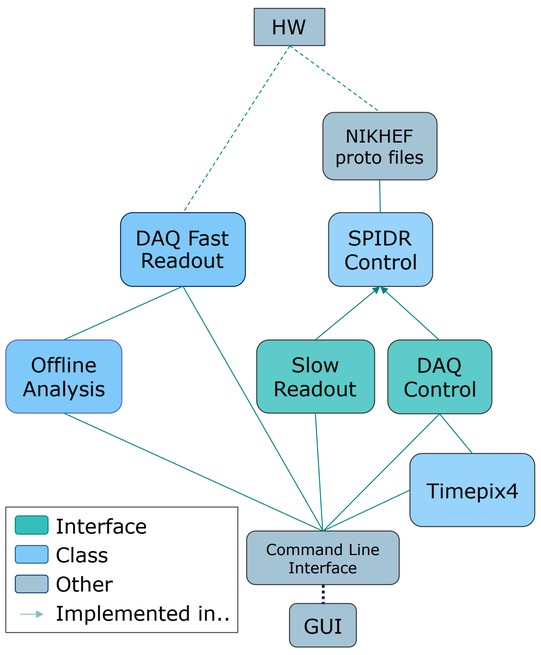
Norfolk, Virginia, USA • **May 8-13, 2023**
CHEP 2023
Computing in High Energy & Nuclear Physics

# A fast and optimized data-acquisition and control software for the Timepix4 ASIC

**V.Cavallini**[2,3], N.V.Biesuz[2], R.Bolzonella[2,3], A.Cotta Ramusino[2]
M.Fiorini[2,3], A.Gianoli[2], X.Llopart Cudie[1], S.F.Schifano[2,3]

1 CERN, Geneva, Switzerland
2 INFN- Sezione di Ferrara, Ferrara, Italy
3 University of Ferrara, Ferrara, Italy

## Software structure




- **Written in C++**
  Fast, Optimized, Low-Level, Object-Oriented
- **Read-out and Control in a unique CLI**
  All you need in a single Command Line Interface program
- **Object-Oriented**
  Software that maps the hardware architecture
- **Flexible Architecture**
  Designed to accomodate various DAQ systems with minimal changes
- **Easy to expand**
  (Optional) bind to Python for fast-plotting
- **Open source**

## Timepix4

Timepix4 is a **hybrid pixel detector readout ASIC** developed by the **Medipix4 Collaboration**.

It consists of a matrix of about **230k pixels** with **55 µm** pitch, equipped with amplifier, programmable threshold discriminator and time-to-digital converter with **195 ps bin size** that allows to **measure time-of-arrival and time-over-threshold**.

It has two different types of links: a slow control link, for the configuration and the slow readout (up to 1 Gbps), and fast links, 16 serial links with a speed of up to 10 Gbps each[1].

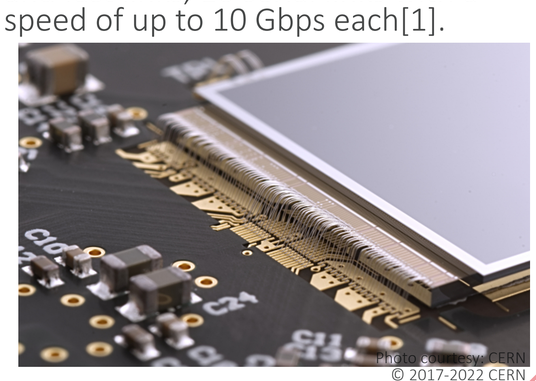


Photo courtesy: CERN


## Data-Acquisition System

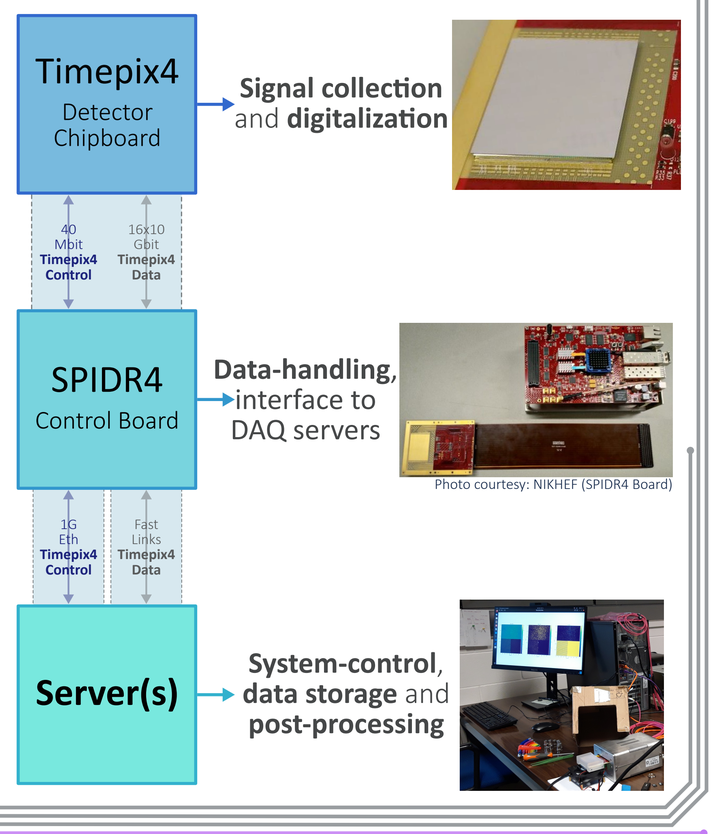


Photo courtesy: NIKHEF (SPIDR4 Board)

## Configuration

### Fast configuration

- **Configure Timepix4 with a simple file**
  Fill the ***.conf*** file customizing the configuration parameters, or leave it with default values. Call the dedicated API to automatically configure Timepix4 according to the file.
- **Full configuration with less than 40 values**
  Fast configuration uses 37 values to fully configure Timepix4. If not specified in the file, default values are used, allowing users to create shorter files.

### Manual configuration

- **Configure Timepix4 using lower-level APIs**
  More than 70 different APIs can be used to manually configure every aspect of Timepix4. You can also create your personalized APIs using the read and write functions for Timepix4 registers.
- **Use both the configuration modes**
  No need to choose between fast or manual configuration, both can be used in the same script.

## Abstract

We present a **new software** for the management of Timepix4 ASIC, based on its data acquisition architecture, entirely **open-source**, **extremely flexible** thanks to the **C++** low-level classes and user-friendly.
The software is organized into a structure of classes that allows to **configure both the Control Board and the Timepix4** and to handle the **slow and fast readout**. Moreover, storage, post-acquisition analysis and every other custom class can be easily added without modifying the basic organization of the software.
This software is **designed to work with every hardware setup** that make use of Timepix4. Changing only a few low-level functions, the ones strictly correlated to the communication protocol, all the higher-level classes will work correctly.

## Data Readout

### Slow readout

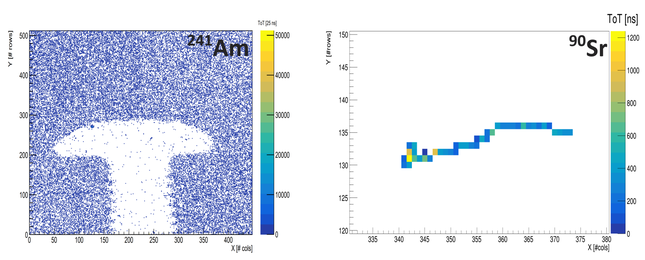
- **Up to 1 Gbit/s**
  Read-out using slow-control link.
- **Continuous polling**
  Acquired data are stored in a dedicated Timepix4 register that is periodically read by the slow read-out thread(s).
- **Dedicated thread**
  Asynchronous thread running in background to get Timepix4 data and write it on file.

### Fast readout

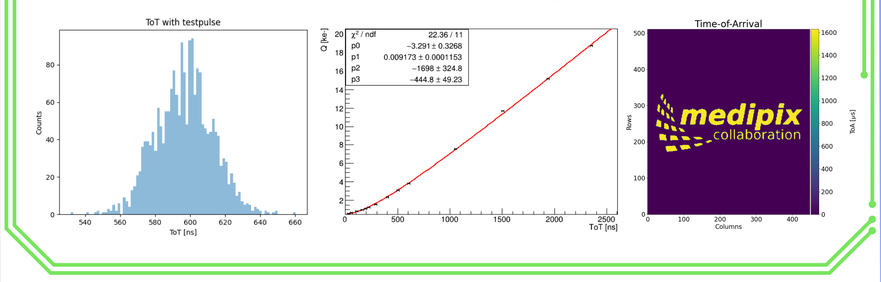
- **Up to 160 Gbit/s**
  Read-out using 16 optic links with a maximum bandwitdh of 10 Gbit/s each.
- **UDP packets**
  Packet are sent directly to the server via UDP protocol, no need for polling.
- **Dedicated threads**
  Two asynchronous thread, using a reader-writer paradigm, running in background to catch Timepix4 data and write it on file.
- **Independent from Control Board**
  Threads don't require reading or writing registers, so they can be used with every Control Board.

## Results

**Acquisition with radioactive sources**
Data-Driven Mode

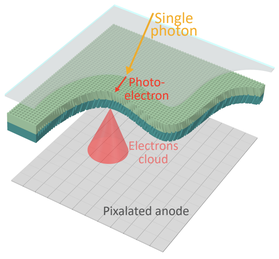



**Calibration with testpulse**
Data-Driven Mode




## Possible future applications

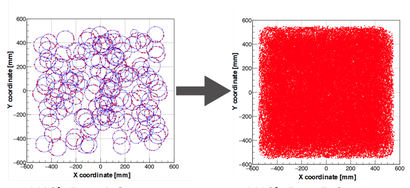
### 4DPHOTON




Development of a new photodetector with large active area able to measure single photons with simultaneous excellent timing and spatial resolution, with a low noise level at room temperature[2][3].

### LHCb - RICH

**LHCb RICH detector Upgrade II**
Need for a new photodetector with fine granularity (<1 mm) and time resolution <100 ps (the smaller the better)




**LHCb Run 1-2**
- $\mathcal{L}$ ~ $4\times10^{32}$ cm$^{-2}$s$^{-1}$
- ~2 k photons in 25 ns
- ~100 rings

**LHCb Run 5-6**
- $\mathcal{L}$ ~ $1\text{-}2\times10^{34}$ cm$^{-2}$s$^{-1}$
- ~100 k photons in 25 ns
- ~5 k rings

[1] X.Llopart et al. "Timepix4, a large area pixel detector readout chip which can be tiled on 4 sides providing sub-200 ps timestamp binning" JINST 17 (2022) no. 01, C01044
[2] M.Fiorini et al, "Single-photon imaging detector with O (10) ps timing and sub-10 µm position resolutions" JINST 13 (2018) no.12, C12005
[3] J.A.Alozy et al. "Development of a single-photon imaging detector with pixelated anode and integrated digital read-out" JINST 17 (2022) no.06, C06007

## Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 819627).
This work was carried out in the context of the Medipix4 Collaboration. The authors would like to thank in particular the CERN and NIKHEF teams for useful discussions, help and support.