Prepared for submission to JINST

25TH International Workshop on Radiation Imaging Detectors
June 30, 2024 to July 4, 2024
Lisbon, Portugal

ATLAS ITk-Pixel DAQ System

Wael Alkakhi[a] on behalf of ATLAS ITk collaboration

[a] *II. Physikalisches Institut, Georg-August-Universität Göttingen, Friedrich-Hund-Platz 1, 37077 Göttingen, Germany*

E-mail: wael.alkakhi@uni-goettingen.de

Abstract: During the ATLAS High-Luminosity Large Hadron Collider (HL-LHC) upgrade, the current inner detector is going to be replaced by an all-silicon Inner Tracker (ITk). The pixel detector, located in the innermost part of the ITk, comprises 9716 modules arranged in 5 cylindrical layers around the beam line. The ITk-Pixel Data AcQuisition (DAQ) system basic read-out chain includes the YARR software, communicating with the FELIX PCIe board acting as an interface connected through lpGBT transceivers to the on-detector front-end (FE) chips ITkPix. The FEs are grouped in triplet (3-FE) and mostly quad modules (4-FE, QM) that are installed on local supports, which are integral parts of the ITk structure. The FELIX system is also used for performing Quality Control (QC) tests during integration. The work describes the development steps and corresponding testing and read-out chain validation results. A representative read-out sub-system will be used to develop and validate the different aspects of the read-out chain. This subsystem can be a Loaded Local Support (LLS) comprising few tens of ITkPix QMs with serial powering (SP) and opto-box connection. In order to have the readout chain validated, developments on trigger and command sending and data reading of YARR FelixClient controller were consequently required, working first on a lab setup with a couple of ITkPix single chip cards (SCCs) and QMs. This step has been carried out successfully, paving the road to the next LLS sub-system readout test.

Keywords: Inner Tracker, ITk Pixel DAQ, YARR, ATLAS

ATL-ITK-PROC-2024-025
01 August 2024

Contents

1 Introduction

After Run 3, the LHC will undergo a significant upgrade, entering the HL-LHC era for Run 4. During this period, the luminosity will increase by about 5-7 times, reaching $\approx 7 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. At a bunch-crossing rate of 40 MHz, the ATLAS trigger-DAQ system during HL-LHC will support a level-0 trigger accept rate of 1 MHz, while around 200 pile-up events are expected.

The current ATLAS inner detector cannot meet the HL-LHC requirements, motivating an upgrade of the ATLAS inner detector. It will be replaced by the all-silicon ITk. The ITk consists of a 5-layer pixel detector surrounded by a 4-layer strip detector, as shown in Figure 1.

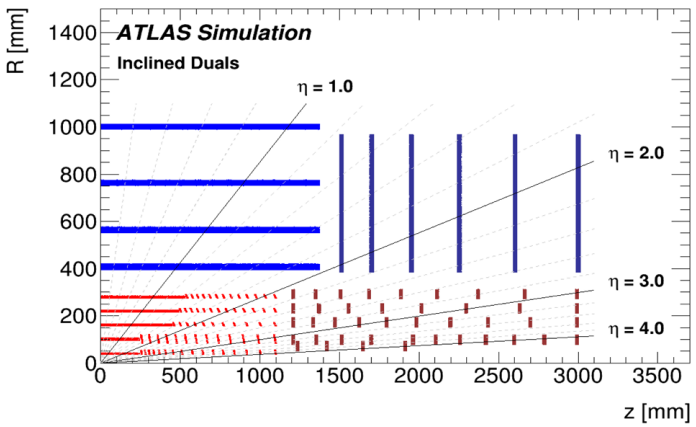

Figure 1. Schematics of a quarter of the Inner Tracker, showing its two sub-detectors: the ITk Pixel detector (red) and the ITk Strips detector (blue). The x-axis represents the beam line, and the point (0,0) represents the interaction point (midpoint) of the ATLAS experiment [1].

The ITk Pixel detector [2] consists of 5 cylindrical layers, arranged in an Inner System(IS), an Outer Barrel(OB), and in addition an Outer Endcap(EC). The IS is formed by the innermost 2 layers of the ITk. The OB and EC are formed by the last 3 layers of the ITk. The OB consists of flat staves and inclined rings, while the EC has vertical rings of modules.

The ITk Pixel detector covers an area of 13 m^2 and spans a high pseudorapidity of $|\eta| < 4$. The pixels used have a high resolution, with a pixel size of 25 µm × 100 µm in some part of the innermost layer and a pixel size of 50 µm × 50 µm for the rest of the ITk pixels.

OB hosts around 4772 Quad Modules (QMs), each with 4 front-end (FE) chips. These QMs are mounted to Loaded Local Supports (LLS), of which there are 2 types, as shown in Figure 2:

- A Longeron hosts 36 of the flat QMs.
- An Inclined Half Ring hosts between 16 and 28 of the inclined QMs.

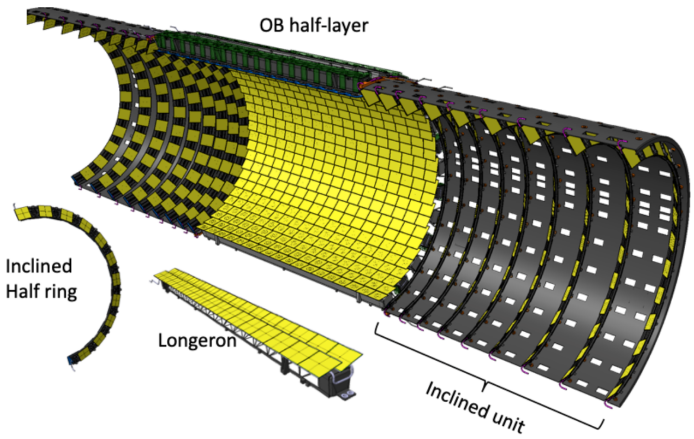

Figure 2. Schematics of OB loaded local supports [3].

The structure of the OB system of the ITk presents significant challenges in reading out a large number of QMs. The main target is to have a valid data acquisition readout system capable of reading out all QMs, starting with one FE and then scaling up to read out all QMs.

In the following, the readout Data Acquisition system (DAQ) of the ITk is described in Section 2. The results of the ITk DAQ software are presented in Section 3, and finally, a conclusion about the project and its outlook is presented in Section 4.

2 ITk Read-out DAQ

During the operation of the ITk, about 10,000 modules of FE chips have to run simultaneously. Commands have to be sent to about 5 billion pixels, and the ITk DAQ system has to be able to cope with the correspondingly huge amount of data received from these modules.

The ITk DAQ system is shown in Figure 3. It consists of the modules of ITkPix FE chips (triplets or QMs). Each QM is connected to one optobox [4], which is a unit that hosts up to 8 optoboards. Each optoboard has 3 ASICs: GBCR, lpGBT, and VTRx+. The GBCR enhances the electrical signal integrity for commands before they are sent to QM and also the electrical signal integrity of the incoming data from the QMs. The lpGBT distributes the commands to the QMs and aggregates the data received from the QMs. The VTRx+ converts commands from optical to electrical and converts the received readout data from electrical to optical signal, the optobox is connected to the ITk DAQ PC via optical cables, with command transmission at 2.56 Gb/s and data

reception at 10.24 Gb/s. The DAQ PC consists of ITk DAQ hardware represented by a FELIX Card, a PCIe FPGA board which interfaces between the optoboard and the ITk DAQ software. The DAQ PC also includes ITk DAQ Software, which is YARR (Yet another Rapid Readout) [5] [6]. YARR creates the commands and sends them to QMs through FELIX and also receives the event data in order to process and stores it in the database for later analysis.

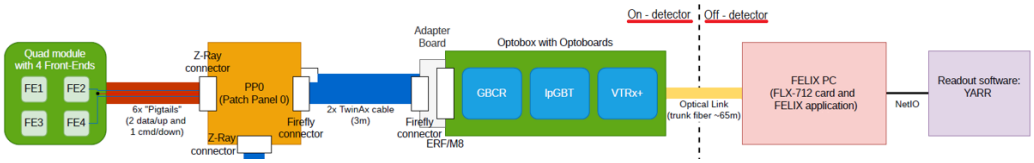

Figure 3. Schematics of ITk Data Acquisition system [7].

The FELIX is operated via the Felix-Star application, which is the central process of the FELIX system. It is a multithreaded operation where many FELIX cards can be operated simultaneously. Felix-Star consists of 3 sub-applications: felix-tohost (from FELIX Card to DAQ software), which is responsible for dealing with the readout data coming from the QM; felix-tofx (from DAQ software to FELIX Card), which is responsible for timing, trigger, and control commands sent to the QM; and felix-register, which provides the ability to access the FELIX register for writing and reading purposes.

In addition, a protocol named Netio-Next is used, which is a fast communication protocol for data messaging between Felix-Star and FELIX clients. The FELIX software provides the user with an application named Felix-Client Interface that interacts with Netio-Next and hides the complexity of Netio-Next.

YARR is a C++-based software which consists of two types of libraries: the Chip libraries, which create commands and triggers and analyze data, and the Controller libraries, which provide communication between hardware and chip libraries. In order to interact with the Felix-Client Interface, a controller library named LibFelixClient is developed. It has a TXCore for command transmission preparation and an RXCore for data preparation for further processing in YARR Chip libraries.

3 Experimental Results

One of the requirements for the loaded local production phase is that LLS undergoes quality control (QC) tests after mounting all the QMs to ensure that the QMs are in good condition after mounting the QMs, some of these QC tests are electrical, such as digital scans to check the digital part of the pixel and analog scans to check the analog part of the pixel. This requires to have an ITk DAQ system that is able to read out all the QMs in the LLS, as mentioned in Section 1. The starting point is to validate the ITk DAQ using one QM and then scale up with the number of QMs under test.

The lab setup shown in Figure 4, is located in the Forschungs- und Technologiezentrum Detektorphysik (FTD) at the University of Bonn. The setup consists of:

- 3 ITkPixV1 QMs.
- Electrical connections.

- Optobox with one optoboard.
- Optical connections.
- FELIX server hosting a FELIX card with installed optoboard software, FELIX software, and YARR software.

Figure 4. Lab setup at FTD.

The validation procedure starts with the validation of one FE by performing a YARR digital scan[1] and then scales up by adding more FEs. The validation of the ITk DAQ read-out for 10 ITkPix FEs with no data transmission errors in 3 ITkPix QMs is shown in Figure 5.

During the scaling-up procedure, the time consumed by the YARR processes during the scan is recorded. The YARR processes include Configuration, Scan, Processing, and Analysis. Configuration is the first stage where YARR sends commands to configure the chips. Scan is when YARR sends the scan command. After the scan, YARR checks the incoming data in the Processing stage and then analyzes it in the Analysis process.

The time consumed by each YARR operation and the total time consumed by all the processes are presented in Figure 6. As the number of scanned chips increases, the time consumed by YARR configuration increases since YARR configures the read-out chips one by one. The scan time consumed by YARR increases slightly as the number of chips increases, which is a positive indication for the scalability of the YARR software since related scan commands are sent to all FEs simultaneously, not FE by FE. The next step will be to extend these tests to a full featured LLS readout, once more QMs are available.

4 Conclusion

The preparation of the ITk DAQ readout system, for the next ATLAS upgrade is very challenging in terms of higher performance and increasing size. The ITk-Pixel readout part should be able to

[1]The digital scan is performed by injecting a digital signal into every enabled pixel to test the chip's digital component.

```
[12:59:48:854][  info  ][  ScanConsole  ][46207]: Processor done, waiting for histogrammer ...
[12:59:49:084][  info  ][HistogramAlgorithm][46297]: Histogrammer done!
[12:59:49:092][  info  ][  StdAnalysis  ][46296]: [6][0x154c8] Total number of failing pixels: 0
[12:59:49:093][  info  ][HistogramAlgorithm][46294]: Histogrammer done!
[12:59:49:102][  info  ][  StdAnalysis  ][46293]: [5][0x15478] Total number of failing pixels: 0
[12:59:49:105][  info  ][  StdAnalysis  ][46296]: [6][0x154c8][0] ToT Mean = 7 +- 0
[12:59:49:114][  info  ][  StdAnalysis  ][46293]: [5][0x15478][0] ToT Mean = 7 +- 0
[12:59:49:177][  info  ][HistogramAlgorithm][46306]: Histogrammer done!
[12:59:49:182][  info  ][HistogramAlgorithm][46279]: Histogrammer done!
[12:59:49:183][  info  ][  StdAnalysis  ][46305]: [9][0x14724] Total number of failing pixels: 0
[12:59:49:187][  info  ][  StdAnalysis  ][46278]: [0][0x1467c] Total number of failing pixels: 0
[12:59:49:195][  info  ][HistogramAlgorithm][46291]: Histogrammer done!
[12:59:49:195][  info  ][  StdAnalysis  ][46305]: [9][0x14724][0] ToT Mean = 7 +- 0
[12:59:49:197][  info  ][HistogramAlgorithm][46282]: Histogrammer done!
[12:59:49:198][  info  ][  StdAnalysis  ][46290]: [4][0x154b8] Total number of failing pixels: 0
[12:59:49:200][  info  ][  StdAnalysis  ][46278]: [0][0x1467c][0] ToT Mean = 7 +- 0
[12:59:49:201][  info  ][  StdAnalysis  ][46281]: [1][0x14647] Total number of failing pixels: 0
[12:59:49:203][  info  ][HistogramAlgorithm][46303]: Histogrammer done!
[12:59:49:203][  info  ][HistogramAlgorithm][46285]: Histogrammer done!
[12:59:49:205][  info  ][HistogramAlgorithm][46300]: Histogrammer done!
[12:59:49:208][  info  ][  StdAnalysis  ][46302]: [8][0x14735] Total number of failing pixels: 2
[12:59:49:208][  info  ][  StdAnalysis  ][46284]: [2][0x14627] Total number of failing pixels: 0
[12:59:49:209][  info  ][  StdAnalysis  ][46299]: [7][0x154d8] Total number of failing pixels: 0
[12:59:49:211][  info  ][  StdAnalysis  ][46290]: [4][0x154b8][0] ToT Mean = 7 +- 0
[12:59:49:214][  info  ][  StdAnalysis  ][46281]: [1][0x14647][0] ToT Mean = 7 +- 0
[12:59:49:220][  info  ][  StdAnalysis  ][46302]: [8][0x14735][0] ToT Mean = 7.000078125 +- 0.021949139614520392
[12:59:49:221][  info  ][  StdAnalysis  ][46284]: [2][0x14627][0] ToT Mean = 7 +- 0
[12:59:49:222][  info  ][  StdAnalysis  ][46299]: [7][0x154d8][0] ToT Mean = 7 +- 0
[12:59:49:226][  info  ][HistogramAlgorithm][46288]: Histogrammer done!
[12:59:49:226][  info  ][  ScanConsole  ][46207]: Processor done, waiting for analysis ...
[12:59:49:230][  info  ][  StdAnalysis  ][46287]: [3][0x14636] Total number of failing pixels: 0
[12:59:49:242][  info  ][  StdAnalysis  ][46287]: [3][0x14636][0] ToT Mean = 7 +- 0
[12:59:49:445][  info  ][AnalysisAlgorithm][46287]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46299]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46302]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46293]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46284]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46278]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46281]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46290]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46296]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46305]: Analysis done!
[12:59:49:627][  info  ][  ScanConsole  ][46207]: All done!
[12:59:49:627][  info  ][  ScanConsole  ][46207]: ##############
[12:59:49:627][  info  ][  ScanConsole  ][46207]: ##  Timing  ##
[12:59:49:627][  info  ][  ScanConsole  ][46207]: ##############
[12:59:49:627][  info  ][  ScanConsole  ][46207]: -> Configuration: 18417 ms
[12:59:49:627][  info  ][  ScanConsole  ][46207]: -> Scan:          12727 ms
[12:59:49:627][  info  ][  ScanConsole  ][46207]: -> Processing:    1 ms
[12:59:49:627][  info  ][  ScanConsole  ][46207]: -> Analysis:      772 ms
```

Figure 5. The output of the YARR digital scan shows a successful scan of 10 used FEs simultaneously, with no data processing errors.

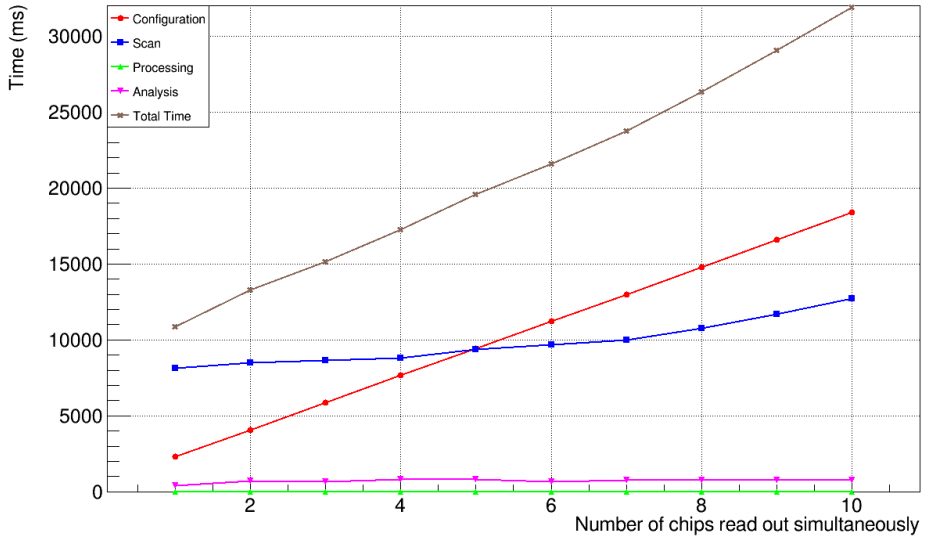

Figure 6. Time consumed by YARR processes as function of the number of FEs under scan.

handle simultaneously about 10k triplet and quad modules. The work described here shows the first steps of evaluating the ITk-Pixel DAQ system, starting by validating the readout chain on a lab setup, then extending the setup targeting a full-featured LLS, with tens of QMs connected as in the final ITk layout. Scans operating simultaneously up to 10 FEs, using 3 digital ITkPixV1 QMs were successfully carried out. The overall execution timing analysis was done, showing positive indications for scalability of the ITk-Pixel DAQ system. Further development efforts will continue once more QMs, currently being assembled, are available for the LLS sub-system testing.

References

[1] ATLAS collaboration, *ATLAS Upgrades*, Tech. Rep. ATL-UPGRADE-PROC-2020-001, CERN, Geneva (2021), DOI.

[2] ATLAS collaboration, *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*, Tech. Rep. CERN-LHCC-2017-021, ATLAS-TDR-030, CERN, Geneva (2017), DOI.

[3] ATLAS collaboration, *Carbon based local supports for the ATLAS ITk-pixel detector*, Tech. Rep. ATL-ITK-PROC-2023-003, CERN, Geneva (2023), DOI.

[4] S. Möbius, *The optosystem: validation and testing of the high-speed electro-optical conversion system for the readout of the atlas itk pixel upgrade*, *JINST* **19** (2024) C04015.

[5] T. Heim, *Yarr - a pcie based readout concept for current and future atlas pixel modules*, *Journal of Physics: Conference Series* **898** (2017) 032053.

[6] N.L. Whallon, T. Heim, M. Garcia-Sciveres, A. Sautaux, H. Oide, K. Potamianos et al., *Upgrade of the YARR DAQ system for the ATLAS Phase-II pixel detector readout chip*, *PoS* **TWEPP-17** (2018) 076.

[7] ATLAS TDAQ collaboration, *FELIX: First operational experience with the new ATLAS readout system and perspectives for HL-LHC*, *EPJ Web Conf.* **295** (2024) 02012.