

0.1 ITk Off-detector Readout/Control Electronics and Interfaces to TDAQ

0.1.1 Scope

The off-detector readout and control electronics are responsible for communication between the on-detector electronics and the rest of the world. This covers control, monitoring, calibration and read-out of physics data, and in some cases DCS data.

ITk readout and control is closely coupled to the TDAQ common readout and control architecture. TDAQ are providing a software framework and common hardware and interfaces that will be used where possible. This architecture is described in the TDAQ IDR, and includes the Front-End Link eXchange (FELIX), the Local Trigger Interface (LTI) and the Data Handler (DH).

All front-end (FE) data and controls flow through the FELIX. The LTI interfaces with the ATLAS Trigger, Timing and Control (TTC) system, and the Data Handler interfaces with the data storage system.

[If no better diagram, probably should make our own?] The diagram below (fig. 0.1.1) from the TDAQ Phase-II IDR [1] shows their view of the system. This diagram is missing the LTI which should be between LOTO and L1 CTP and the detector FELIXs, and the Controller.

The FE-control part of the system (covering clocks, triggers, resets and configuration) operates independently of the readout part in most cases, but there are places where they converge, for example, when performing a calibration loop.

Overseeing operation will be an ITk Controller PC, providing an interface and hub for ITk co-ordination of the system. This should be capable of independently controlling multiple divisions, but it is not yet known whether this could be a single shared system for all of ITk, or dedicated instances for Strips and Pixels. An operator console PC completes the system. It should be thought of as an expert terminal, located, for instance, in the satellite control room.

Readout is handled by a combination of a FELIX units and a Data Handler PCs, and FE-control will be via LTI units coupled to portions of the FELIX firmware. The Controller will be responsible for coordinating these devices, for instance when calibrating, or while starting a physics run. The bulk of the readout heavy-lifting will be performed by the FELIXs and Data

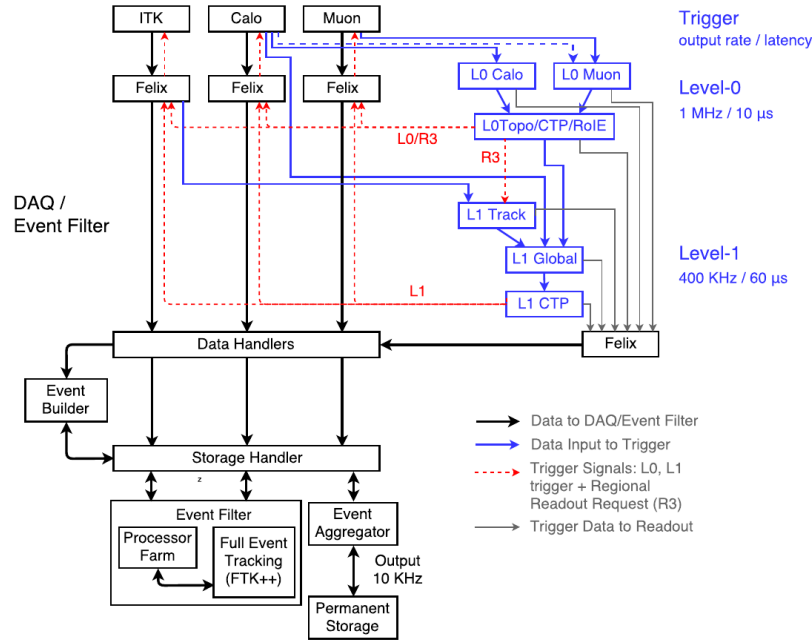


Figure 1: TDAQ global View of the two-level architecture. (ref TDAQ IDR fig 13) (Note: missing Controller and LTI)

Handlers, while interfaces to the ATLAS global triggers and control, and local signal generation and distribution will be provided by LTIs.

In each of the TDAQ common units - LTI, FELIX and Data Handler - ITk specific firmware/software will be needed. The Controller will run ITk software, but using TDAQ libraries to communicate with the lower-level systems. This mix of common and specific infrastructure predicates a number of interfaces that need to be carefully described.

The ITk-FELIX component will be a set of firmware for dealing with FE links, data formats and critical processing. Formatting of triggers etc. into ITk specific protocols will also take place here.

The ITk Data Handler will be a software package that resides within the TDAQ standard PC/software architecture.

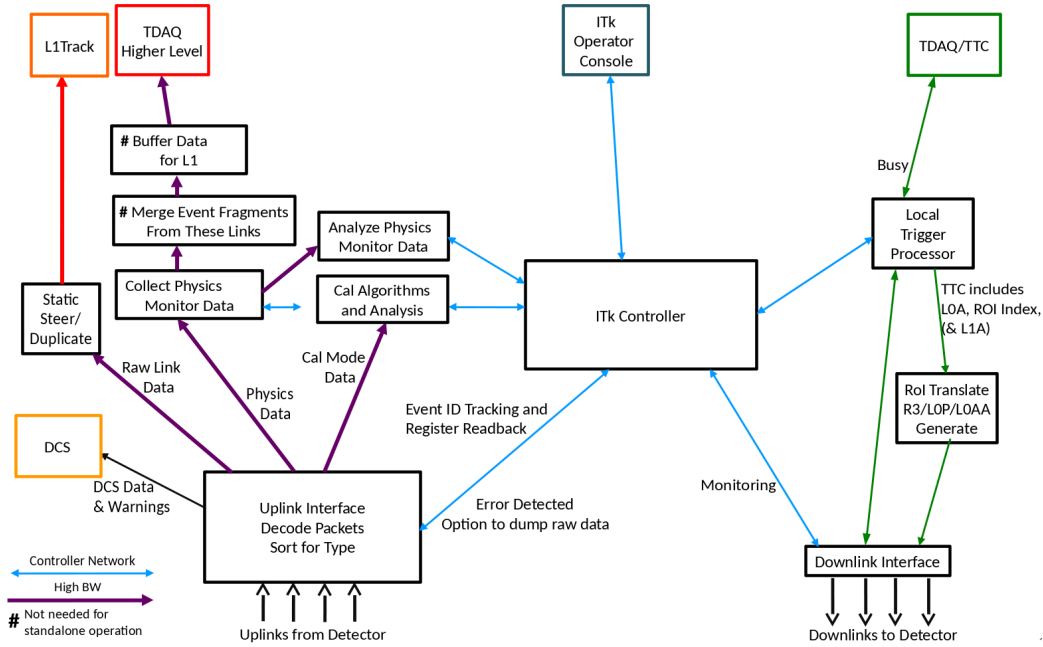


Figure 2: Readout Overview

0.1.2 FELIX

The ITk are one of the few sub-detectors that connect FELIX directly (via the versatile link) to their on-detector electronics. A high level description of the FELIX is a box that interfaces between a detector specific signalling (e.g. GBT) and Ethernet. This is a change from the current (i.e. SCT) system where all (electronic) low-level access to the detector modules is performed within one VME crate and mostly within one VME board. This alternative method is thought to be viable but needs to be explored in more detail before it is adopted. This is only possible with ITk specific firmware built into the FELIX to provide the immediate interface to the Versatile links since the protocol on these links will be unique to ITk.

The following describes general requirements from the ITk point of view.

In the counting room, the FE fibres terminate at a FELIX. Here an FPGA will provide resources for decoding the FE-protocol. Within FELIX, the FE data frames and packets are processed by ITk specific firmware, responsible for decoding the custom data encoding/format, performing all ITk functions

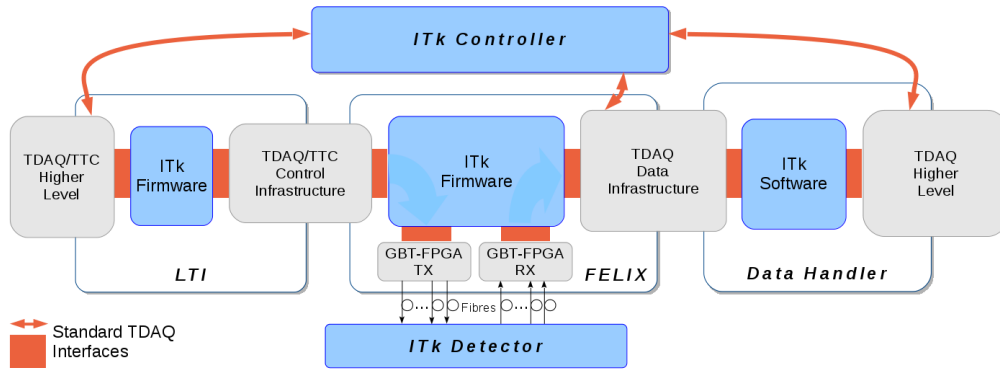


Figure 3: Diagram showing FELIX, Data Handler, LTI and Controller interfaces

and interfacing to standard FELIX data-formatters and transport (for transferring the data to the Data Handlers).

0.1.3 FELIX Functions

FE-frame interface, decoding: The FE (sub-)link configuration is customisable and detector specific, As data from each module are not aligned with respect to data from other modules, and the data is transferred serially, a dedicated 8b/10b decoder/deserialiser and packet detector is needed for each (sub-)link. It is expected that the demultiplexing and 8B/10B decoding could make use of common libraries provided by CERN or TDAQ.

Data-type handling: The link from each module carries multiple data types. Firmware will separate these data into different queues as needed. In general all data will go to the Data Handler, but with a few important exceptions:

- **Error flagging/handling and monitoring:** To ensure prompt recovery from errors, it may be advantageous to deal with errors within the FELIX firmware, or to send an alert to the Controller. Any errors detected in the FE-frame or clock and data recovery (where used) will be reported. If errors become frequent on a link, the raw data may be

dumped through ITk Control to the ITk Operator Console for diagnosis. Each packet will be examined for any recognizable corruption and if any is found, an error flag will be sent to ITk Control with the option of dumping the bad packet to the ITk Operator Console, discarding the packet or accepting the packet even with the corruption.

- **Calibration:** Data that is returned as part of a calibration loop may need to be dealt with on FELIX - for local histogramming and/or event counting. More details in the Calibration section (0.1.7)

Busy If any of the queues on FELIX exceed a fullness threshold, the busy needs to be asserted.

Event integrity checking (low level): Beyond checking for data corruption, event data needs to be checked for synchronisation and correlation with sent-trigger information. As this may require a prompt response, it should be performed here.

Control functions: The FELIX sends clocks (via the GBT directly) and control signals (via e-links) to the detector. Most of the control function is handled by the LTI, but some may be duplicated such that they can be applied to each downlink individually. E.g. local triggers, resets. In particular the data-link level encoding of TTC signals into the detector FE-protocol will likely be done in the FELIX. Some signals, e.g. configuration streams, will need to be delay paths to facilitated the synchronous signalling, e.g. triggers.

0.1.4 Data Handler

The Data handler implements the final detector-specific data processing layer before event data is sent to the storage handlers, e.g. formatting and/or monitoring [1]. The Data Handler is a TDAQ PC with a common software framework in which ITk software components are run. The infrastructure can be shared between physics and calibration running.

FE data will be parsed and decoded into hits or register data. Data is likely to be gathered by event ID, building event-fragments. These will be passed onto TDAQ components, via common software components.

Stand-alone local running: For calibration mode, histograms can be populated, and fits performed where needed (similarly for monitoring in physics mode). Results can be passed to ITk Control for other cross-checks and analysis and finally stored (this should be TDAQ provided storage).

Fragment merging: Merging event-fragments into macro-fragments [mentioned above, but needs more detail here]

Register readback: Forward any register readback to ITk Control.

Event integrity checking (high level): Fragments checked for correct L0ID and BCID etc. This functionality will be shared with that of the FELIX firmware, although at this level there is far more detail of inter-trigger counters etc. to make use of.

0.1.5 LTI

The ATLAS-wide Trigger, Timing and Control (TTC) system connects to the ITk via Local Trigger Interfaces (LTIs). The LTI is also connected directly to the Level-0 Trigger System. The LTI forms part of the hierarchy of control signal distribution and at the lowest level will connect to multiple FELIXs. The ratio of LTI to FELIX is not yet known, but is expected to be of order 16 FELIX per LTI. This leads to about 8[?] LTIs feeding into a top-level LTI for ITk Strips, and more for Pixels.

The unit is responsible for generating local (e.g. for a subdivision of the ITk) triggers, other broadcast commands (e.g. BCR, ECR) and calibration strings [maybe in FELIX instead?].

Much of the LTI control function could be part of the FELIX, particularly more localised (e.g. link specific) signalling. This sharing will need to be optimised when details are clear - but as these are essentially control functions, they're detailed here.

The LTI is expected to provide many generic functions: Interfacing to the higher level systems, busy aggregation, monitoring and histogramming, trigger generation (and gating, when in stand-alone) etc.

Local Triggers: Generation of triggers for debug, calibration etc. of a partitions or smaller parts of the detector, e.g. a stave.

Command broadcasts (including local resets): Provides Command stream generation and sending. Interfaces to the Controller.

Register refresh: Managing the drip-feeding of automatic FE register refreshes. May need to maintain a mirror of FE registers in the LTI. This needs to respect a number of synchronising elements, e.g. Orbit or ECR.

Busy: Gating of local trigger when busy etc.

0.1.6 ITk Controller:

The Controller will manage the manage the data-flow, and form a hub for status and error flagging. The unit will likely be a PC, connected to all other systems via a dedicated control network. In most cases only commands and status updates will travel on the network, but it may be required to handle debug data dumps if other means are not present.

During data-taking, the Controller will setup the core systems to receive physics triggers (including configuring the modules), build monitoring histograms and then keep track of status and handling any errors when running.

During non-physics running, module calibration will be co-ordinated from the Controller.

0.1.7 Calibration

Calibration depends on closely coupled loops - the operation of these needs to be optimally shared between the FE ASIC, the FELIX firmware and the Data Handler software. Much of this is dependent on the deadtime (unused time between loops) of the system.

The calibration procedure will involve sending the entire chip configuration at the start and individual portions of the configuration as part of the loop structure configuring individual registers for each burst of triggers. A scan is made up of bursts of triggers, with a different configuration applied for each burst. A series of these scans is analysed and used to build a picture of the module status and performance.

The immediate result of the trigger loop will be a representation of the data acquired during this burst. This will usually include the occupancy

histograms, but other data will be useful, including histograms of chip occupancy, cluster size and count. For low-level investigations, a direct map of the bits on the link might be collected.

Histograms can either be built, in the FELIX or in the Data Handler. For Strips, there is an option to histogram hits in the FE ASIC in a very low-level form (hit counter per strip). When histogramming in the Data Handler, where the data might be available only asynchronously, the loop indices to which the data belongs needs to be communicated to the process building the histogram.

Going in to more detail about how this occurs: a calibration scan is carried out using a hierarchy of loops over bursts of triggers. The different levels include the main scan parameter(s) as well as mask changes (for charge injection). For the inner loop, a trigger will be an arbitrary short command sequence sent to the FE chips in order to generate a response. This includes injecting a calibration charge, sending L1, R3, L0 triggers, as well as register readback commands. In some cases this sequence will be specified in terms of raw signals to be sent (i.e. sent without any encoding). This command will be repeated many (e.g. 100) times and in order to make the complete scan as fast as possible, an automatic adjustment of the repetition frequency depending on the amount of data being received is desired. Otherwise, delays created in this loop will be multiplied by the repetition from higher parameter loops levels.

0.1.8 Monitoring

During data taking (but not necessarily enabled all the time) the physics data will need to be monitored. This function will also be an important part of stand alone operation during integration.

The primary purpose is to look for faults in modules that may not be obvious from examining data that passes the higher-level trigger and to respond to any warnings issued by the on-detector electronics. This function must have no effect on the physics data flow.

Monitoring data will be passed from the FELIX and Data Handlers to the Controller for assessment. Potentially all data sent from the detector should be available to be monitored.

- **Collect and analyse Physics Monitor Data** Monitoring of link occupancies, how much more data can be sent down the stream. In

addition to the raw event size, and the space between packets, it should be possible to keep track of how much time has passed between the transmission of the trigger and reception of the last data.

- **Triggers that are not useful for physics** Include looking for stuck cells, or high occupancy due to threshold misconfiguration (SEU).
- **Primitive operations** This may be a stream, or a single block of data triggered by user. Primitives set the output configuration, some of the options being:
 - Raw data from GBT/elink
 - Chip data (decompressed and packetized)
 - All event data
 - nth event
 - Large events (above threshold bytes)
 - All error data (as decided by error checker), including some number of events before and after
 - Matches pattern (e.g. hit in detector element)

0.1.9 L1Track Options

FELIX L1Track pre-processor: L1Track will have multiple track-finder units arranged in approximately 16 ATCA shelves. Data from the ITk (mostly Strips, but also outer-layer Pixels) will need to be routed to these shelves, and in some cases duplicated to a second shelf. There is a static correspondence between a data uplink and a shelf (or shelves). To reduce the number of links, bonding is expected to take place on the FELIX, where, for example, 10 10Gb GBT uplinks are combined into a single 100Gb fibre.

The FELIX may also be able to duplicate links where needed.

References

- [1] Simon George et al. Initial Design Review for the Phase-II Upgrade of the ATLAS TDAQ System. Technical Report ATL-COM-DAQ-2015-172, CERN, Geneva, Oct 2015. Draft for comments by the TDAQ community, <https://cds.cern.ch/record/2058270>.