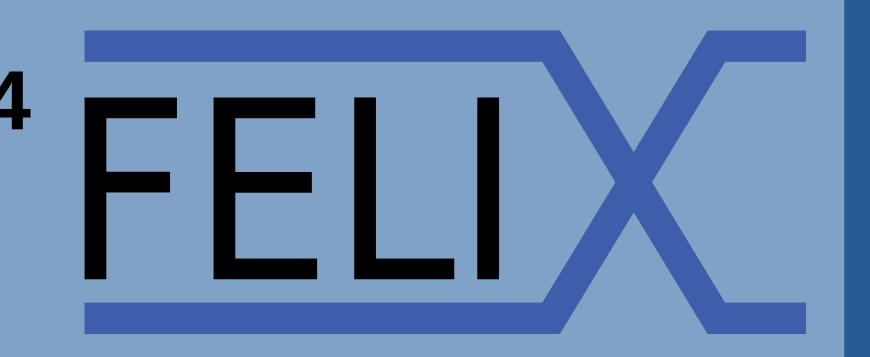
# FELIX Phase-II, the ATLAS readout system for LHC Run 4

# Melvin Leguijt on behalf of the ATLAS TDAQ Collaboration



### **Abstract**

The Front End Link eXchange (FELIX) system, initially deployed for ATLAS in LHC Run 3, will evolve for Run 4, serving all subdetectors. The system will consist of 350 servers with new custom PCle FELIX cards and 200 GbE interfaces, handling data at 1 MHz readout rate with a total throughput of 4.6 TB/s. The new PCle cards, featuring an AMD Versal Premium FPGA/SoC and advanced connectivity, run upgraded firmware to decode data and manage timing and control information. The server and software will also evolve to adapt to the increased trigger rate and data throughput. Preliminary reviews in 2022 and 2024 confirmed the validity of the firmware and card design.

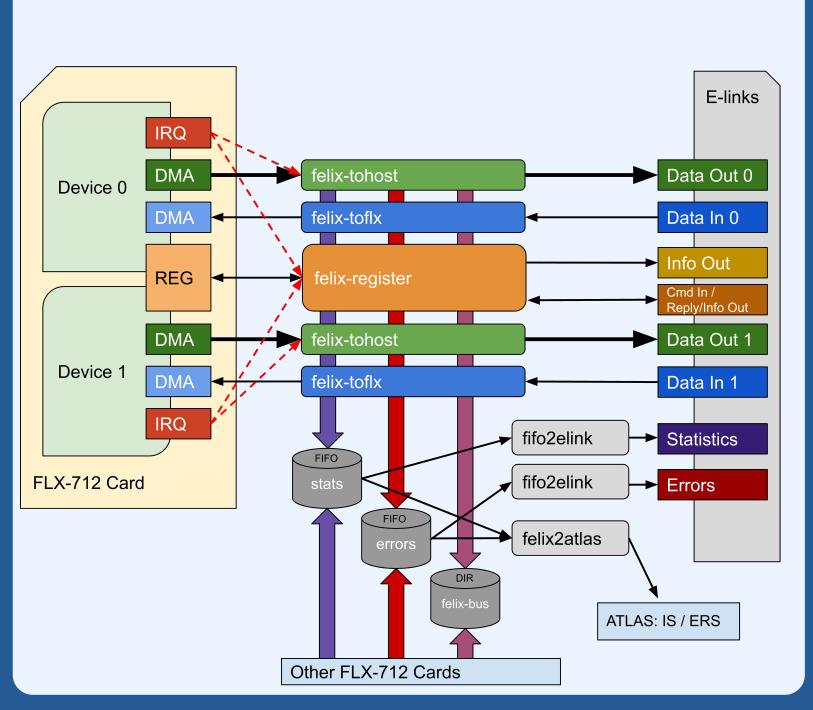
### Software

### DAQ software:

Several applications running on the host server communicate via PCIe with FELIX FPGA cards and via Ethernet with servers for handling event data and for monitoring, calibration and control.

### FPGA

The FPGA includes an ARM processor that hosts a built-in self-test and "knypaegje" (see below)

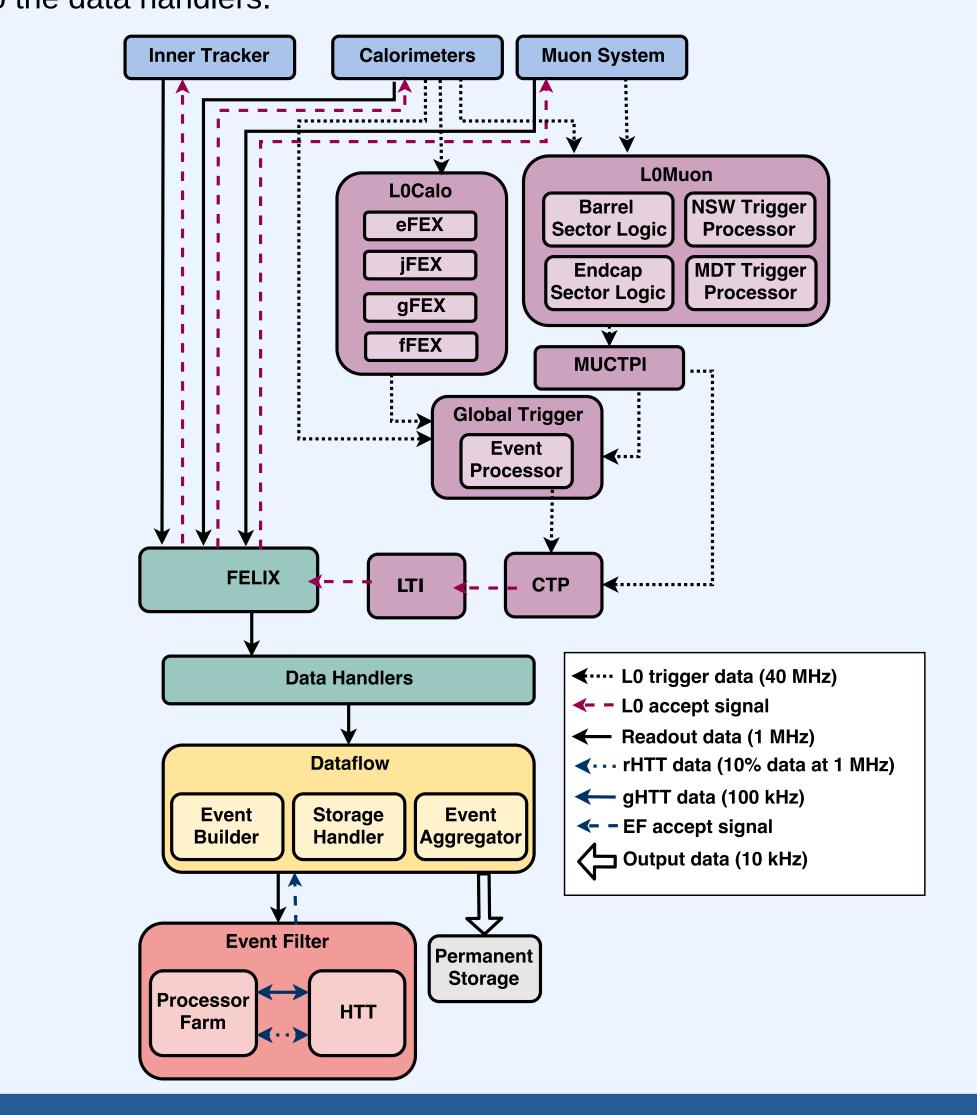


### Hardware

The main hardware platforms are currently the FLX-182 and the FLX-155. Both cards offer the same main functionality and are designed to fit in a server enclosure.

## The FELIX system

The complete FELIX system consists of a server, with custom drivers and software, and custom electronics, including software and firmware. The FELIX server is connected to the network through which the data is relayed to the data handlers.



# FLX-182

AMD Versal Prime VM1802 PCle Gen 4 x 16 (240Gb/s)

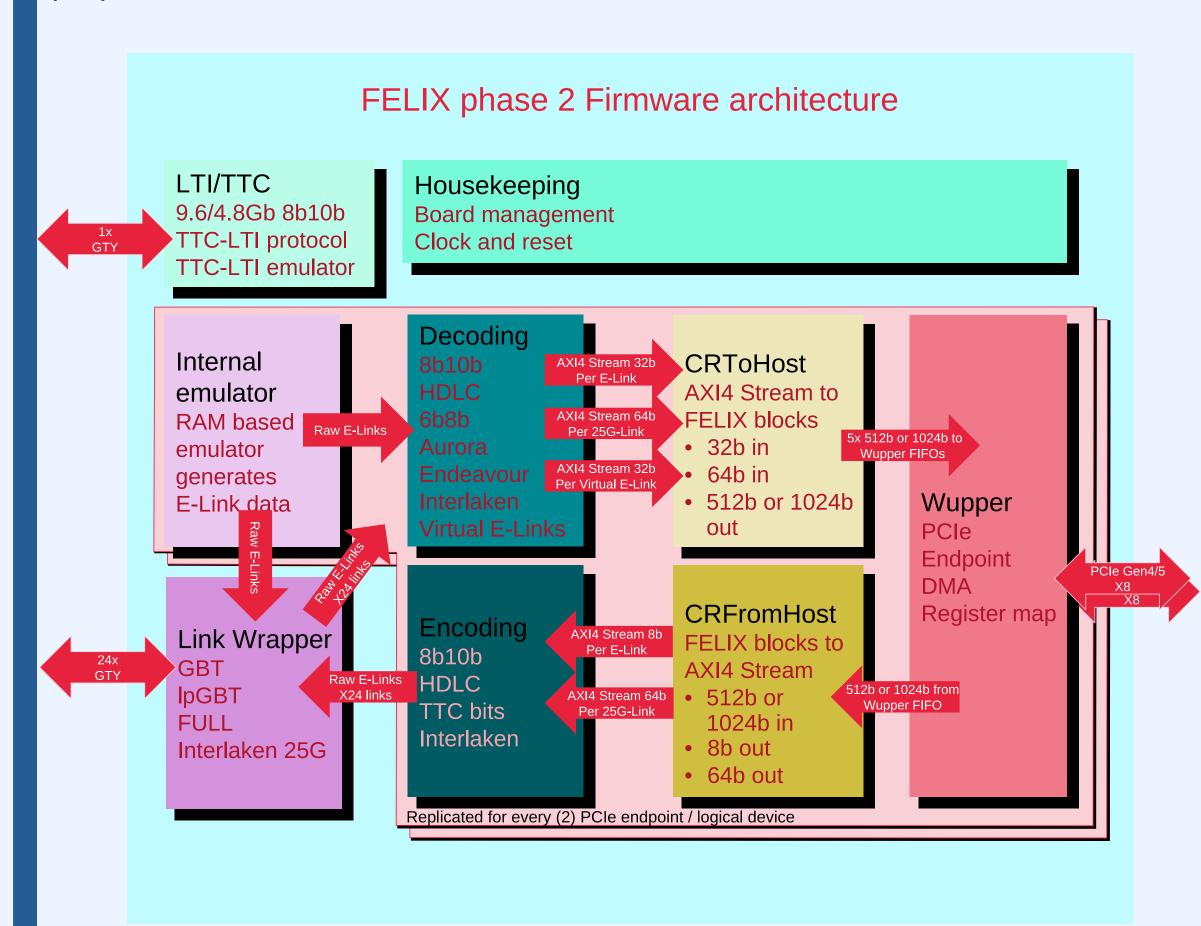
24 bidirectional optical links (25Gb/s)

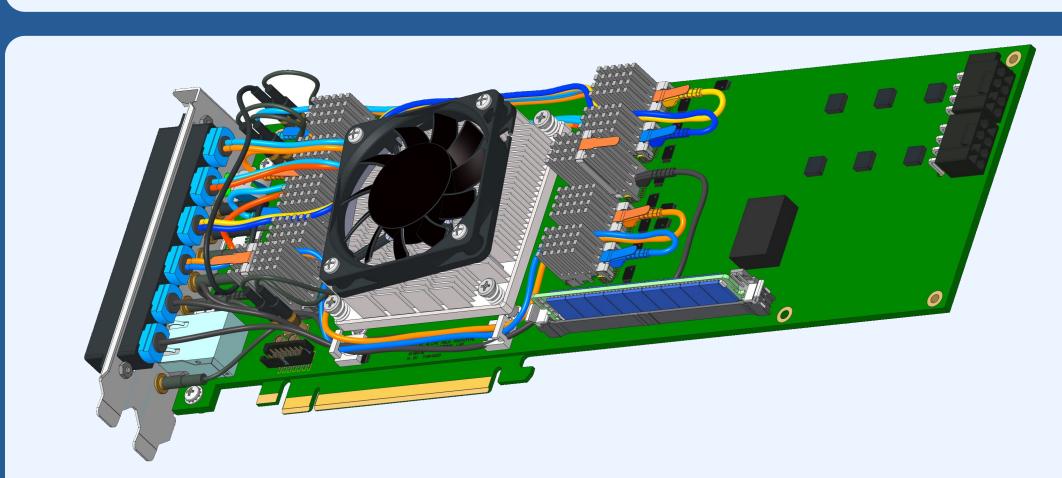
1 Firefly for Timing Trigger & Control (TTC)

### Firmware

The firmware is designed for continuous throughput of data, it is compatible with different link-wrapping protocols and supports various transmission protocols (8b10b, Aurora, etc.).

The firmware distributes the timing and triggers from the Local Trigger Interface (LTI) to the front end electronics.





## **FLX-155**

AMD Versal Premium VP1552

PCIe Gen 5 x 16 (480Gb/s)

48 Bidirectional optical links (25Gb/s)
1 Firefly for Timing Trigger & Control (TTC)

1 Firefly for 100Gb Ethernet

## Timing distribution

Timing distribution to the front end electronics is done with the Timing Compensated Link (TCLink). This is designed for long term phase compensation.

The FELIX cards are tasked with a deterministic clock recovery from the LTI.

### Knypaegje

Our tests showed, that the clock recovered by the GTYe5 transceivers on the Versal chip, exhibited a phase difference w.r.t. the reference clock, in the form of a doubly-peaking distribution. The question was, what extra information can we get out of the transceivers? The first suspect was the equalization settings, but changing these settings did not seem to improve performance. The next idea was that there might be information in the eye diagrams. At first, the expectation was that the eye might shift around the x-axis, but a correlation between eye opening and startup phase was found.

The goal was to develop something that could make an eye diagram without the use of external hardware/software (e.g. no Vivado hardware server). This resulted in a solution that is either firmware or software. The decision was made to implement an algorithm in software that communicates with the GTYe5 transceivers using an AXI4-lite bus.

Thus, Knypaegje (the Frisian word for wink) was born. Knypaegje is a program that runs on the ARM processor and measures a small number of eyes (currently 10). After these measurements, it checks to see if the current eye opening is "in the top of the cloud". If this is not the case, the RX of the receiver is reset to see if the current eye is this threshold.

A Knypaegje demonstrator developed using the FLX-182 as a hardware platform has proven the concept, as shown on the left. The final implementation is under development.

### TCLink:

https://gitlab.cern.ch/HPTD/tclink

Knypaegje:

https://gitlab.cern.ch/atlas-tdaq-felix/felix-versal-tools/knypaegje

GTY AXI driver kernel module:

https://gitlab.cern.ch/atlas-tdaq-felix/felix-versal-tools/gty-axi-driver Petalinux environment:

https://gitlab.cern.ch/atlas-tdaq-felix/felix-versal-tools/flx-petalinux

EEL IV ...alaaita.

FELIX website: https://atlas-project-felix.web.cern.ch/atlas-project-felix/

