ITk Pixel Optosystem

6th September 2023

Daniele Dal Santo

UNIVERSITÄT BERN

AEC
ALBERT EINSTEIN CENTER
FOR FUNDAMENTAL PHYSICS

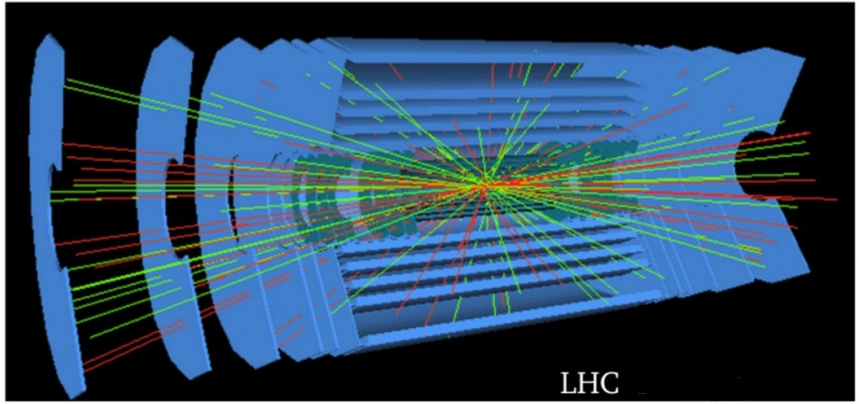
ATLAS EXPERIMENT

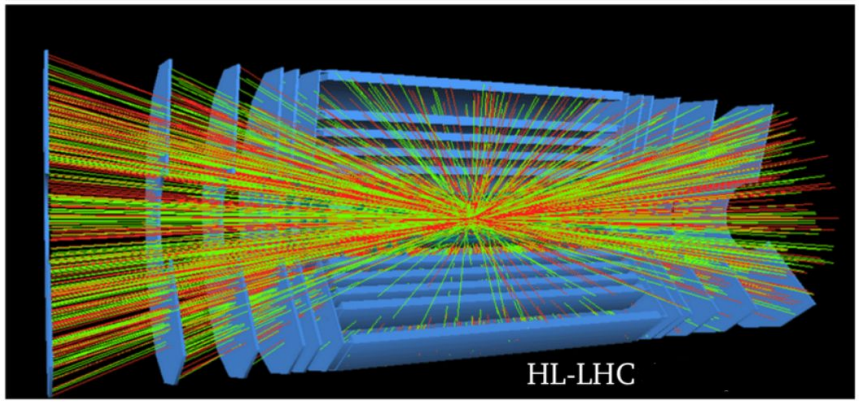
Motivations for ITk

The High-Lumi LHC will deliver *200 collisions per bunch crossing* @ 40MHz. Upgrading the *ATLAS Inner Detector* is necessary to maintain an effective track reconstruction.

Characteristics required for Inner Tracker (ITk):

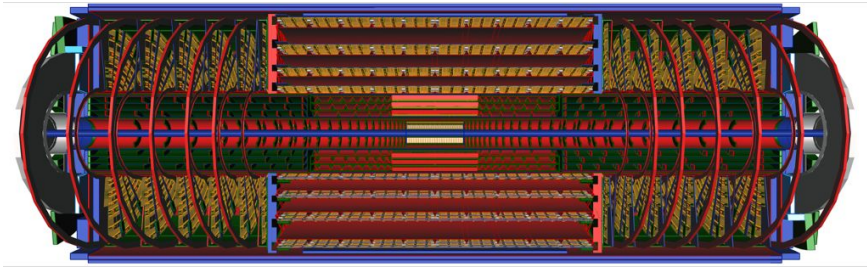
- highly segmented all-silicon detector (from 92M pixel channels and 6M strip channels to 5G and 50M)
- increased coverage (up to $|\eta|=4$)
- increased radiation-hardness
- faster read-out electronics

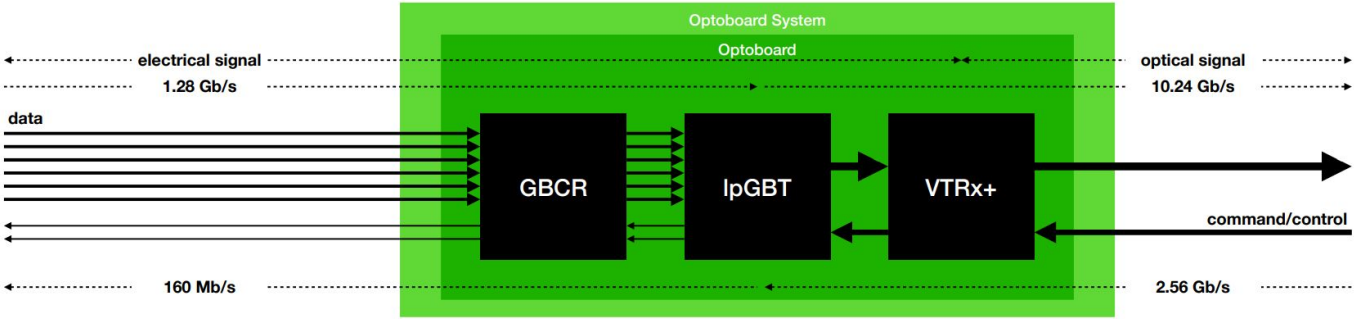




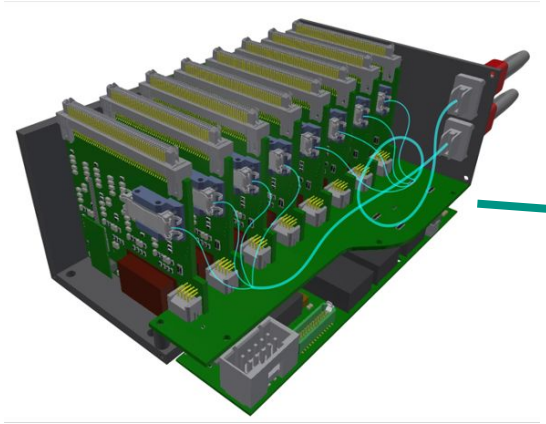
ITk Pixel Data Transmission Chain

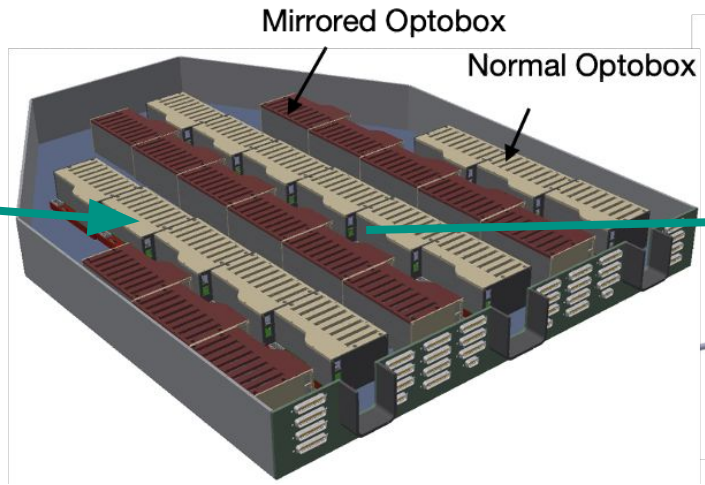
- The ITk Pixel must be read out at 1 MHz (50 Tbps)
- One of the pivotal components of the ITk is the Optosystem, whose fundamental unit is the Optoboard
- This handles the conversion of electrical signals to optical (and viceversa) and their aggregation

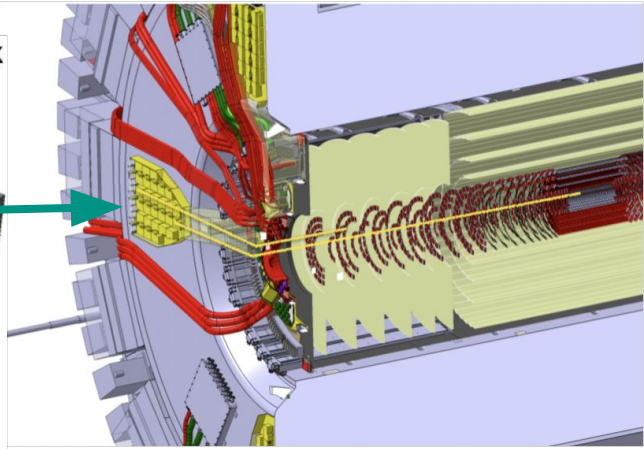




ITk Pixel Optosystem Overview





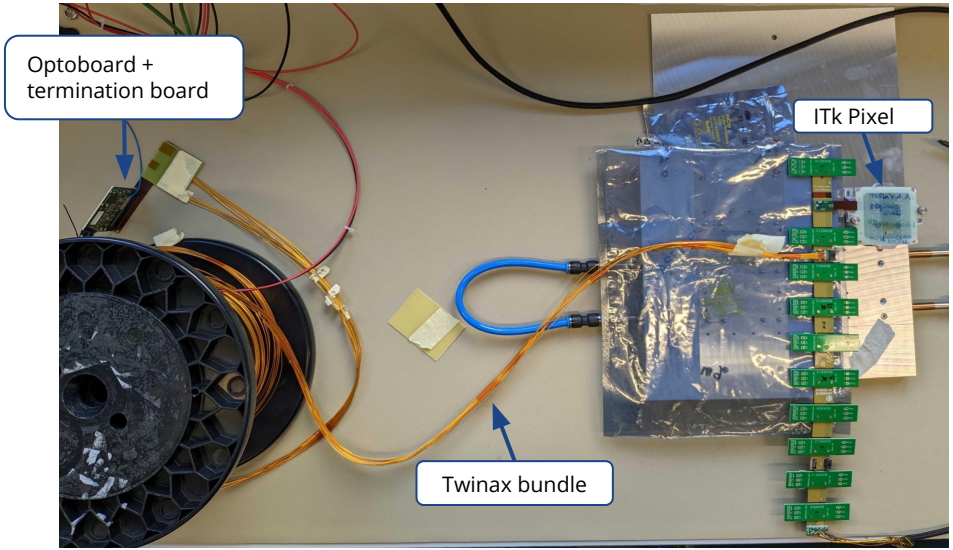


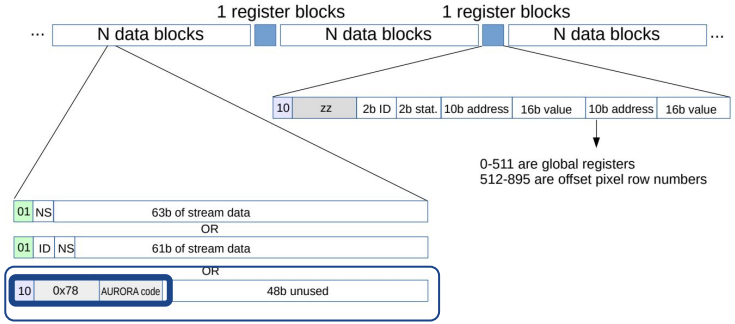
- Each of the Optoboxes contains up to 8 Optoboards.
- 28 Optoboxes are housed in Optopanels.
- These create twinax and fibres+power+monitor cable channels.
- There are four Optopanels on the endplate at each side of ITk, inside ATLAS.

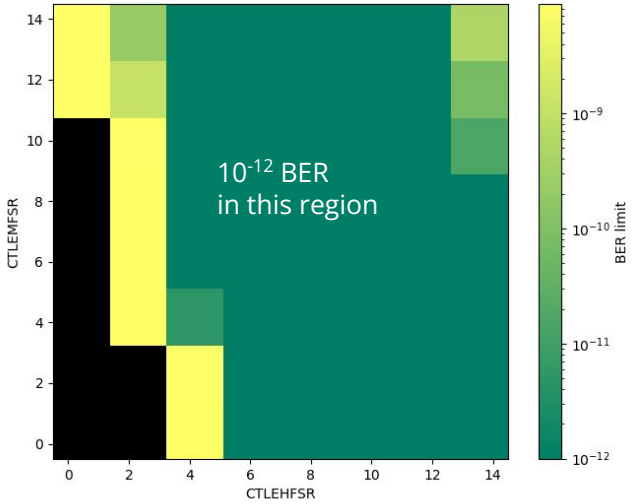
Data Transmission Quality

Test of the final data transmission chain:

- Optoboard optimization to ensure high signal quality from ITk Pixel via patch panel and 6m twinax bundle

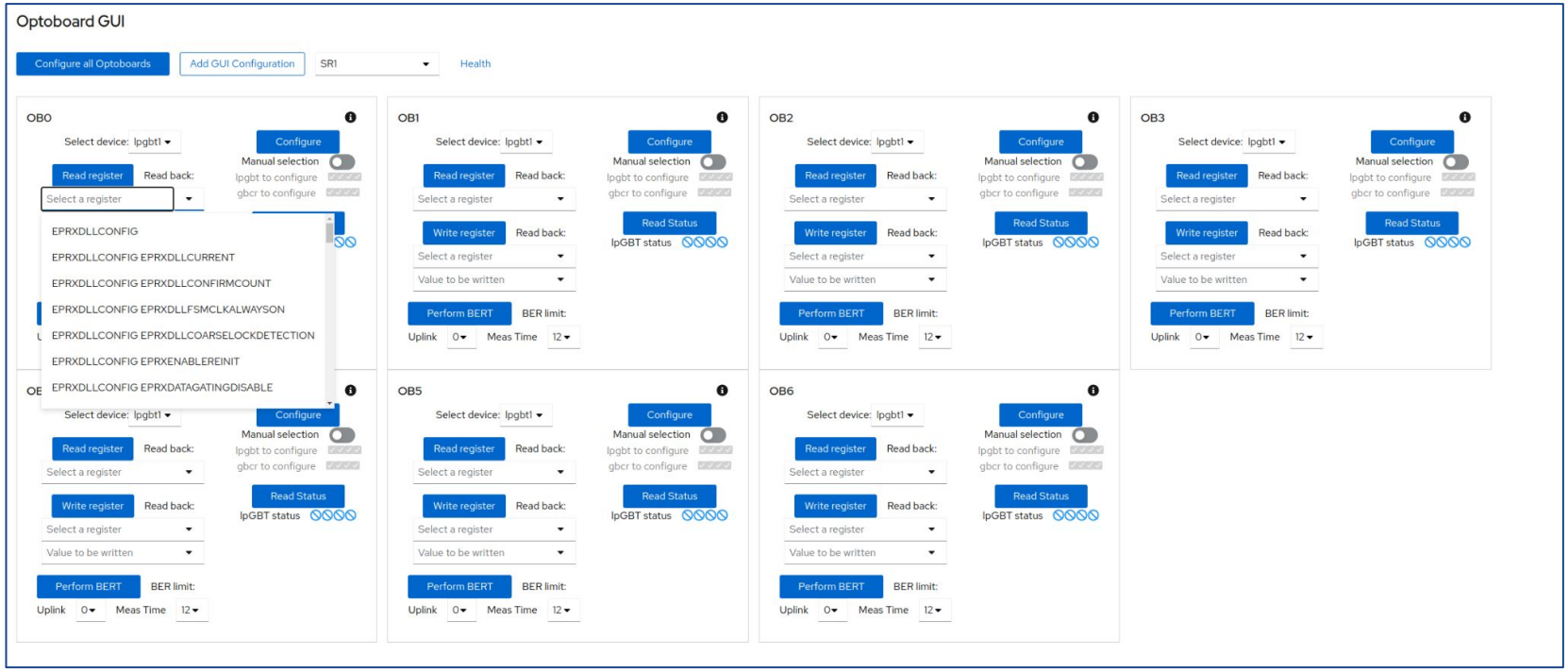






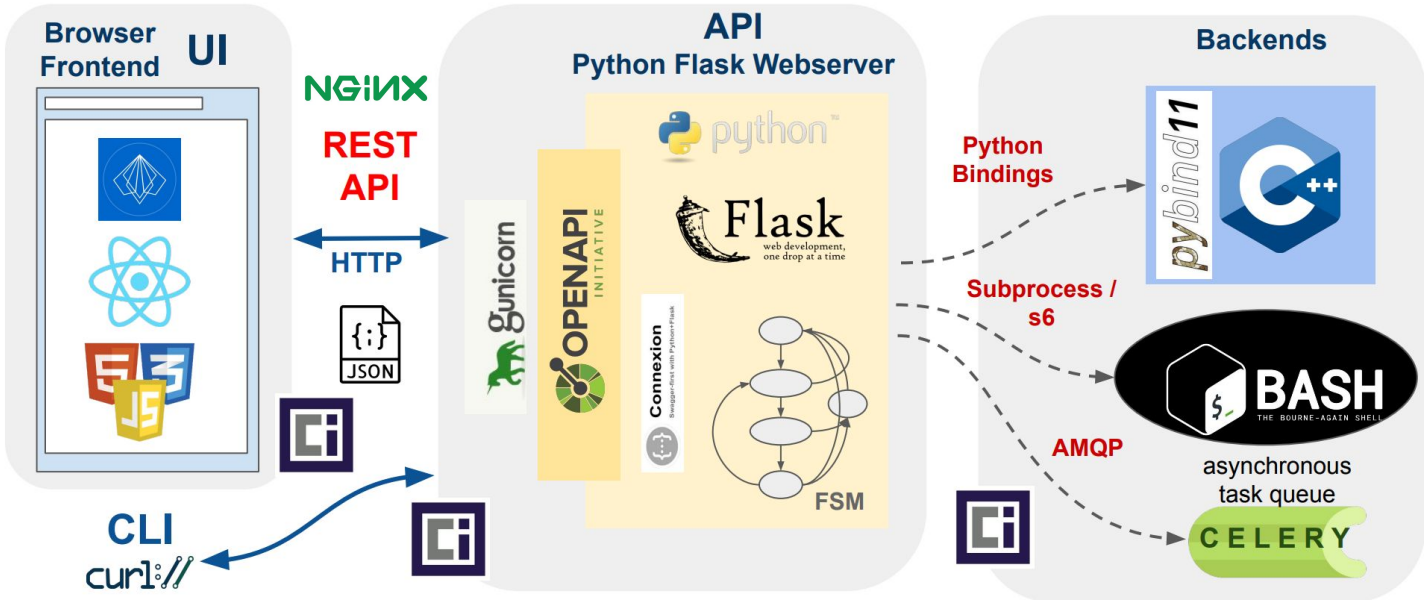
Optoboard GUI

- development of a user-friendly GUI to interact with the Optoboards

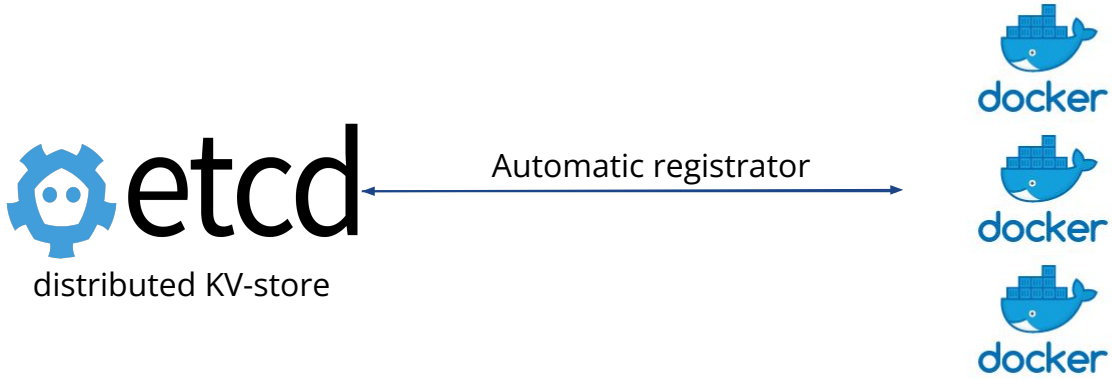


Optoboard GUI: under the hood - Part 1

- UI, API and backend in separate docker containers published in gitlab-registry
  - NGINX for static file serving and reverse proxy
  - openAPI standard for Flask endpoint definition
  - Celery + RabbitMQ to connect web app to backend



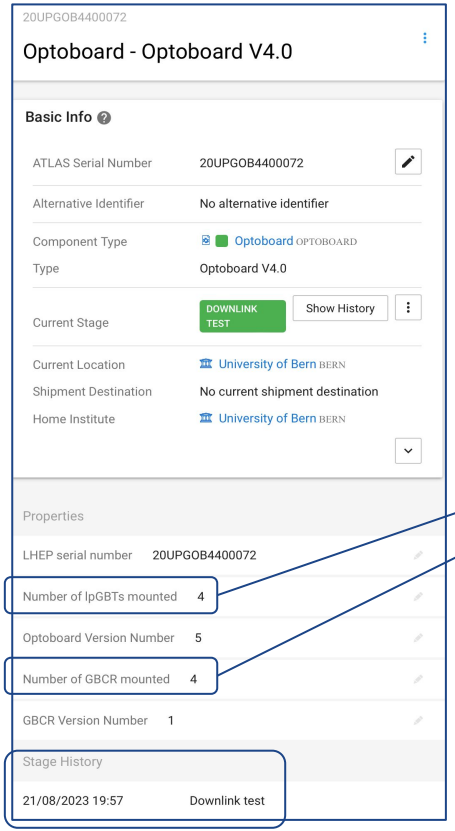
Optoboard GUI: under the hood - Part 2

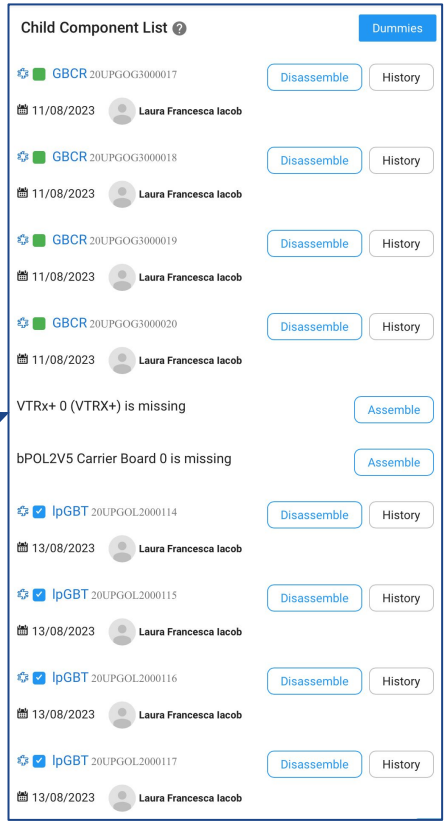


- used for dynamic service discovery with automatic registration of labelled containers
- single source of truth for runtime configuration

ITk Production Database

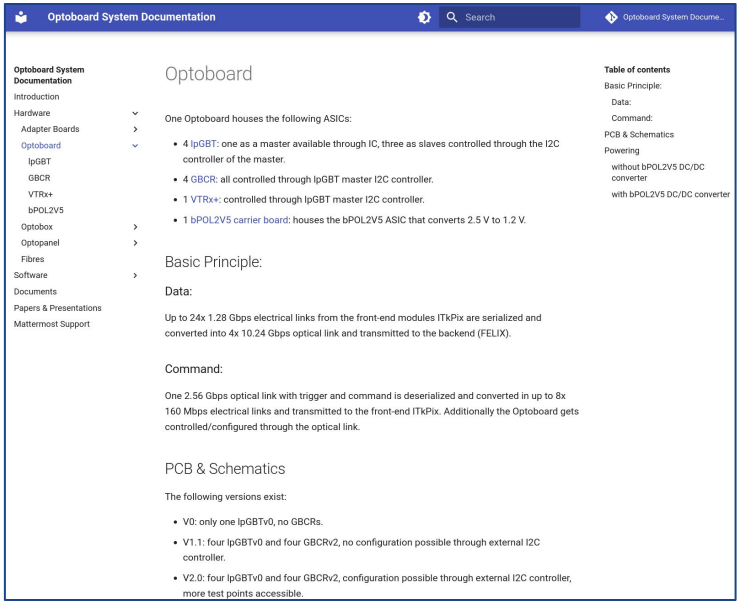
- ITk is made of O(500k) components
- ATLAS ITk Production Database guarantees:
  - component traceability
  - storage of test results
- easy access via [itkdb API]





Optosystem Documentation

- improved documentation
  - automated for code with MkDocs



source

`method` `Lpgbt.read_adc`

```
read_adc(adcchannel)
```

Read the ADC channel of the lpGBT.

Time-out set to 5 s.

Args:

- `adcchannel` (int): ADC channel to measure, from 0 to 15

Returns:

- `adcval` (int): ADC value of the channel

Raises:

- `ValueError` : wrong ADC channel input
- `TimeoutError` : ADC conversion takes to long

ITk Pixel Optosystem

6th September 2023

Daniele Dal Santo

UNIVERSITÄT BERN

AEC
ALBERT EINSTEIN CENTER
FOR FUNDAMENTAL PHYSICS

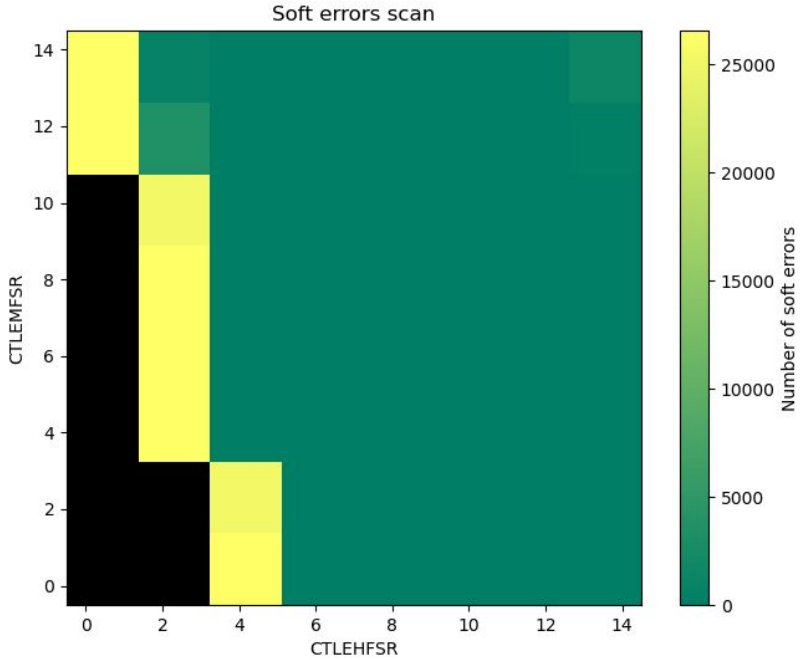
ATLAS EXPERIMENT

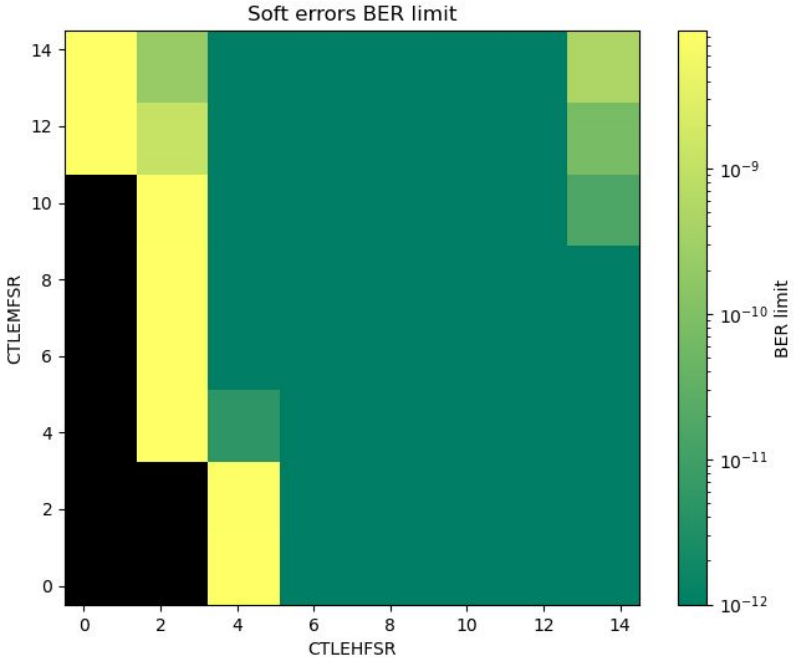
Info on soft error counter

- info about the structure of the ITkPix output data: Manual for the RD53B
  - ITkPix encodes 64b in 66b adding 2b of preamble that can take values 01 (data) or 10 (idle or register block)
    - idle signal contains 2b of preamble + 0x78 + Aurora code -> 18b whose validity can be checked
    - to unscrumble a xor operation is performed between 1) current bit 2) bit in position -38 3) bit in position -57 => errors propagate to the headers and around 62% of the bits are effectively checked
- scrambling is applied by both ITkPix and lpGBT
- unscrumbling applied on scrambled data == identity transformation unless there are errors introduced by the data transmission between the two operations

Results Molex

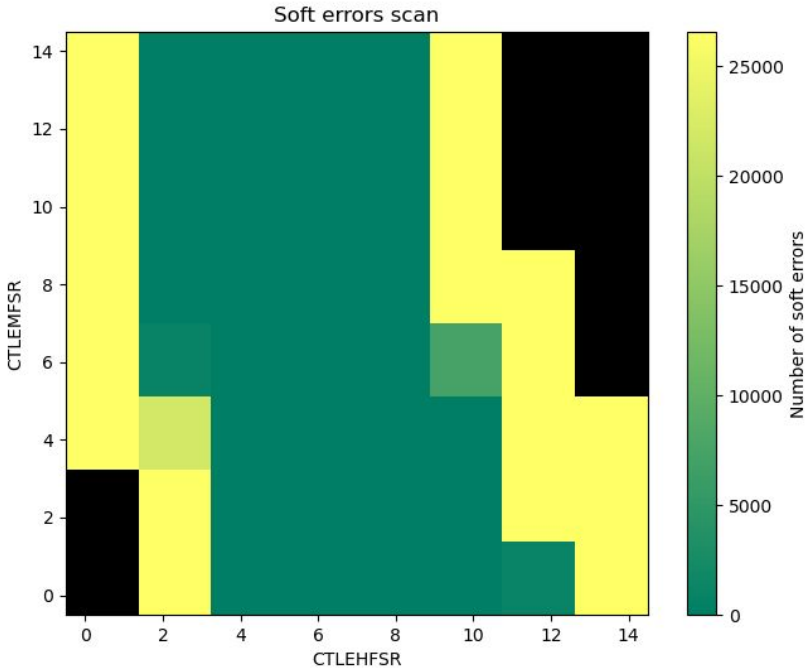
- 3800s for each heatmap point
- 43/64=67% <= 10^-12; 42/64 with no errors

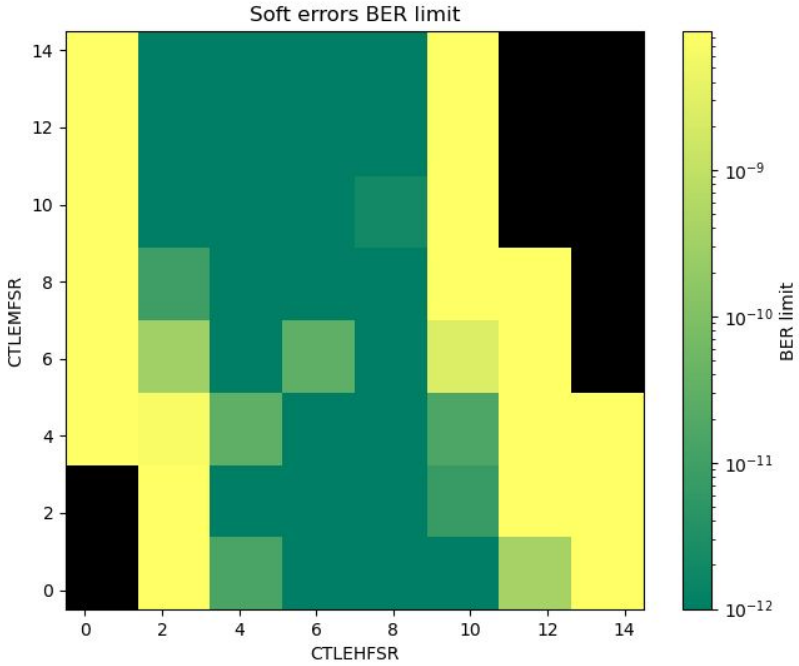




Result Samtec

- 3800s for each heatmap point
- 27/64=42% <= 10^-12; 24/64 with no errors





Microservice structure

- UI, api, celery, rabbitmq on localhost
- UI static files served by Flask server
- openAPI standard for Flask endpoint definition, Flask binding to 127.0.0.1
- Celery + RabbitMQ to connect Flask with backend