ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPGW021

GENERIC DIGITIZATION OF ANALOG SIGNALS AT FAIR – FIRST PROTOTYPE RESULTS AT GSI

Ralph J. Steinhagen*, Ralph C. Bär, Andreas Franke, Alexander Krimm, Kai Lüghausen, David Ondreka, Alexander Schwinn, Matthias Thieme, GSI Helmholtzzentrum, Darmstadt, Germany

Abstract

FAIR operation and notably the new FAIR Control Centre will be based on a 'fully-digital' control paradigm for which about 500 generic digitizers covering analog bandwidths and sampling frequencies from a few MHz to a GHz will be deployed. The aim is to acquire all pertinent accelerator systems and beam parameters to facilitate a multi-mission of continuous performance tracking, (semi-)automated feedbacks and setup tools, early detection and isolation of hardware failures or near-misses, and to provide a dependable generic platform for equipment experts that enables post-mortem analyses or remote diagnostics.

The goal of the controls integration was to provide a generic abstraction of the vendor-specific electro-mechanical form-factor and software interfaces based on modern software-defined-radio (SDR) principles [1]. In addition to ns-level-synchronised time- and frequency-domain based acquisitions, the interface provides a wide range of generic user-configurable signal post-processing routines common for SDRs and also found in many modern bench-top oscilloscopes, spectrum or vector-network analysers.

The acquired raw and derived signals are exported to the FAIR control system using a standardised front-end software architecture (FESA) and a common middle-ware (CMW) [2–4]. Further integration goals were to simplify possible future extensions, compactness, readability, reusability, testability, and long-term maintainability of the code-base which led to the re-use of established open-source signal processing and data fitting frameworks such as GNU-Radio and ROOT [1, 5, 6]. While explicitly kept open for new or other specific digitizers or SDRs, the initial integration, prototyping, and testing have been done for the PS3000, PS4000, and PS600 series of digitizers from Pico Technology [7].

INTRODUCTION

The generic digitization project aims to provide data acquisition (DAQ) to (e.g. purely analog) devices that do not provide their own DAQ or those that have requirements on signal bandwidth, dynamic range, ADC resolution, or complex signal post-processing that cannot be met by the standard embedded front-end controller systems (FEC) for FAIR [8, 9]. The system improves upon OASIS's original 'distributed time-domain oscilloscopes' concept described in [10, 11], extends the concepts also for frequency-domain signals, and – based on more than a decade of experience with such systems – deploys more modern commodity-type hardware, separation-of-concerns (SoC) and new software-defined-radio technology concepts.

* R.Steinhagen@GSI.de

DEPLOYMENT ARCHITECTURE

In 2003 high-performance digitizer used to be very costly. Hence one digitizer was typically shared among several systems, their signals routed often through long signal cables to RF relay switching matrices, before being actually digitized. As of 2019, even high-performance digitizers became much more affordable and the RF switches and long cables started to dominate the total costs and complexity of the given installation.

Thus, FAIR employs a one-digitizer-per-device paradigm with the DAQ placed as close to the accelerator device as possible, minimising cable lengths, omitting costly RF switching matrices and their reliability issues, thus also optimising the signal integrity, and in turn crucially minimising the controls integration complexity. This paradigm change and the use of more powerful server-type CPUs also facilitates more modern, device-specific SDR-type signal post-processing further described below.

CONTROLS INTEGRATION

The selected commodity hardware provides good analog and digital performances, reliability figures, long-term supply chain availabilities and competitive total cost of ownership that rivals most custom-, in-house or specific purpose-built systems (based e.g. on VME, NIM, Micro-TCA etc.).

Electro-Mechanical Integration

In order to nevertheless provide the ruggedness, remote monitoring, and form-factors required in an industrial accelerator environment, we designed an open-hardware electro-mechanical integration for the digitizers operationally deployed at GSI and FAIR, shown in Figures 1 and 2.

Figure 1: GSI-designed integration of PicoTech's PS3000-series digitizer into a rugged 19"-1U form-factor.

The small modular 19"-1U form-factor facilitated also an easy upgrade of analog equipment in the existing GSI facility which do not have their own dedicated DAQ and that are often located where free rack space is scarce.

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPGW021

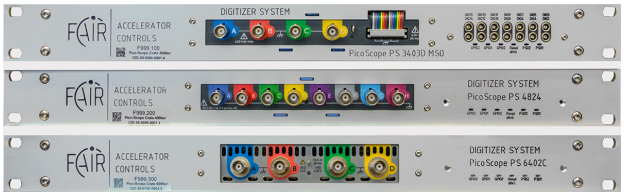


Figure 2: GSI-designed integration of PS3000, PS4000, and PS6000-series digitizer into a rugged 19"-1U form-factor.

About 250 digitizer systems are being deployed at GSI. The form factor will also be extended for SDRs and other digitizer platforms in the future.

Health-Monitoring-Interface

For reliability and availability reasons, each digitizer enclosure is redundantly powered by two power supply units (PSU), its USB power separated from the FEC, and includes a health-monitoring-interface (HMI) that monitors and tests the automatic fail-over circuitry as part of an 'as-good-as-new' verification. It monitors the temperatures of digitizer and PSUs as preventative indicators, and allows both remote software- and hardware-based resets of the enclosure, in the unlikely event of frozen FPGA firmware or issues with the host drivers.

While operation will depend on some digitizers providing input to slow beam interlock systems, the high level of monitored redundancy is primarily needed due to the very large number of systems to be deployed ($\approx$ 500), because many of the digitizers will be installed in locations that are inaccessible during beam operation, and to limit the number of unscheduled repair or maintenance interventions.

As a non-critical functionality, the HMI board furthermore provides simple user-configurable digital IO- and SPI-type interfaces to allow in-field prototyping or to perform, for example, simple (non-critical) control of relays or any other user-specific hardware. The embedded Raspberry Pi may be used as a FEC itself – albeit in a non-operational prototype environment with limited ms-scale timing synchronisation.

Front-End Controller Software Integration

An overview of the FEC software architecture stack is shown in Figure 3. The deployed digitizers provide up to 8 analog channels and up to 16 digital external trigger ports, out of which one is usually connected to the FEC's timing receiver card, while the others may be connected to the user-defined device-specific digital signals. In order to precondition the device's specific analog and digital signals to fit the digitizer input constraints (amplification, filtering, mixing, clamping, etc.), equipment owners are responsible to provide analog front-ends, if necessary.

The digitizer FEC consists of a 1U form-factor high-performance commodity server, may control up to four digitizers, and houses a White-Rabbit (WR) timing receiver to receive and generate precise hardware triggers [8]. The WR receiver signal either triggers the digitizer DAQ directly or is interleaved as a tag to continuously synchronise the transmitted data stream. Throughout the facility-wide timing

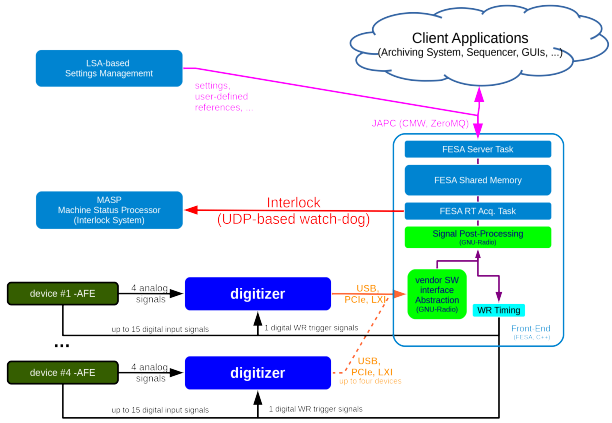


Figure 3: The digitizer's FESA (light blue blocks) and GNU-Radio (green blocks) system overview.

network, the acquisition triggers are cable-length-adjusted and phase-stabilised down to a few nanoseconds across multiple digitizers and other standard FEC-based systems [9].

The primary task of the FEC is to provide an abstraction layer for the vendor-specific physical hardware interfaces (e.g. USB, PCIe, LXI, etc.), low-level driver software and external FESA controls interface to other controls sub-systems and user-level applications (e.g. the Archiving System, Sequencer, GUIs, etc.) [2–4]. This modular abstraction facilitates future upgrades within vendor-specific product lines or integration of products with similar functionalities but different driver or hardware interfaces.

The initial digitizer interface relies on USB 3.0 as a robust standard that is expected to last. Besides the modular separation between digitizer and FEC rack-mounted enclosures, USB is relatively easy to integrate into a Linux software environment. Notably, the USB interface provides substantially higher data transfer bandwidths compared to earlier and in particular oscilloscope-based DAQ controls integrations where these used to be an important bottleneck. The large bandwidth also facilitates SDR-applications with continuous streaming rates in the order of 125 MS/s, as well as triggered time-limited acquisitions of signals with analog bandwidth in the hundreds of MHz to GHz range (base-line: typically 10 acquisitions per second with up to 5 ms long segments sampled at 2 GS/s).

Signal Post-Processing Blocks

As part of the separation-of-concerns paradigm, the low-level DAQ integration is performed using the GNU-Radio framework (GR) and relies on its 'signal-source', '-processing block' and '-sink' paradigm [1]. In this context, the digitizer has been integrated as a GR source and the interface to FESA as a GR sink as exemplarily shown in the simple GR signal flow-graph in Figure 4.

The different post-processing modules may be combined into online and user-definable chains. While the GNU-Radio library already provides a myriad of signal processing blocks, a limited number of additional post-processing blocks have been implemented that are less common in SDR applica-

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPGW021

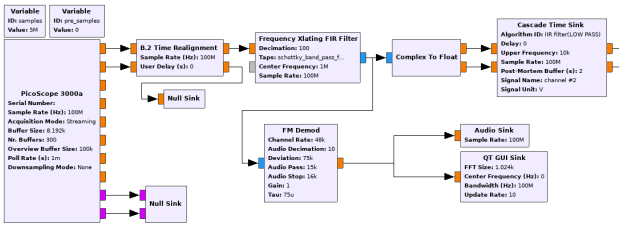
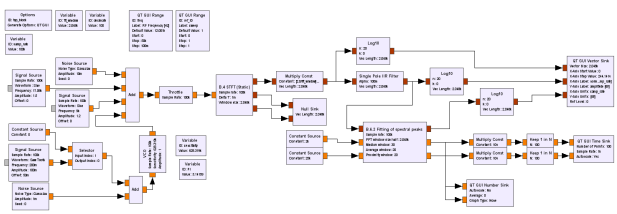


Figure 4: GR signal-graph implementing an FM receiver using a digitizer's data-source, audio and FESA sink.

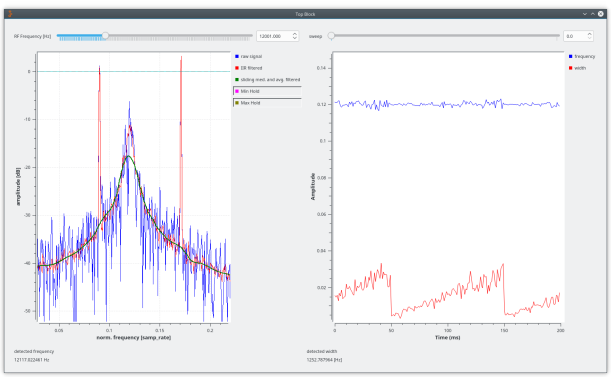
tions but that are frequently used in an accelerator operation and post-processing context such as spectral peak finding routines, frequency-domain fitting, and generalised time-domain χ^2-type fitting routines. In order to simplify further extensions, compactness, readability, re-usability, and maintainability of the implementation, open-source signal processing and data fitting libraries are being used for the post-processing [1, 5, 6].

This SoC abstraction allows those with signal processing expertise and little to no knowledge about controls processes to focus on the required post-processing, and those with controls software expertise to focus less on the specific devices and signal post-processing, as the post-processing can be tested independently by the device expert either with simulated data or the same hardware in the laboratory without FESA using GR's `gnuradio-companion` toolkit.

An example unit-test implementation is shown in Figures 5(a) and 5(b) that show the flow-graph under test and corresponding GUI output that is available to the developer and system expert the `gnuradio-companion` tool.



(a) Unit-test definition of a GNU-Radio post-processing sketch.



(b) Corresponding gnuradio-companion GUI snapshot of the unit test.

Figure 5: Unit test definition for the signal post-processing module performing a spectral peak finder and fitting routine, and corresponding GUI test output.

MEASUREMENT EXAMPLES

The first prototype digitizer and FEC FESA server have been deployed during the 2018/2019 experiment run to assess the integration, operational deployment processes, system interdependencies and other aspects that could not be tested in a lab-only environment during the initial design phase. These tests were very valuable and provided the opportunity to reassess and improve some of the system's aspects prior to moving to the large scale where modifications become more difficult. Figures 6 and 7 provide some examples of the first signals measured and integrated via the new generic digitizer framework.

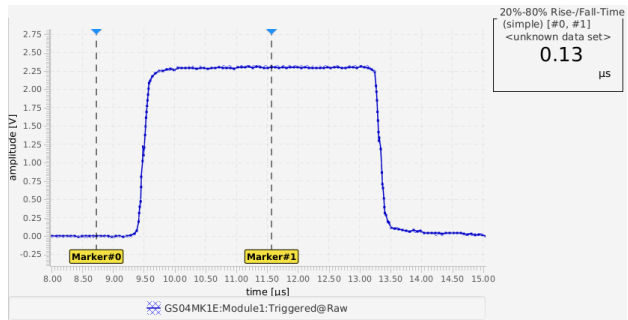


Figure 6: Example SIS18 extraction kicker waveform.

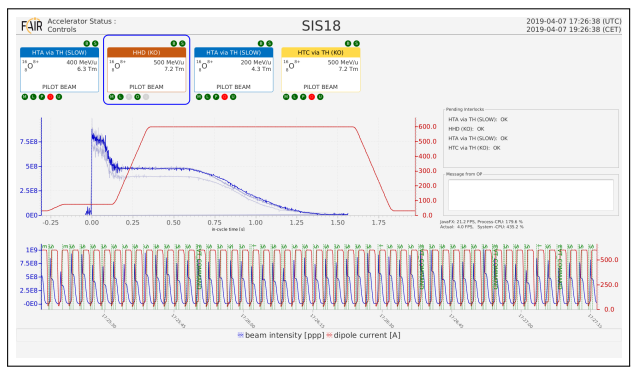


Figure 7: Machine status display showing the different running beam-production-chains, interlock status summary, digitized main dipole current and beam intensity as a function of time. The display is available both in the control room and via a web-site.

An important lesson learned: while the low-level digitizer, DAQ schemes, FESA server, and front-end communications have been tuned for a high throughput performance, it turned out that for the large number of signals and new opportunities to combine signals from vastly different devices, the top-level GUI framework needed some improvements in terms of latency and display performance as reported in [12].

ACKNOWLEDGEMENTS

The authors express their thanks to Janez Golob at CosyLab for providing help with the initial prototyping and digitizer driver integration into the GNU-Radio frame-work.

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPGW021

REFERENCES

[1] GNU Radio Website. (accessed May 2019). [Online]. Available: http://www.gnuradio.org

[2] M. Arruat, L. Fernandez, S. Jackson, F. Locci, J.-L. Nougaret, M. Peryt, A. Radeva, M. Sobczak, and M. V. Eynden, "Front-End Software Architecture (FESA)," in *Proc. 11th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'07)*, Oak Ridge, TN, USA, Oct. 2007, pp. 310–312.

[3] S. Matthies, H. Bräuning, A. Schwinn, and S. Deghaye, "FESA3 Integration in GSI for FAIR," in *Proc. 10th Int. Workshop on Personal Computers and Particle Accelerator Controls (PCaPAC'14)*, Karlsruhe, Germany, Oct. 2014, pp. 43–45.

[4] V. Rapp and W. Sliwinski, "Controls Middleware for FAIR," in *Proc. 10th Int. Workshop on Personal Computers and Particle Accelerator Controls (PCaPAC'14)*, Karlsruhe, Germany, Oct. 2014, pp. 4–6.

[5] R. Brun and F. Rademakers, "ROOT: An object oriented data analysis framework," *Nucl. Instrum. Meth.*, vol. A389, pp. 81–86, 1997.

[6] "ROOT – A C++ framework for petabyte data storage, statistical analysis and visualization," *Computer Physics Communications*, vol. 180, pp. 2499–2512, Dec. 2009.

[7] Pico Technology Limited. (2019). [Online]. Available: https://www.picotech.com/

[8] P. Moreira, J. Serrano, T. Wlostowski, P. Loschmidt, and G. Gaderer, "White Rabbit: Sub-nanosecond Timing Distribution over Ethernet," in *Proc. of 2009 International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. Brescia, Italy: IEEE, Oct. 2009, pp. 1–4.

[9] R. Bär, "News from the FAIR Control System under Development," in *Proc. 10th Int. Workshop on Personal Computers and Particle Accelerator Controls (PCaPAC'14)*, Karlsruhe, Germany, Oct. 2014, pp. 37–39.

[10] S. Deghaye and D. Jacquet and I. Kozsar and and J. Serrano, "OASIS: a New System to Acquire and Display the Analog Signals for LHC," in *Proc. 9th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'03)*, Gyeongju, Korea, Oct. 2003.

[11] L. Bojtar and C. Charrondiere and Y. A. Georgievskiy and F. C. Peters and I. S. Zharinov and Stephane Deghaye, "OASIS Evolution," in *Proc. 11th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'07)*, Oak Ridge, TN, USA, Oct. 2007, pp. 322–324.

[12] R. J. Steinhagen, H. Bräuning, A. Krimm, and T. Milosic, "Redesign of the JavaFX Charts Library in View of Real-Time Visualisation of Scientific Data," presented at IPAC2019, Melbourne, Australia, May 2019, paper THPRB028, this conference.