

MODULAR SOFTWARE ARCHITECTURE FOR THE NEW CERN INJECTOR WIRE-SCANNERS

A. Guerrero[†], D. Belohrad, J. Emery, S. Jackson, F. Roncarolo,
European Organization for Nuclear Research (CERN), Geneva, Switzerland

Abstract

In the scope of the LHC injector upgrade, new wire-scanner devices have been installed in the LHC injector circular accelerators. This paper outlines the software architecture and choices taken in order to provide the scanner experts with comprehensive diagnostics as well as operators with straightforward size measurements. The underlying electronics acquire large amounts of data that need to be accessible for expert and machine development use and need to be processed before being presented for daily operational use, in the shape of a beam profile and its derived size. Data delivery and measurement computation are accomplished by means of a modular structure, using functionally distributed real-time processes that handle the different data views, with minimal interference in the processing, and minimal exchange of data among modules.

INTRODUCTION

A wire-scanner is generally based on a thin wire traversing at high-speed circulating particle beams. Monitoring turn by turn the distribution of secondary particles generated by the beam-wire interaction (bottom plot in Figure 1) as a function of the wire position (top plot in Figure 1), allows reconstructing the transverse beam profile.



Figure 1: The top plot shows position data for both in and out scan. On the bottom plot beam profile data is displayed.

Such measurements are performed daily by accelerator operators and experts to infer the size of the beam and via the beam optics and beam momentum spread the transverse emittance of the different beams played in the machines.

At CERN a new generation of Beam Wire-Scanners (BWS) has been developed, installed, and recently commissioned [1] in the scope of the LHC injector upgrade (LIU), during the accelerator restart after the long shutdown (LS2). New engineering concepts were applied to build the hardware and electronics of these devices [2] [3] [4] [5], thus giving rise to a new complete software system built and constrained within the available controls infrastructure and underlying standards. In particular, the Front-End Software Architecture (FESA) [6] [7] was chosen as

the framework for the design and development of the system modules.

SYSTEM DESCRIPTION

The BWS applies key innovations to its kinematic unit: moving parts located only in vacuum, magnetic and optical techniques to power and air vacuum signal exchange. The unit performs trajectory control of the shaft via a solid rotor resolver [8] and provides carbon wire measurement and a high accuracy position measurement using an optical encoder [9]. Electronics driving the system consist of an Intelligent Drive (BWSIDC) in charge of the powering and wire movement and an acquisition crate collecting data from the different actuation parts as well as from the optical encoder measurements. Diagnostics and tests possible in *local mode* at this level include verification of cabling and parts of the kinematic subsystem, powering and scanning procedures without beam interaction for diagnostics, all without tunnel access.

Communication with kinematic control and acquisition electronics is done through Ethernet by means of the IPbus protocol and associated module and libraries for firmware and software development [10]. The IPbus protocol is a simple packet-based control protocol for reading and modifying memory-mapped resources within FPGA-based IP-aware hardware devices which have a virtual A32/D32 bus, thus enabling the replacement of VME control (a usual standard in our operational instrumentation) in our case.

The secondary particle shower detection is performed by a scintillator coupled to four Photo-Multiplier Tubes (PMT) to acquire four signal amplitude ranges with the aim of covering the large dynamic range of beam energies and intensities across the LHC injectors [11]. The PMT devices are powered with a custom board optimised for large pulse mode and fast recharge. The gain of the PMT is set via a high voltage power supply controlled through a 4-channel commercial board accessed through the VME bus. PMT output currents are amplified in two stages and driven into parallel high-speed digitizers, feeding the so called VFC [12]. The VFC is a CERN FPGA-based multipurpose carrier with VME interface, designed to be the new standard acquisition platform for the Beam Instrumentation Group. The VFC wire-scanner application firmware was built in house and adapts to the different accelerator synchronisation. The acquisition takes place asynchronously to the bunched beam at 500MHz and with a 14-bit resolution. Together with the sensor current outputs, the beam revolution frequency and bunch timing are written into a memory bank composed of two 8Gb DDR chips. On request, acquired digitised beam information is integrated on the fly and placed in the VME bus for the CPU to recuperate

[†] ana.guerrero@cern.ch

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

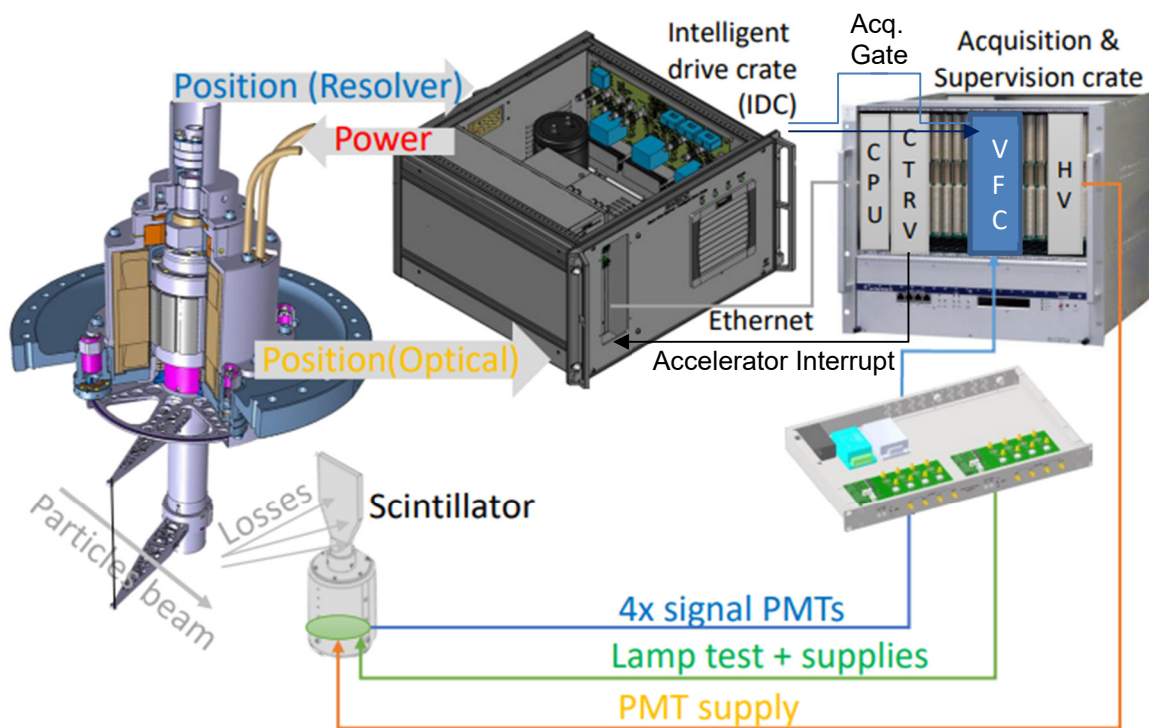


Figure 2: From left to right, mechanics installed in the accelerator, secondary particle detector, IDC, digitisers and VME crate containing the different VME acquisition and control boards.

through VME fast block data transfer. Data transfer is performed using an in-house product called EDGE that automatically generates a driver from a database description. The VFC firmware also controls the amplifier stage and a lamp for test purposes through an UART interface.

The synchronisation to machine timings of the different elements involved in the measurement following a user's request for a specific time in a concrete cycle is achieved by programming an interrupt in a VME timing receiver board (CERN product) that will send a pulse to the BWSIDC. The controller is then triggering the start and end of the acquisition gate in between rotational shaft angles corresponding to the window of measurement provided by the user. These angles can be pre-set using the data derived from a calibration function computed in the laboratory during the pre-installation phase.

Figure 2 shows a diagram of the full hardware system.

The front-end software is running on a MEN-A25 CPU with CENTOS Linux which communicates with the control and acquisition boards either through the VME bus or through Ethernet. It is composed of 5 different FESA classes, each one mostly mimicking the functionality emerging from the hardware design and encapsulating the low-level data corresponding to the main function performed: wire movement across the beam pipe, secondary particle integration for bunch profile construction, PMT working point parametrisation, measurement synchronization and finally operational beam profile measurement.

FESA

Currently, the standard infrastructure for equipment front-end software in the CERN accelerators, the Front-End Software Architecture (FESA) is a complete environment for equipment specialists to design, develop, deploy, and test their equipment software, which produces as an end product a FESA class and FESA devices. The focus of the instrument developer is set on the structure and the flow control of the application domain, and to standardise the outcome of this analysis into a framework. The framework orchestrates the activity, real-time processes and user interface exchanges calling routines provided by the application developer to apply the equipment-specific behaviour.

The internal real-time functionality can be as modularised and parallelised as allowed by the constraints imposed by the underlying hardware components of the system. The outcome of the different threads or concurrency layers talking to the external hardware is stored in a shared memory accessed also by the defined interface. An operational FESA class receives input parameters from database declared fields for server initialisation and from external sources via a predefined interface using a device-property model. The acquired and processed data is retrieved via the same mechanism.

Thus, all data exchanges are done through one server dealing with all the requests coming from the different user scripts and applications accessing the class. In our case, expected applications accessing the class are hardware expert's Python scripts to retrieve hardware status information and raw data, Java and Python expert applications

to control and display low level information for instrument diagnostics, operational applications to produce measurements and set parameters interesting to operators and accelerator specialists, multipurpose WEB interfaces [13], ad-hoc scripts launched during machine development periods while studying the machine behaviour or the instrument behaviour in test conditions and last but not least, data retrieval for long term storage.

FUNCTIONAL MODULES

As we have seen in the system description, the hardware and electronics composing the BWS system have two fundamental functional modules controlling and acquiring both sensors and actuators, one handling the movement of the wire and one in charge of the particle distribution acquisition. The only link between them is the hardware signal triggering the acquisition gate. The parameters involved in the generation of this trigger are an essential concern for meaningful profile measurements in cycling machines where the cycle time at which the wire is flying in the beam is precisely requested by the user performing the scan and necessarily synchronised to the played cycling sequence. It is this synchronisation functionality using the timing infrastructure of the accelerator which makes up a third functional module and drives the former two modules. These three quasi-independent modules become integrated with one another by means of a fourth module, the operational bunch profile measurement, dedicated to the construction of the requested measurement. For practical reasons, we decided to split the particle distribution acquisition into two different modules, namely bunch profile acquisition and PMT gain thus making up a total of five functional modules.

More and more specialists and operators want to access system data and beam data at different processing levels. The functionality and type of the exchanged data between the FESA server and the users may be as varied as the type of user looking at and/or actuating them, but we can differentiate three main types: machine operators, accelerator specialists and system specialists. The described functional decomposition of the software enables at the same time an easy encapsulation of the data to provide different data views of the system and different security access depending on the user.

Wire Displacement

Hardware-wise the wire displacement is the most complex functionality of the system, however this is not the case at the software level. This module is in charge of starting a scan with or without external synchronisation and of reading a bulk of registers with acquired data and status. No processing of data is required for publishing. The decision to design this module separately also obeys two particularities at the functional internal level, the introduction of the IPBus protocol for the data transfer on one hand and the handling of critical settings on the other. Whereas in the other modules errors in the parametrisation of the devices do not have consequences other than a wrong measurement, this module includes a few parameters that could

potentially disturb the feedback in the movement or the lifetime of some of the instrument components. The firmware implements all needed protections to avoid any harm to the instrument but nevertheless a strict policy must be applied to the initialisation and modification of critical settings. The access to the module should be confined to the instrument experts.

Regarding the choice of data transfer between the BWSIDC and the software module, the use of a VFC was initially considered to profit from the tools and automatic driver generation from the EDGE register description. Finally, we opted to use ethernet with IPbus as the communication technology to reduce the VME bus data traffic at the same time as allowing parallelising the data transfer of the modules retrieving most of the acquired raw data. In order to maintain the standard register description, the module initialisation required the incorporation of a translation of the current EDGE description into the XML description supported by the UHAL library of IPbus.

Bunch Profile Acquisition

This module deals with the different aspects of the bunch transverse profile construction from the digitised particle shower distribution detected: It prepares the VFC and amplifier stage for acquisition and reads four integrated data streams and the raw data of a non-saturated channel for publication. From the firmware signal level detection facility, it tries to derive the ideally ranged channel. When requested, attempts to compute the bunch phase with respect to the retrieved or produced bunch clock depending on the accelerator concerned for correct integration of the bunch. Both the PS and PSB injector electronics do not have access to the bunch timing but only to the revolution frequency, however the harmonic used is known before-hand. The bunch timing is thus generated by the firmware dividing the turn length by the harmonic. The phase will depend on the type of beam, scan cycle-time requested and radio frequency signal offset.

The main internal requirement for handling the bunch acquisition is high speed data transfer due to the large amount of raw data produced in the four digitalisation channels.

For test purposes a lamp control is also included in its functionality.

PMT Gain

One could argue that this functionality can be integrated in the bunch profile acquisition module since it is an essential working component of the beam signal amplitude. However, it is distinctly decoupled from the hardware point of view and at the same time, needs no dynamic control which is a key point for adding this module. As the system has been conceived, the working point of the PMT filter combination needs tuning only during a commissioning phase to correctly cover the ranges of energy and intensity without neither saturating all channels nor decreasing the signal to noise ratio too much.

There is also a practical reason for this choice as discussed later.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

Machine Timing Synchronisation

Machine synchronisation and timestamp related functionality is handled within this module.

All LHC injectors are cycling machines. The selection of the cycle and time in the cycle to scan is a must for a meaningful size measurement and therefore the actuation has to synchronise with machine events to respect within one or two milliseconds, depending on the accelerator, the wire beam crossing time. When the user requests a certain cycle time an interrupt is programmed taking into account the measured time for the wire to fly to the centre of the vacuum chamber, according to a calibration function assigning a position in the chamber to the angle of the rotating fork. The timestamp generated by the interrupt allows posterior data correlation across modules.

Operational Bunch Profile Measurement

The transverse size measurement starts from a user request holding a few already mentioned parameters such as cycle, cycle-time, applicable harmonic number in some cases, and others such as speed and acquisition window. This module takes care of the distribution of commands and settings to the other modules for scan preparation. Once the interrupt is programmed, the unit simply awaits the notification from the other modules that the different requested data are available. When the trigger fires, the derived events will start the corresponding sequence of tasks in each of the acquiring modules. Each sequence ends in the notification of the success if the requested data is ready or, error, in case a problem was encountered, so that the error description can be retrieved. When all needed data is rendered, the correlation of the position and amplitude of the integral of each of the acquired turns takes place together with the size computation by fitting the constructed bunch profile to a gaussian.

ARCHITECTURE

The main purpose of the BWS system is to provide bunch profile and size measurements in accelerator operations. Whereas from an operational perspective the complexity of the system is hidden, the specialized usage and tuning requires access to the large amount of data and status produced while scanning by the different components of the system. Comprehensive diagnostic and testing tools need interfacing to maintain and monitor the system.

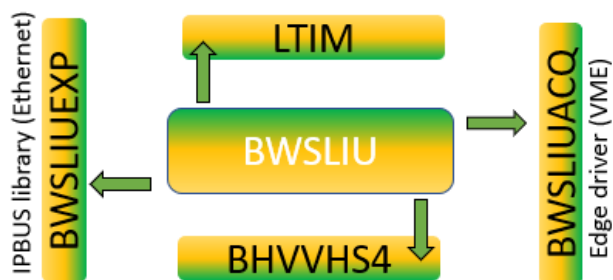


Figure 3: BWS software system event-driven architecture. Central unit FESA class with four satellite FESA classes, each representing a functional module.

The high degree of independence of the functional modules emerging from the underlying hardware design and the distinct nature of the data produced, allows the implementation of software modularisation where one coordinating unit, the operational bunch profile measurement module, orchestrates via few operation settings, the other modular components dealing with highly specific tasks. This organisation (Fig. 3) fits an event-driven architecture, using FESA classes as modules, with one central BWSLIU class having the capability of orchestrating the operations and the other satellite classes BWSLIUEXP, BWSACQ, LTIM and BHVVHS4 performing their concrete task and exposing the associated data. Such modularisation takes into account prior existence of two FESA classes, LTIM and BHVVHS4 that could be incorporated without changes only by splitting the bunch acquisition module into two, one dealing with the actual acquisition, the other handling the fine-tuning of the working point.

While FESA is very flexible and the described modular structure could be designed inside the framework, an approach to push the decoupling of the modules by using a class per functional unit adds several advantages:

- Delocalisation of modules not using VME bus, for relaxing the load of the CPU both in memory and processing.
- Interface distribution, the server is not unique anymore with higher capability of absorbing requests from the varied application set, thus, sharing the load of data publication. In this way, each set of functional users also have their encapsulated view of the system.
- Implementation in smaller units which are easier to maintain, modify and use.
- Module additions or substitutions without interference on the other units. This type of modularisation allows easy prototyping, testing, and introduction of simulation blocks when a part of the hardware is not available.
- Facilitates the test procedure. During the prototyping and testing phase multiple procedures were put in place to validate and diagnose the system during hardware commissioning and individual system tests. These can be launched without all the system being in place.

CONCLUSION

The implementation of FESA classes as building blocks in a functionally modular structure, in this case using an event driven architecture, grants high flexibility to the system in terms of prototyping and testing. It also simplifies coding structures that facilitate maintenance, encapsulates data views with extensive diagnostic possibilities and focuses interfaces adapted to the type of user exploiting them.

Following the restart of the LHC injectors the system has been deployed successfully in each of the accelerators. The commissioning was certainly facilitated by this modularisation due to the formerly described assets.

REFERENCES

- [1] J. Emery *et al.*, “Commissioning of the LHC Injectors BWS Upgrade”, in *Proc. 10th Int. Beam Instrumentation Conf. (IBIC'21)*, Korea, Sep. 2021, to be published
- [2] M. Koujili *et al.*, “Fast and high accuracy wire-scanner”, in *Proc. 9th European Workshop on Beam Diagnostics and Instrumentation for Particle Accelerators (DIPAC'09)*, Basel, Switzerland, May 2009, pp. 188–190.
- [3] M. Koujili, Y. Ait-Amirat, B. Dehning, A. Djerdir, J. Emery, J. H. Alvarez, “Design of an actuator for the fast and high accuracy wire scanner”, in *2011 IEEE International Electric Machines Drives Conference (IEMDC)*, May 2011, pp. 1450–1455. doi: 10.1109/IEMDC.2011.5994822
- [4] R. Veness *et al.*, “Experience from the Construction of a New Fast Wire Scanner Prototype for the CERN- SPS and its Optimisation for Installation in the CERN-PS Booster”, in *Proc. 4th International Beam Instrumentation Conference (IBIC2015)*, Melbourne, Australia, September 2015, 2016, TUPB061, pp. 479-482.
doi:10.18429/JACoW-IBIC2015-TUPB061
<https://cds.cern.ch/record/2263484>
- [5] J. Emery *et al.*, “Design and validation methodology of the control system for a particle beam size measurement instrument at the CERN laboratory,” in *2017 American Control Conference (ACC)*, May 2017, pp. 4221–4228. doi: 10.23919/ACC.2017.7963604
- [6] M. Arruat, J-J. Gras, A. Guerrero, S. Jackson, M. Ludwig, J-L. Nougaret, “CERN front-end software architecture for accelerator controls”, in *Proc. of International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'03)*, Gyeongju, Korea, 13-17 October 2003, pp. 342-344.
- [7] M. Arruat *et al.*, “Front-End Software Architecture”, in *Proc. ICALEPCS07*, Knoxville, Tennessee, USA, Oct 2007, WOPA04, pp. 310-312.
- [8] J. Emery, P. Andersson, F. Roncarolo, and Y. Thoma, “A low fluctuation control strategy for PMSM direct drive system targeting Particle Beam Instrumentation Application” in *Proceedings, 3rd IEEE Conference on Control Technology and Applications (CCTA2019)*, Hong Kong, China, August 19-21, 2019. doi: 10.1109/CCTA.2019.8920688
- [9] J. L. Sirvent, J. Emery, and J. M. A. Poveda, “Design of an optical fibre based angular position sensor for wire scanners complying with ultra-high vacuum, high temperature and radiation conditions of the CERN’s accelerators”, 2012, <https://cds.cern.ch/record/1491608>
- [10] C. Ghabrous Larrea *et al.*, “IPbus: a flexible Ethernet-based control system for xTCA hardware”, *JINST* **10** (2015) no.02, C02019. doi: 10.1088/1748-0221/10/02/C02019
- [11] J. Sirvent, “Beam secondary shower acquisition design for the cern high accuracy wire scanner”, Ph.D. dissertation, Barcelona University, Dec. 2018
- [12] M. Gonzalez-Berges, J.O. Robinson, M. Saccani, V. Schramm, M.A. Stachon, “Test-bench Design for New Beam Instrumentation Electronics at CERN”, in *Proc. ICALEPCS2019*, New York, NY, USA, Oct 2019, pp. 323-327. doi:10.18429/JACoW-ICALEPCS2019-MOPHA049
- [13] S. Bart Pedersen, S. Jackson, “Graphical User Interface programming challenges moving beyond Java Swing and JavaFX”, in *Proc. ICALEPCS2019*, New York, NY, USA, Oct 2019, pp. 637-640. doi:10.18429/JACoW-ICALEPCS2019-MOPHA173