 1.28 Gb/s, for both RD53A & ITkPix).

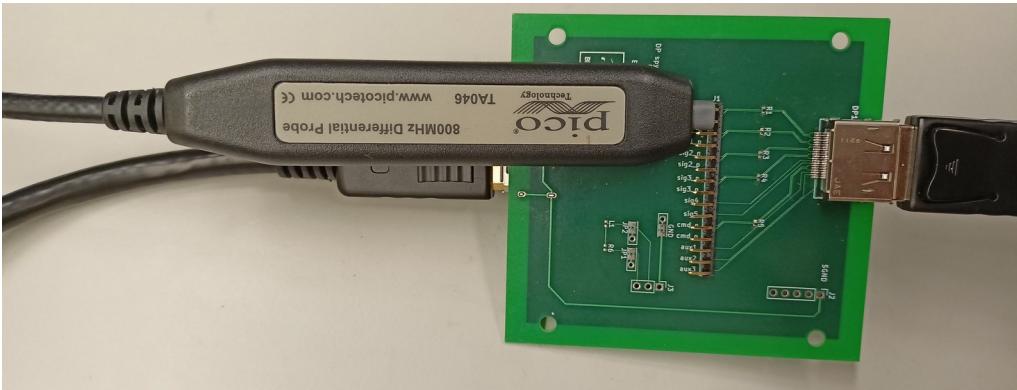
DataMergeMUXOut (0/1/2/3) = 0 (duplicating data from chip on all the lanes).

AuroraActiveLanes = 1 (reading out only one lane).

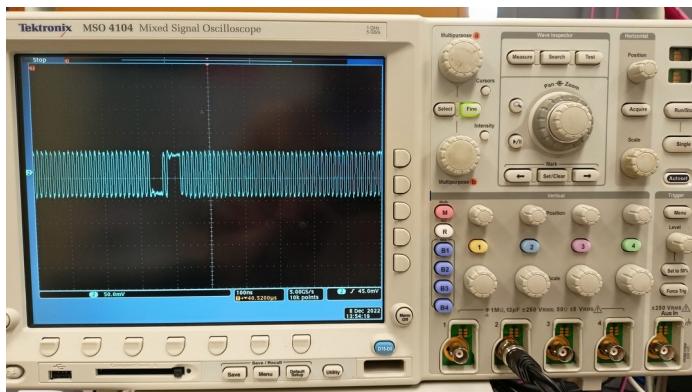
Chip successfully configured!



Spy card for the Pixel chip

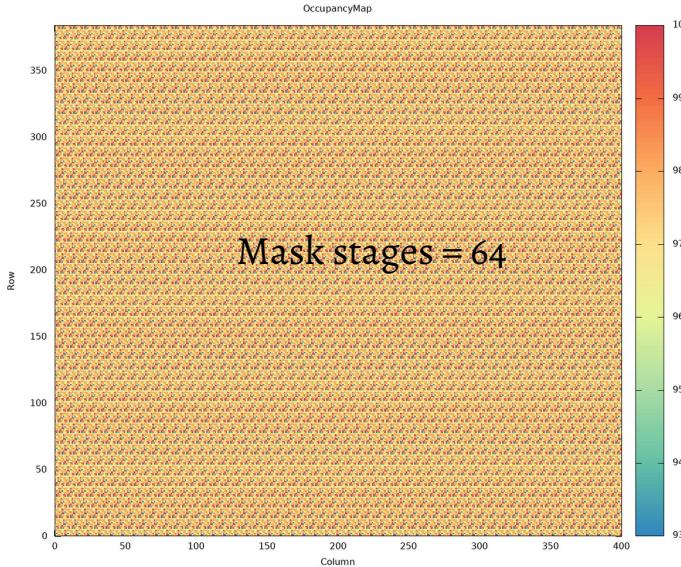
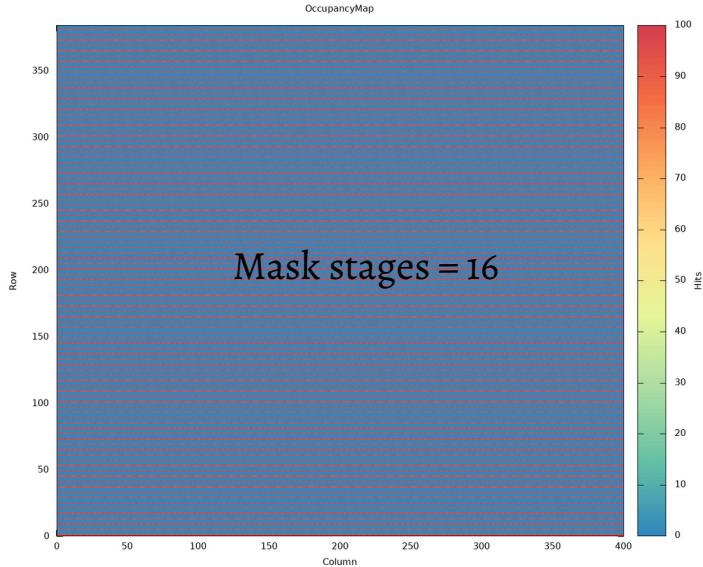


- Created at LBNL, plenty available if interested.
- Helps in tapping at the command and data lanes directly from the chip (no probe pins on the SCC).
- Can be very useful for debugging signals.
- Polarity of the data lanes is inverted between the two DP ports.
- Hence, needs to be taken care of, especially when running a YARR scan.



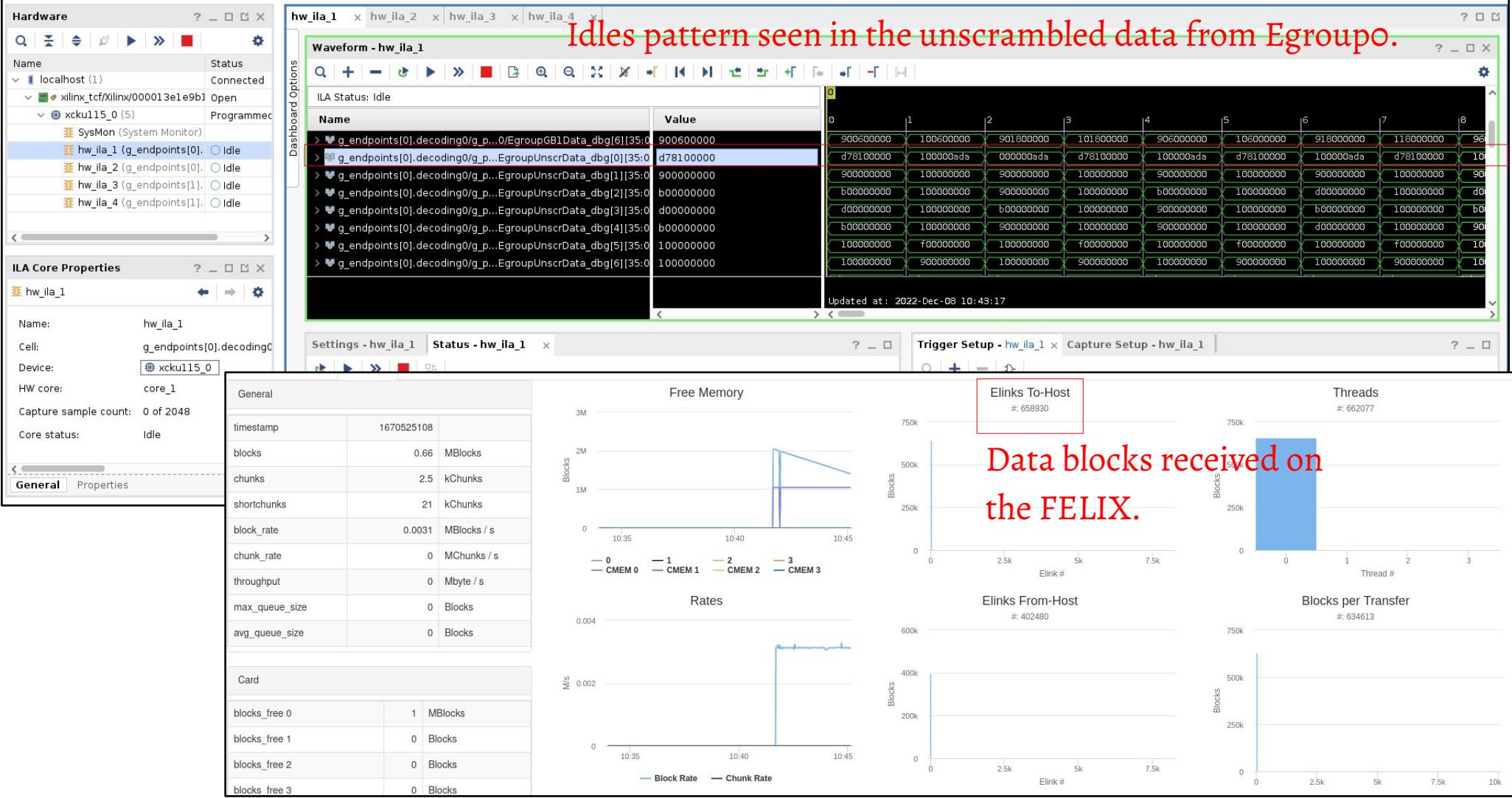
FELIX command idles for the pixel firmware also seen.

Running a digital scan...



- Running the scans has been a bit unreliable, often some data from the chip is lost. Felixcore server also crashes unexpectedly.
- Results of the digital scan with step size = 4 (16 mask stages) seems sensible.
- However, for digital scan with step size = 1 (64 mask stages) has interesting pattern, probably some data from the chip is missing.

- Changed the firmware-based trigger to software-based in [RD53bTriggerLoop](#).
- Added wait time of 5 seconds in the [NetioTxCore](#) library, to introduce latency.
- Trigger frequency = 100 and trigger count = 100.



Next steps

- To debug the issue of missing chip data in the YARR scans.
- Switch to a pixel quad module, and repeat the exercise.
- Scale up the system to use the 8-pixel quad module SP-chain.
 - Need another optoboard for that, and also design a new PCB for the adapter board connection.
- Modify the firmware to be able to read the register frames from the chip.
- Controller-specific changes for the hard reset settings.