THE SIRIUS MOTION CONTROL REPORT

H. D. Almeida, F. P. Figueiredo, M. P. Donadio*, J. R. Piton
LNLS, Campinas, Brazil

Abstract

Sirius is the new 4th generation synchrotron light source being built in Campinas, Brazil. The motion control report was created to describe all the steps taken to choose the set of motors, motor drives, and controllers that the hardware (GAE) and software (SOL) support groups will recommend. The steps include researching motion control systems in other Synchrotron laboratories, talking to the Sirius beamline designers, defining requirements and testing. This presentation describes the report, showing the information gathering process and latest results.

INTRODUCTION

The method to define the motion control solution for Sirius will be based in 7 steps:

1. Description of current motion control systems from the UVX synchrotron at LNLS, Brazil, and also from other laboratories, so that it is possible to have a base reference of what can be done and what are the current established practices related to motion control.
2. Definition of the requirements for the Sirius motion control system. It focuses on deciding, together with the beamline operators and scientists, what will be necessary for the motion control system to provide (precise movements, coordinated movements, fast movements, etc). It will also include requirements suggested by the support groups, either because they are useful (debugging support, installation on standard racks, etc), or necessary (adherence to safety standards, availability of software drivers, etc). Also, it will be considered the current expertise in equipment used in UVX as a requirement.
3. Initial device selection, where the requirements are formalized and the list of possible devices is presented. The result of this step is a filtered and standardized description of everything that beamline operators need from the motion control system and also a list of devices that will be part of the selection process.
4. Definition of tests. A set of tests will be elaborated in a way to identify the devices that comply with the requirements or not. The result of this step is a set of programs, types of equipment and textual descriptions necessary to execute the tests.
5. Execution of tests. All devices that we can acquire from the pre-selected list will be tested as necessary. The result of this step is a list of devices that passed the tests.
6. Reevaluation of tests. This step is basically doing the fourth and fifth steps again, if necessary.
7. Final device selection. The devices will be selected according to tie break rules. The result of this step is the set of recommended motion control devices for Sirius, which complete the report.

CURRENT STATE OF MOTION CONTROL SYSTEMS

The UVX light source at LNLS uses motors in the beamlines to control the beam setup, for example, to place mirrors in position, and also to control the experiments, for example, by selecting the energy from the monochromator.

Controllers

The main controllers used are the Galil DMC-4183 and Parker OEM 750X, which also has an integrated driver. There are a few IMS (integrated controller and motor) models 14A4, 17C4-EQ and 23C7-EQ and a few All-motion, models: EZHR17EN and EZHR23NHC.

Drivers

Basically, there are 3 drivers in use: Galil DMC 4140, Parker OEM750X, and Phytron ZMX+.

Motors and Control System Setup

There are many examples of motors used for beamline setup. For example, there are motors for positioning the first mirrors on the beamline entry point. The mirrors position the beam before entering the monochromator. The monochromator base may also contain motors, which will influence both the entry and exit positions of the beam at the monochromator. After the monochromator, motorized slits will select relevant parts of the beam, for example, the most uniform section of the beam, or some part with certain polarization properties. Another motorized set of slits farther away from the monochromator blocks scattered light. The table that holds all the equipment after the monochromator may rotate starting from the monochromator, to guide the light exiting the monochromator at different angles.

All those motors are used to set up the beamline, for example, to focus the beam at a precise point or to select the energy on fixed energy beamlines. They are set up before the experiments start.

Other motors are used during the experiment. Motors may move inside the monochromator to select the energy while doing scans. Sample holders move to select and position different samples to be tested. The samples or the cameras may be rotated to measure diffraction at different angles. The undulator needs to move synchronized with the monochromator to select different energies.

The motor types used at UVX are stepper motors and piezo. The stepper motor brands are Arsape, Haydon Kerk,

[*] marcio.donadio@lnls.br

ISBN 978-3-95450-189-2

CONTROL SYSTEM EVOLUTION AND THE IMPORTANCE OF TRIAL AND ERROR

P. Duval, M. Lomperski, DESY, Hamburg, Germany
J. Bobnar, Cosylab, Ljubljana, Slovenia

Abstract

In this paper we address the importance and benefits of trial and error in control system evolution. Here we refer to the control systems of particle accelerators and large machines, whose control systems, although complex, will not lead to catastrophe in case of failure. We likewise focus on the evolution of control system software, although the issues under discussion will apply to and are often driven by control system hardware. We shall contrast classical Darwinian evolution via natural selection with control system evolution, which proceeds rather via artificial selection, although there are numerous software memes which tend to replicate according to their 'fitness'. The importance of general trial and error, i.e. making mistakes and learning from them, in advancing the capabilities of a control system will be explored, particularly as concerns decision making and overcoming *Einstellung*.

INTRODUCTION

A mature accelerator control system will be able to address a wide variety of problems which might arise throughout the controlled facility's natural lifecycle. Solving new problems or a push to provide better solutions to old problems will generally lead to control system evolution, even if this amounts to little more than keeping up with industrial or commercial components. The way one goes about problem solving will in turn have a marked influence on the pace of this evolution. We will discuss many of these aspects below, finishing with a few concrete examples of control system evolution at play.

GOALS AND PROBLEM SOLVING

The God Complex and Einstellung

When we are well-versed in our control system and at the same time faced with a new problem or challenge we are apt to fall prey to the God Complex, i.e. that "no matter how complicated the problem, you believe that your solution is correct." [1, 2] This is furthermore often compounded by what psychologists refer to as the *Einstellung* effect, or the "predisposition to solve a given problem in a specific manner even though better or more appropriate methods of solving the problem exist". [3,4]

The danger is not that our problem won't get solved. It most likely will. The danger is that we might not only miss an opportunity to explore new ideas, we might also end up wasting resources, and/or missing the big picture entirely due to our rush to implement a known solution.

Priming and Anchoring

Indeed our choices in problem solving and decision making are often due to an implicit memory effect known as priming [5], where exposure or familiarity with one stimulus (or solution paradigm) can influence our response to another. The classic trivial example: "How many animals did Moses take on the ark?" (answer: 0) might appear to have little to do with our decision making until we realize that our *expectations* can be easily primed, a case of priming known as *anchoring* [5]. For example, imposing an artificial deadline of one week to try some solution automatically suggests a level of difficulty. Worse, refusing to consider a new solution because "everyone else does it differently" suggests a knowledgeable rejection of the new solution. Unfounded expressions such as "one week" or "everyone else" often serve only to anchor our expectations at some level.

Accumulated Advantage

The previous example of anchoring ("everyone else does it differently") is also an example of the effect of accumulated advantage, often referred to as the *Matthew Effect*, (from Matthew 25:29 in the King James version of the Bible) [6]. In point of fact, our opinions are strongly related to and often dependent on those of others. The crowded restaurant *must* serve better food than the empty one next door! In an experiment by Duncan Watts [6], two sets of college students could download garage band music from two web sites. The sites were identical except that in one case the students could see the *likes* and downloads of everyone else. It's not surprising that in one case there were a handful of hit songs and in the other the *likes* and downloads showed a flat distribution.

Trial and Error

We should in any case be aware of the aforementioned challenges to our problem solving abilities. Whether we admit that we already know the solution to a new problem or not, the practice of trial-and-error cannot be avoided. The basic algorithm of trial-and-error can be described as:

1) Define what constitutes a solution to our problem.
2) Try *something*.
3) Check to see if the problem is solved. If not:
4) Modify *something* into a more promising direction and repeat step 3). Or, if the problem is solved:
5) Quit.

ISBN 978-3-95450-189-2

SOFTWARE TESTS AND SIMULATIONS FOR CONTROL APPLICATIONS BASED ON VIRTUAL TIME

M. Hierholzer*, M. Killenberg, T. Kozak, N. Shehzad, G. Varghese,
M. Viti, DESY, Hamburg, Germany

Abstract

Ensuring software quality is important, especially for control system applications. Writing tests for such applications requires replacing the real hardware with a virtual implementation in software. Also the rest of the control system which interacts with the application must be replaced with a mock. In addition, time must be controlled precisely. We present the VirtualLab framework as part of the Chimera Tool Kit (formerly named MTCA4U). It has been designed to help implementing such tests by introducing the concept of virtual time, and combining it with an implementation basis for virtual devices and plant models. The virtual devices are transparently plugged into the application in place of real devices. Also tools are provided to simplify the simulated interaction with other parts of the control system. The framework is designed modularly so that virtual devices and model components can be reused to test different parts of the control system software. It interacts seamlessly with the other libraries of the Chimera Tool Kit such as DeviceAccess and the control system adapter.

INTRODUCTION

To test software automatically, a virtual test environment has to be provided. With the VirtualLab framework the necessary tools for this tasks are available. The framework is part of the Chimera Tool Kit and is available as open-source software [1].

This paper explains the procedures of writing tests for control applications at the example of a low-level RF controller server for FLASH-like machines. The low-level RF system used at FLASH [2] and XFEL [3] uses ADC and DAC boards based on MicroTCA.4 [4], connected to down converters and vector modulators for 1.3 GHz controls. The machine is pulsed with 10 Hz. The control loop for the fast phase and amplitude stabilisation is running on FPGAs on the ADC/DAC boards. Trigger pulses are sent to the FPGA and with delay to the low-level RF controller server. The server is based on the DOOCS middleware and running on the frontend CPUs. It presents the interface to the control system and performs several slow tasks, like generating tables for setpoint, feed-forward, gain etc. based on input parameters provided by the operator, and executes slow control loops for drift compensation and adaptive feed-forward. This controller server is a critical element required for the operation of the machine. Therefore thorough tests are crucial to prevent machine failures and unnecessary down time.

To avert regression failures of the software staying undetected and being included in the production system during the next software update, automated continuous integration tests should be implemented. This requires full automation of the tests.

VIRTUAL DEVICES

Virtual devices can be used to achieve a full automation of tests. In contrast to testing on real hardware devices, tests based on virtual devices can fully govern the function of the device. Faults can easily be injected to test exception handling.

Figure 1 shows the layout of the test example. The test routines take control over the low-level RF controller server to be tested. The server is connected through a dummy register set with a state machine and a control loop algorithm reflecting the relevant behaviour of the FPGA firmware. This algorithm is connected through signal sinks and signal sources with a simple cavity model. These signal sinks and sources help creating the modularity needed for reusing parts of the virtual components for different tests.

A simple example for a test routine is shown in Figure 2. The test routine first sends the command to ramp up the gradient through the control system. Next it waits until the procedure is completed and finally it tests the result by comparing the actual current gradient of the cavity model with the nominal set point.

The virtual devices used for the test may be an imperfect approximation of the real devices. This presents no issue if the approximation is good enough for the application to function normally. In this particular case the actual control loop is not part of the test, which strongly relaxes the requirements on the cavity model. The control loop implementation and the cavity model can be tuned to each other to minimise the effort. Faults (like quenches of super-conducting cavities) don't need to be properly simulated, as long as there is no sensitivity in the tested software to those details. Simply switching off the measured signal might be enough to simulate such condition.

USE VIRTUAL TIME TO AVOID RACE CONDITIONS

The example test routine shown in Figure 2 uses a system time-based sleep function to wait until the rampup procedure is completed. This approach severly suffers from potential race conditions: A fault shall be injected at a particular point of the rampup procedure. The test routine is running asynchronously to the server in a separate thread. To inject the fault in the right moment, the sleep time between starting the rampup procedure and the fault injection has to be tuned precisely. Otherwise, the fault may be injected in a different

[*] martin.hierholzer@desy.de

ISBN 978-3-95450-189-2

DEVELOPMENT AND CURRENT STATUS OF A CARBORNE GAMMA-RAY SURVEY SYSTEM, KURAMA-II

M. Tanigaki*, R. Okumura, K. Takamiya, N. Sato, H. Yoshino, H. Yoshinaga, Y. Kobayashi
Research Reactor Institute, Kyoto University, Kumatori, Osaka 590-0494, Japan

Abstract

A carborne gamma-ray survey system, named KURAMA (Kyoto University RAdiation MApping system), was developed as a response to the nuclear accident at TEPCO Fukushima Daiichi Nuclear Power Plant in 2011. A CompactRIO-based system KURAMA-II has been developed as the successor of KURAMA, and served for various activities on the radiation monitoring in Eastern Japan. We continue developing KURAMA-II as a tool not only for the current monitoring activities, but also for the immediate responses in nuclear incidents in future. The current status and on-going developments of KURAMA-II will be introduced along with the recent status of the east Japan.

INTRODUCTION

The magnitude-9 earthquake in Eastern Japan and the following massive tsunami caused a serious nuclear disaster of Fukushima Daiichi nuclear power plant. Serious contamination was caused by radioactive isotopes in Fukushima and surrounding prefectures. KURAMA [1] was developed to overcome the difficulties in radiation surveys and to establish air dose-rate maps during and after the present incident. KURAMA enabled operations of a large number of in-vehicle units for large-scale surveys owing to its high flexibility in the configuration of data-processing hubs or monitoring cars. KURAMA has been successfully applied to various activities in the radiation measurements and the compilation of radiation maps in Fukushima and surrounding areas.

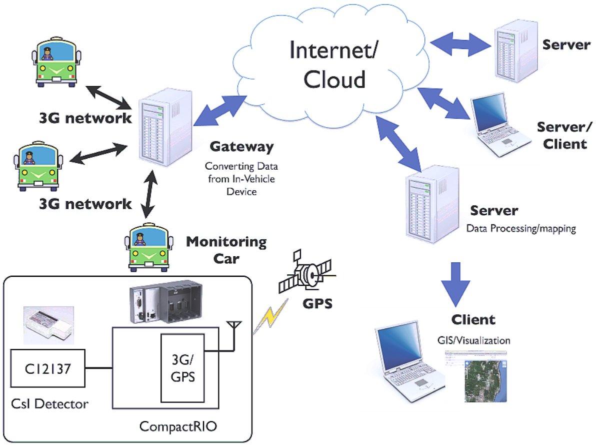


Figure 1: System outline of KURAMA-II.

As the situation becomes stabilized, the main interest in measurements moves to the tracking of the radioactive materials that have already been released into the environment surrounding the residential areas. Such monitorings can be realized efficiently if vehicles that periodically move around the residential areas, such as local buses, delivery vans or postal motorcycles, have compact and full-automated KURAMAs onboard. KURAMA-II [2] is designed for such purposes, characterized by its compactness, autonomous operation, and additional functions such as the measurement of pulse height spectrum. In this paper, the system outline of KURAMA-II as well as the results of continuous monitoring using KURAMA-II will be introduced.

SYSTEM OUTLINE OF KURAMA-II

Long term (in the order of tens of years) and detailed surveillances of radiation are required in the residential areas exposed to the radioactive materials by the nuclear accident. Such monitoring can be realized by implementing radiation monitoring units into moving vehicles in residential areas such as local buses, delivery vans or postal motorcycles. KURAMA-II is developed for such usages.

System outline of KURAMA-II is shown in Fig. 1. KURAMA-II stands on the architecture of KURAMA, but the in-vehicle part is totally re-designed for the autonomous, continuous operations in vehicles [3]. The platform is replaced from a conventional laptop PC to a CompactRIO controller of National Instruments to obtain better toughness, stability and compactness. The radiation detection part is replaced from the conventional NaI survey meter to a Hamamatsu C12137 [4], a CsI detector characterized as its compactness, high efficiency, direct ADC output and USB bus power operation. The mobile network and GPS functions are handled by a Gxxx module for CompactRIO by

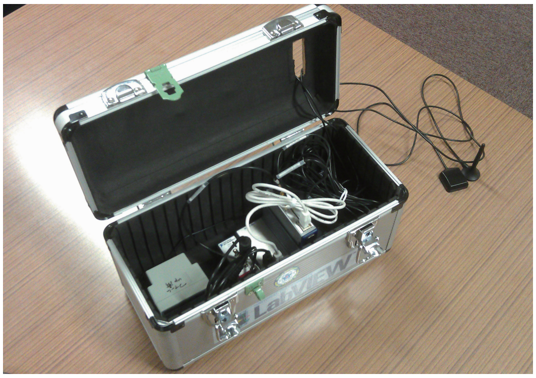


Figure 2: In-vehicle unit of KURAMA-II. A CsI detector and a CompactRIO controller are compactly placed in a tool box with the size of 34.5 cm ×17.5 cm ×19.5 cm.

[*] e-mail:tanigaki@rri.kyoto-u.ac.jp

ISBN 978-3-95450-189-2

AUGMENTED USER INTERACTION

R. Bacher, DESY, Hamburg, Germany

Abstract

The advent of advanced mobile, gaming and augmented reality devices provides users with novel interaction modalities. Speech, finger and hand gesture recognition or even gaze detection are commonly used technologies, often enriched with data from embedded gyroscope-like motion sensors. This paper discusses potential use cases of those technologies in the field of accelerator controls and maintenance. It describes the conceptual design of an intuitive, single-user, multi-modal human-machine interface which seamlessly incorporates actions based on various modalities in a single API. It discusses the present implementation status of this interface (Web2cHMI) within the Web2cToolkit framework. Finally, an outlook to future developments and ideas is presented.

MOTIVATION

Zooming applications by performing a pinch gesture at touch-sensitive displays, talking with the personal assistant to retrieve information from the internet, or controlling video games through a gaming console recognizing arm and body motions are all popular and intuitive interface features currently in common use. These technologies, well known in the consumer market, have extremely enriched the way in which people interact with their devices. Even in the rather conservative market of industrial applications, novel interaction technologies are gaining in importance, e.g. to simplify quality assurance of manufacturing processes in the car industry or to improve the efficiency of warehouse logistics.

In addition, a novel concept known as “App” and optimized for these unique technological features has been introduced which has revolutionised the conceptual design, look-and-feel and handiness of graphical user applications.

Hardware commissioning and maintenance use cases might profit from such novel interaction capabilities (modalities). For instance the alignment of mirrors mounted on an optical table to adjust a laser beam spot often requires a “third hand”. Interacting with control applications via spoken commands could be an appropriate alternative. Likewise wearing rough and dirty working gloves during cooling water maintenance work is not adequate for touch sensitive devices. Interacting via hand or arm gestures might be a better choice. Accessing on-line documentation is often indispensable for efficient inspection work. Wearing see-through augmented reality glasses controlled by head movements displaying routing schemes alongside with control applications could substantially improve measurement operations in the field.

Even control room work provides use cases for novel interaction modalities. Remote-controlling an overhead mounted screen showing some overview or control application panels might be considerably simplified by recognizing spatial gestures such as clenching a fist or snapping fingers. Beam steering requires uninterrupted eye contact with trend charts or other display updates. Controlling a virtual knob by recognizing hand rotation could eliminate the risk of losing device focus which often happens with mouse-based operations.

Today’s youth are more than familiar with these novel interaction capabilities and app-like user applications. Providing up-to-date tools for future control room operators and maintenance technicians appears to be a must.

A PARADIGM CHANGE

Today’s users of accelerator control applications have developed intuitions based on click-like interactions. In an accelerator control room the mouse is still the standard user input device to interact with graphical applications. Being well accepted by the operators it provides a very accurate pointing capability and standardized user actions normally associated with graphical widgets. Mouse-based interactions are highly reliable unambiguous single-user actions. They are best suited for complex applications containing a wealth of graphical widgets.

Thus the introduction of any new interaction capabilities will be accompanied by a serious paradigm change regarding how software programmers design graphical operations and maintenance applications and how operators or maintenance personnel interact with them.

Gesture-Based Interaction

Spatial hand- and arm gestures provide only a rough pointing capability, and experience shows that the user’s arm tends to fatigue quickly, a phenomenon known as “gorilla arm”. In addition head gestures such as turning or nodding might also be considered.

In practice only a very limited number of gesture types are available which are partially standardized and not a priori associated with graphical widgets. Consequently the design and look-and-feel of applications must accommodate these restrictions. A multi-page application design consistent with the app concept, where each page provides a well-confined and standardized functionality with an unambiguous gesture-to-action mapping, appears to be best suited.

In general hand- and arm gestures are less reliable and require a specific arming / disarming procedure to prevent the user from unwanted interaction.

Spatial gestures are not limited to single-user interaction only. It depends on the technology of the gesture recognition device used how many gestures from different persons can be tracked individually. Devices with embedded infrared stereo cameras, multi-axis gyro

ISBN 978-3-95450-189-2

A CYTHON INTERFACE TO EPICS CHANNEL ACCESS FOR HIGH-LEVEL PYTHON APPLICATIONS

J. Chrin, Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

Abstract

Through the capabilities of Cython (a Python-like programming language with the performance of C/C++), a Pythonic interface to an in-house C++ Channel Access (CA) library, CAFE, has been developed, thereby exposing CAFE's numerous multifaceted and user-friendly methods to Python application developers. A number of particularities of the PyCafe extension module are revealed. These include support for (i) memoryview and other data types that implement the new Python buffer protocol (allowing data to be shared without copying), (ii) native thread parallelism, and (iii) pointers to callback functions from wherein CAFE methods may be effortlessly executed in asynchronous interactions. A significant performance improvement is achieved when compared with conventional Pythonic CA libraries. The PyCafe interface has been realized within the context of high-level application development at SwissFEL, Switzerland's X-ray Free-Electron Laser facility.

INTRODUCTION

The Python programming language is enjoying an increasing profile within accelerator facilities, and is, furthermore, the topic of a number of tutorials and contributions to this workshop. It is both an interpreted and object oriented programming language, with dynamic typing and binding. Its straight-forward syntax is advocated for promoting conciseness and readability, thus enhancing rapid code development. The availability of third-party computational modules such as NumPy [1] and SciPy [2], coupled with modules for the visualization of multi-dimensional data [3], add to its appeal as a platform for beam dynamics applications at SwissFEL, Switzerland's X-ray Free-Electron Laser [4], and elsewhere, e.g., [5]. The very nature of Python's dynamic semantics, however, inevitably results in an adverse effect on performance, especially with respect to low-level computation involving mathematical operations and looping constructs. Faster processing time may be achieved by using Python modules written largely in C/C++ as is the case for NumPy and SciPy, else by wrapping pre-existing C/C++ code or libraries into Python. The latter may be accomplished in a number of ways. The more traditional approach is to use Python's C Application Programming Interface (Python/C API), which necessitates a greater expertise in C than in Python. Other methods include the use of specialized tools such as SWIG, SIP, CFFI, the Boost.Python C++ library or the `ctypes` package from Python's own standard library. While all of these possibilities have their own specific flavour and target, the focus of this work is to explore the emerging Cython [6] technology to provide a high-performing Python interface to EPICS (Experimental Physics and Industrial Control System) [7, 8], through a well-tested, in-house, C++ Channel Access (CA) client library, CAFE [9–12]. In this way, a full complement of CAFE's multifaceted CA methods are readily made available to application developers in Python.

THE CASE FOR CYTHON

Cython is a high-level, object-oriented, and dynamic programming language, whose principle merit is to provide a Python-like style of coding while maintaining the performance level of C. Cython is equipped with the `cython` compiler that translates Cython source code into optimised C/C++ code, which in turn is compiled into a Python extension module. Performance enhancement is achieved through Cython's declaration of static C (and certain Python) type variables and with its ability to interface to C/C++ libraries, thus bypassing execution of bytecodes by the Python Virtual Machine (VM). Its primary use cases can thus be anticipated. Speed critical, CPU-bound, segments of Python code may be shifted into the Cython domain, wherein the dynamic Python runtime semantics are replaced by the static C semantics, and existing code in external C/C++ libraries may be natively accessed [13].

Cython is a fully-fledged language that is a super-set of Python, making it enticingly simple to customize, simplify or otherwise Pythonize interfaces as they are wrapped. Cython keeps abreast with developments in the Python programming language, accommodating new features as they appear, as exemplified by its support for the new Python protocol buffer and the *typed memoryview* object. Cython's ability to generate highly optimized code, when compared with those of other wrapping tools, also places it in good stead. It is thus well suited for the task of exposing the many, well-tested, methods of the C++ CAFE library to Python.

THE PyCafe PYTHON MODULE

The PyCafe extension module exposes the Cython interface to the CAFE C++ library (CyCafe). CAFE provides a concise, complete, and clean interface with minimal details of the low-level CA implementation propagating to the user, and an abstract layer tailored for beam dynamics applications. The underlying code base ensures that CAFE clients are entirely decoupled from the CA servers, allowing, e.g., data aggregator daemons and Graphical User Interfaces (GUIs) to function in every eventuality, i.e., irrespective of changes to the connection state of the EPICS channel. CAFE's multifaceted interface further provides the flexibility that eases CAFE's use as CA host to C/C++ based scripting and domain-specific languages [11, 12]. In CyCafe, for instance, methods

ISBN 978-3-95450-189-2

RECENT BEAMLINE INTERLOCK SYSTEM AND STARS AT THE PHOTON FACTORY

T. Kosuge, H. Nitani, H. Ishii, Y. Saito, Y. Nagatani,
High Energy Accelerator Research Organization, Tsukuba, Ibaraki, Japan

Abstract

More than 20 beamlines are installed at the Photon Factory for synchrotron radiation research. Each beamline is equipped with an interlock system to protect users from radiation hazards and avoid vacuum-related troubles. The system is controlled by a programmable logic controller (PLC). Currently, touch panels and their corresponding communication protocols comprise the user interface. They work satisfactorily but are expensive.

We have developed a new type of beamline interlock system that has a PC user interface based on a simple transmission and retrieval system (STARS). We will describe the details of this new system as well as STARS.

BEAMLINE INTERLOCK SYSTEM

An interlock system is installed on every beamline at the Photon Factory in order to protect the users from radiation hazards, maintain a vacuum environment for the beamline, and protect the beamline components from high heat load damage.

Every beamline interlock system works with a programmable logic controller (PLC). The following three types of interlock systems run at the Photon Factory:

- Legacy system with an LED panel
- Field bus and touch panel user interface
- Newly developed field bus and PC-based user interface

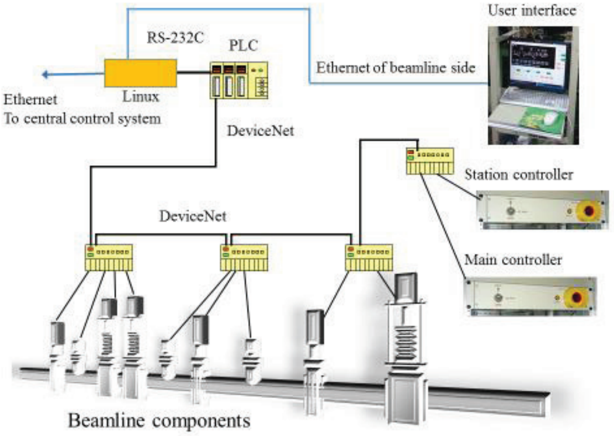


Figure 1: Beamline interlock system with field bus and PC-based user interface.

In 2015, a prototype of the beamline interlock system with PC user interface was installed onto a beamline named BL-19. We checked the reliability and effectiveness of this system and after fixing a few problems, we succeeded in the development of a new and satisfactory beamline interlock system as shown in Figure 1. The PC-based user interface of the beamline interlock system is provided with a simple transmission and retrieval system (STARS) and it is flexible.

All the legacy systems and touch panel user interfaces will be replaced with the newly developed PC-based user interface in the future. The replacement will be done annually.

STARS

STARS [1, 2] is an extremely simple message transferring software for small-scale control systems. It is installed in various systems at the Photon Factory such as the beamline control system, room access control system, and key handling system.

STARS Server and Clients

STARS consists of a server program (STARS server) and client programs (STARS clients). The server is written in Perl and works on various operating systems (Linux, Windows, Macintosh, etc.). It is a small program that can run on low power computers such as embedded Linux. STARS provides flexible system design capabilities to a developer.

STARS clients are connected to the STARS server using a TCP/IP socket, and they communicate with other clients through the STARS server using text-based messages. STARS users can develop a control system in their preferred programming language on various operating systems if they can handle the TCP/IP socket and text. Addition a new function to the system is possible by making a new STARS client and connecting it to the STARS server. STARS clients can be connected or disconnected without stoppage of the system.

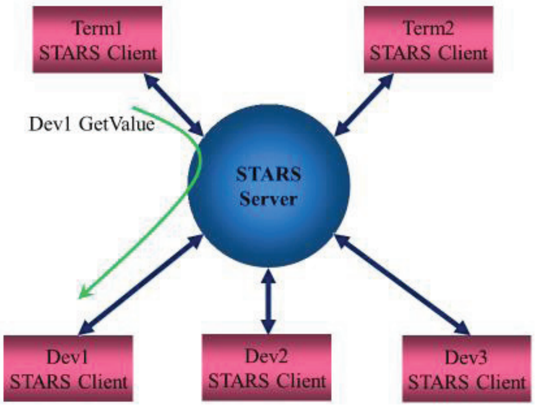


Figure 2: Example of message transferring on STARS.

ISBN 978-3-95450-189-2

PERSONAL SAFETY SYSTEM IN SESAME

A. Abbadi, A. Hamad, I. Saleh, SESAME, Allan, Jordan

Abstract

SESAME (Synchrotron-light for Experimental Science and Applications in the Middle East) is a third generation synchrotron light source under construction in Allan, Jordan. The personal safety system PSS aims to protect the personnel from radiation hazards coming from accelerator's operation by controlling the access to the radiation area and to interlock the operation of accelerator's sub-systems according to the area status. Phase 1 of SESAME PSS has been installed and commissioned successfully for Microtron and Booster tunnel. Rockwell L72s safety PLC (up to SIL 3 applications) [1] has been used for Microtron, Booster and Storage Ring PSS. Phase 2 includes the Installation of the storage ring's PSS; two new PSS cabinets with remote safety IO modules have been installed for the storage ring PSS and connected to the main PSS PLC via Ethernet safety CIP. Phase 3 is the personal Safety System for SESAME day one beam-lines which is currently under construction. Many procedures and interlocks have been implemented in order to allow SESAME Booster, Storage Ring and Beam-lines personal safety to be managed in a systematic, risk based manner.

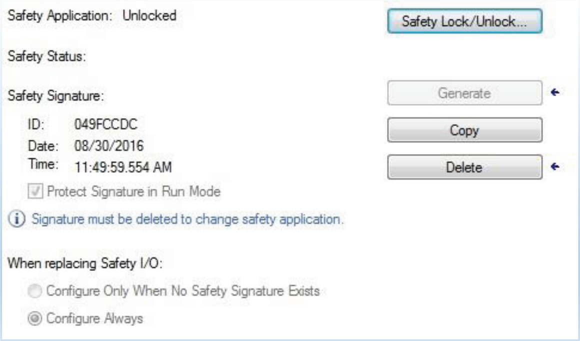
PLC BASED PERSONAL SAFETY SYSTEM

In 1998, the first part of a seven-part international standard was published to define the requirements for programmable electronic systems used in the safety related parts of controls systems. This standard is known as IEC 61508, "Functional safety of electrical/electronic/ programmable electronic safety-related systems". This seven part standard is driving the direction for future safety PLC developments [1].

The personal safety system of SESAME is a PLC based control system which provides an easy tool to implement the PSS sequences and procedures in a protected programming tool where all changes are recorded. PLC provide the benefit to control the remote safety Input and Output modules through a safety certified communication bus; this reduces cabling and reduces the risk of cabling errors.

Safety PLC has redundant microprocessors, Flash and RAM that are continuously monitored by a watchdog circuit and a synchronous detection circuit, safety PLCs have an internal 'output' circuit associated with each input for the purpose of 'exercising' the input circuitry. Inputs are driven both high and low for very short cycles during runtime to verify their functionality. Safety PLCs are suited for applications at SIL 2 and SIL 3 where they can be certified for use in most common safety applications so it may prove more cost effective to use the certified package versus taking a new control architecture through the certification process [1].

Figure 1 shows the safety PLC properties in the programming software. The controller can be Locked/Unlocked by a password, no modification is allowed on the safety logic if the controller is locked or working in running mode [1]. Safety signature should be generated after each logic modification; this signature shows the date and time of modification.


Safety Application: Unlocked

Safety Status:

Safety Signature:

ID: 049FCCDC
Date: 08/30/2016
Time: 11:49:59.554 AM

☑ Protect Signature in Run Mode

ⓘ Signature must be deleted to change safety application.

When replacing Safety I/O:

○ Configure Only When No Safety Signature Exists
● Configure Always

Figure 1: PSS safety PLC (Allen Bradley Guard -Logix L72s) properties.

SIL (SAFETY INTEGRITY LEVEL)

In comparison to similar facilities, the personal safety system at SESAME has been designed and selected to meet the requirements of SIL3, the SIL rating of a safety function is a measure of the degree of confidence in its overall ability to provide the safety function for nearly all of the time and under nearly all circumstances [2]. The most widely adopted functional safety standard is the International Electrotechnical Commission's IEC 61508 [3]. The standard divides SIL ratings into four levels, one through four, each representing an order of magnitude reduction in risk.

SYSTEM LAYOUT

Figure 2 shows the layout of personal safety system PLCs where the safety related functions are performed. A separated PLC for machine PSS and another one for the beam-line PSS.

RECENT IMPROVEMENTS TO THE RIKEN RI BEAM FACTORY CONTROL SYSTEM

M. Komiyama[†], N. Fukunishi, A. Uchiyama, RIKEN Nishina Center, Wako, Saitama, Japan
M. Hamanaka, T. Nakamura, SHI Accelerator Service, Ltd., Shinagawa, Tokyo, Japan

Abstract

The RIKEN Radioactive Isotope Beam Factory (RIBF) is a cyclotron-based heavy-ion accelerator facility that provides the world's most intense beams of unstable nuclei for nuclear physics studies. A major part of the components of the RIBF accelerator complex is controlled by the Experimental Physics and Industrial Control System (EPICS). Here, we present recent improvements to the EPICS-based RIBF control system. First, the alarm system was improved to support stable beam delivery during a long-term experiment. We introduced the Best Ever Alarm System Toolkit (BEAST) based on the Control System Studio (CSS) platform to our control system and started to monitor the vacuum systems and magnet power supplies in order to detect hardware issues or anomalous behaviors of the accelerator components. Second, the data-logging and setting software system was renewed for nearly 900 magnet power supply units. These power supplies are controlled by several different types of controllers and the configuration of the magnet power supplies has become very complicated because of several recently performed extensions and updates of the RIBF accelerator complex. Hence, we have developed new control programs in order to simplify the recording and setting of the data relevant to the operation of all the magnet power supplies.

INTRODUCTION

The RIKEN Radioactive Isotope Beam Factory (RIBF) is a cyclotron-based accelerator facility for nuclear science investigations. It consists of two heavy-ion linac injectors and five heavy-ion cyclotrons, one of which is the world's first superconducting ring cyclotron (SRC). Several acceleration modes can be achieved by changing the combination of accelerators used, and one of them is chosen according to the beam condition required by users [1]. A major part of the components of the RIBF accelerator complex, such as magnet power supplies, beam diagnostic devices, and vacuum systems is controlled by the Experimental Physics and Industrial Control System (EPICS) [2]. On the other hand, the radio frequency (RF) systems used in the RIBF accelerator complex are controlled by different dedicated systems [3]. However, all the essential operation data sets of the EPICS and other control systems are integrated into the EPICS-based control system. In addition, two types of interlock systems, which are independent of the control system, are constructed in the RIBF facility. One is a radiation safety interlock system for human protection [4], and the other is a beam interlock system (BIS) that protects the hardware of the RIBF accelerator complex from unacceptable beam losses caused by the mis-operation of high-power heavy-ion beams [5]. Figure 1 shows an overview of the RIBF control system.

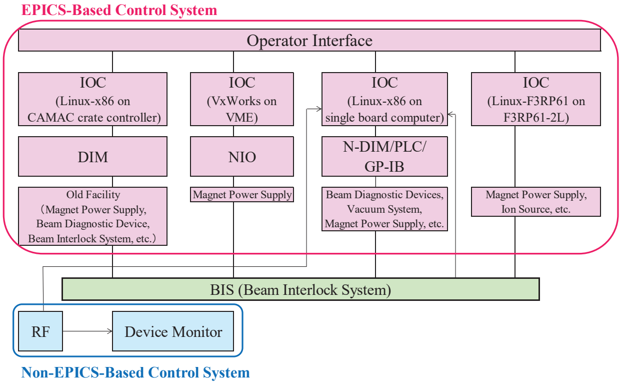

EPICS-Based Control System
Operator Interface
IOC (Linux-x86 on CAMAC crate controller)
IOC (VxWorks on VME)
IOC (Linux-x86 on single board computer)
IOC (Linux-F3RP61 on F3RP61-2L)
DIM
NIO
N-DIM/PLC/ GP-IB
Old Facility (Magnet Power Supply, Beam Diagnostic Device, Beam Interlock System, etc.)
Magnet Power Supply
Beam Diagnostic Devices, Vacuum System, Magnet Power Supply, etc.
Magnet Power Supply, Ion Source, etc.
BIS (Beam Interlock System)
RF
Device Monitor
Non-EPICS-Based Control System

Figure 1: Overview of the RIBF control system.

IMPROVEMENT OF THE ALARM SYSTEM

At the RIBF accelerator facility, various types of heavy-ions are accelerated for various types of experiments. The most operationally difficult example is a 345-MeV/nucleon ^{238}U in which we use one injector linac and four ring cyclotrons in a cascade. Operators of the RIBF accelerator are required to control a large number of accelerators components safely during beam tuning and to maintain their fine tuning during experiments lasting for long periods (typically one month). In this case, the operators should adjust and monitor 600 magnet power supply units and monitor the status of 60 units used for systems such as vacuum pumps and gate valves of the accelerators and the beam transport lines. In addition, when a trouble occurs during an experiment, the operator is required to determine the cause of the trouble and, if possible, remove it immediately. Therefore, disorders of the components, which may cause an accident, should be detected as soon as possible, especially for high-intensity operations of the RIBF accelerator complex performed routinely owing to recent performance updates. However, the existing alarm system, the Alarm Handler of the EPICS only covers some components of the ion source. Hence, we have started to upgrade the existing alarm system.

We newly installed a distributed alarm system, the Best Ever Alarm System Toolkit (BEAST) [6] under the Control System Studio (CSS) [7] that is an Eclipse-based collection of tools to monitor and operate large scale control systems. In the RIBF control system, we have

[†] misaki@riken.jp

INTEGRATION OF STANDALONE CONTROL SYSTEMS INTO EPICS-BASED SYSTEM AT RIKEN RIBF

A. Uchiyama[†], M. Komiyama, M. Fujimaki, N. Fukunishi
RIKEN Nishina Center, Wako, Saitama, Japan

Abstract

The RIKEN RI Beam Factory (RIBF) was constructed as an extension of our old accelerator facility, the RIKEN Accelerator Research Facility (RARF). The major part of the accelerator components newly developed have been integrated into the Experimental Physics and Industrial Control System (EPICS), but several old standalone control systems were carried over and some new components adopted their own standalone control systems. These non-integrated systems are grouped into two major categories. The first is hard-wired control systems and the other is based on a two-layer remote-control system without a middle layer. From the view point of efficient accelerator operation, the entire control system should be integrated. For this reason, we have replaced hard-wired devices with EPICS-compatible devices, namely, the N-DIM (originally designed by the Nishina Center), and the FA-M3 (Yokogawa Electric Corporation, Tokyo, Japan) controllers. Additionally, to access data in a two-layer system from EPICS, we have introduced a MySQL-based system as the middle layer, and have developed a feature to connect the database through the CA protocol. Thus, we could successfully integrate the system, and it is now possible to acquire all the data through EPICS.

INTRODUCTION

The RIKEN RI Beam Factory (RIBF) accelerator facility consists of five cyclotrons, including a superconducting ring cyclotron, and two linear accelerators, which act as injectors [1]. For the RIBF, we constructed a distributed control system based on EPICS for almost of the accelerator components, including the magnet power supplies, beam diagnostic instruments, and vacuum control systems [2]. On the other hand, some of the accelerator components are controlled by non-EPICS-based small control systems because the RIBF project is also utilized some components of the RIKEN Accelerator Research Facility (RARF) [3], constructed as part of a previous project. In addition, two-layer control systems, which consist of controllers and client PCs, have also been adopted since the start of the RIBF project because they require only a relatively short development time.

From the viewpoint of accelerator operation, the control systems should be integrated because the operational efficiency of the accelerator depends on the interfaces of the control systems being unified. However, due to manpower and time constraints, it would be difficult to integrate all the non-EPICS-based system in a single step. In this study, we report the methods that could be adopted to introduce EPICS-compatible controllers, as well as the data integration methods for non-EPICS-based small control systems.

STANDALONE SYSTEMS

Before the present study, the non-integrated systems remained are the control system for the 18-GHz electron cyclotron resonance ion source (18-GHz ECRIS) [4], that for the hyper electron cyclotron resonance ion source (Hyper ECRIS) [5], the beam Faraday cup control in RILAC [6], the radio-frequency (RF) systems, the temperature-measuring system, as well as several two-layer systems. We examined a mean of integration into EPICS suitable for each component one by one. Cross-operations are required for some of these standalone control systems because of management requirements.

INTEGRATION METHODS

18-GHz ECRIS

Hard-wired controllers had been used for the 18-GHz ECRIS control system since its commissioning in 1995. These old controllers should be modernized as soon as possible because the 18-GHz ECRIS is used very actively for both RIBF and standalone experiments in RILAC, the latter including superheavy element search experiments. We decided to use a Yokogawa F3RP61-2L programmable controller, in which EPICS is embedded [7], because this configuration has already been implemented in the RIBF control system. The control system for the 18-GHz ECRIS was completed in 2015 [8].

Beam Faraday Cup in RILAC

In the same way as in the case of the 18-GHz ECRIS control system, the RILAC beam Faraday cup control was based on hard-wired devices. We replaced the hard-wired devices with a N-DIM [9], and an EPICS input/output controller (IOC), which consists of single-board computers running Linux x86 [10] to provide a channel access (CA) service for the N-DIMs.

Hyper ECRIS

Hyper ECRIS, developed and operated by the Center for Nuclear Study, University of Tokyo (CNS), is used to provide a variety of metal ion beams for injection in the RIBF [6]. The control system was constructed as a standalone system with a closed network, for which the main controller was a MELSEC-A series programmable logic controller (PLC). Hyper ECRIS is usually controlled by using a Microsoft Windows-based client PC, located in the ion source room. Therefore, this client PC in the ion source room had to be accessed remotely by the accelerator operators from the RIBF control room via remote

[†] a-uchi@riken.jp

ISBN 978-3-95450-189-2

AUTOMATED AVAILABILITY STATISTICS

P. Duval, M. Lomperski, H. Ehrlichmann, DESY, Hamburg, Germany
J. Bobnar, Cosylab, Ljubljana, Slovenia

Abstract

The availability of a particle accelerator or any large machine with users is not only of paramount importance but is also, at the end of the day, an oft quoted number (0 to 100%) which represents (or is taken to represent) the overall health of the facility in question. When a single number can somehow reflect on the maintenance, operation, and engineering of the machine, it is important to know how this number was obtained. In almost all cases, the officially quoted availability is generated by hand by a machine coordinator, who peruses the operation statistics over the period in question. And when humans are involved in such a calculation there might be a latent tendency to avoid the stigma of low availability. So, not only might there be scepticism at 'impossibly high' availability, but comparing quoted availability from one machine with another might turn out to be virtually meaningless.

In this paper we present a method for calculating the machine availability automatically, based on the known (and archived) machine states and the known (and archived) alarm states of the machine. Ideally this is sufficient for a completely automatic determination of the availability. This requires, however, a perfect representation of all possible machine states and a perfect representation of all possible fatal alarms (those leading to down time). As achieving perfection is ever an ongoing affair, the ability for a human to 'post-correct' the automated statistics is also described.

INTRODUCTION

A particle accelerator facility has an operations schedule (potentially 24 hours/day 7 days/week) where the facility is obligated to supply users or experiments with beam. Any unanticipated deviation from this operations schedule is regarded as non-availability. Quite naturally, machine coordinators strive to present a perfect score of 100 % availability at the weekly operations meeting. Traditionally a machine coordinator will pore over machine data, spreadsheets, logbook entries, etc. to obtain the *official* availability of the facility over the period in question.

We are motivated to generate this availability number automatically for several reasons. First and foremost, we can remove the human element entirely if the official availability is generated entirely automatically. Secondly, we can free up a significant amount of time spent by the coordinator calculating such a number by hand. Finally, we can monitor the availability on-line during operations.

REQUIRED SERVICES

The necessary ingredients to device such a system for automatically calculating machine availability over a selected time range consist of three central services. There must be a machine state server which correctly defines all possible declared states in which a facility can be in at any time. There must be a central alarm server with a clear definition of what constitutes a fatal alarm. It should also be realized that the condition of a fatal alarm is inextricably bound to the machine state, as we shall see below. There must also be a central archive server which keeps a history of the state and fatal-alarm information.

Machine State Server

The possible states of an accelerator facility are defined by the machine coordinators and the facility will be in *some* state at any given time. Theoretically the choice could be as simple as *running* or *not running*, but is generally more complicated. For completeness, the TINE [1] state server also recognizes the state *unknown* if there is no proper declared state. Otherwise the state of a machine will be declared to the state server and the machine will be assumed to be in that state until another state declaration is made. The set of all possible machine states is completely configurable.

There exists the question as to what to do about *problems*. Either *problems* is a valid machine state which is likewise declared or *problems* is an attribute assigned to one of the other defined valid machine states. For instance: “this is a declared *user run* but we have *problems*”. The TINE state server can handle either option, but we point out that the current configuration treats *problems* itself as a valid declared machine state. This implies that some service must determine that we cannot be in an operational state and officially declare the state *problems*.

Central Alarm Server

The principal ansatz concerning availability is that “if the machine is not available then there must be at least one fatal alarm in one of its subsystems.” And if we are treating *problems* as a declared state then a corollary to this ansatz is that “if we are in the *problems* state then there must be at least one fatal alarm”.

We perhaps begin to see a number of consistency checks unfolding before our eyes. If the state is *problems* and there are no fatal alarms then this is by definition wrong and needs to be investigated and fixed. Furthermore, if there is a fatal alarm then the state must

MULTIPURPOSE VACUUM CHAMBER - AUTOMATION, INTERLOCK-SYSTEM AND SELF-OPERATING VACUUM ROUTINES

R. M. Caliari*, H. G. P. Oliveira, G. L. M. P. Rodrigues, C. H. Tho,
Brazilian Synchrotron Light Laboratory, Campinas, Brazil

Abstract

The LNLS' Beamlines Engineering Group is developing new technologies for SIRIUS Beamlines. To validate and test those technologies, the group has previously worked in a Multipurpose Vacuum Chamber (MPVC). This chamber has a powerful pumping system, several viewports, feedthroughs, RGA and a cryocooler system installed. This configuration delivery a very flexible test environment and relative short pump time to achieve UHV conditions, essential to reduce the test and validation time.

This paper will detail the MPVC automation structure, presenting the state machines, interlock logic and other diagrams that exemplify the automatic routines concepts. Operations like pumping from atmospheric pressure to HV, ionic pump flash and ventilation routines are automatic.

The automated system was developed in LabVIEW, using a cRIO and Touch Panel interface. Based in Ethernet connection and published shared variables, the system has a friendly user interface and archiving for the main variables.

The system was developed in a multi user collaborative ambient. The paper will show the advantages and disadvantages we faced working with LabVIEW as multiuser development tool.

INTRODUCTION

This work presents de project of a multipurpose vacuum chamber for the beamline engineering group at LNLS, dedicated for new beamlines projects for Sirius, the new Brazilian 4th generation synchrotron [1, 2]. Monochromators [3], KB Mirrors, Front Ends [4] and many others. The first experimental results were already presented on previous conferences [5].

SYSTEM PURPOSE AND DESIGN

The Beamline Engineering group started the mechanical design on early 2015, aiming one flexible environment, with some principal characteristics:

- Ultra-high vacuum (UHV) environment (target: 10-11 mbar with backing procedure and 10-8 mbar without baking);
- Low vibration environment;
- View ports in different angles, to make it flexible to implement optical instrumentation from outside the chamber;
- Low diffraction coefficient viewport from VAb Vakuum-Anlagengau, for interferometer characterization from outside;
- Many power, signal and temperature feedthroughs;
- Cryogenic support system;
- Residual Gas Analyser;
- Complete automation operation;
- Safety system;
- Low electrical noise environment;
- IPS protection for main components.

Those characteristics were obtained as foreseen requirements for the future development projects, for the new Sirius beamlines, their prototypes and concept proves. The automated routines are the base for a pump station under development, which's mechanical sketch shall be similar to some published works [6].

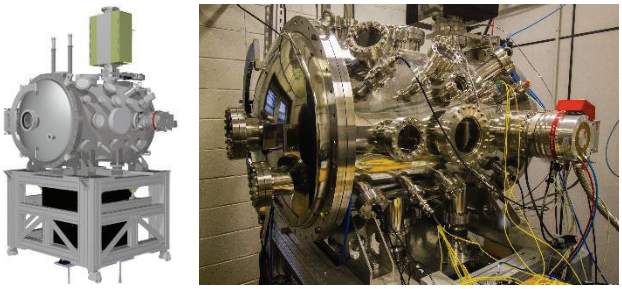


Figure 1: 3D model and actual picture of the Multipurpose Vacuum chamber, a flexible environment for the beamline engineering group.

The detailed design was made, and the 3D model can be seen in Figure 1. The group experience conducted the project for a big UHV chamber (800 mm diameter and 1000 mm depth), and simulations were made to ensure a robust low vibration environment Figure 2.

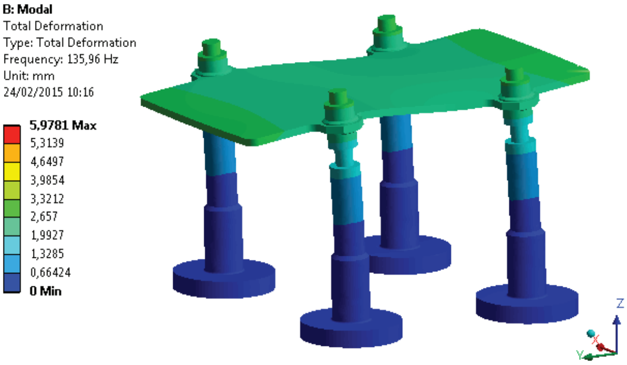


Figure 2: Modal Analyses to guaratee the low vibration environment (first vibration mode).

With the requirements and mechanical projects set, the final release, with hardware, software and interfaces solutions are presented on this text.

* ricardo.caliari@lnls.br

DEVELOPMENT OF A VIRTUAL ACCELERATOR FOR SIRIUS

X. R. Resende*, A. H. C. Mukai, L. N. P. Vilela, I. Stevani
Brazilian Synchrotron Light Laboratory (LNLS), Campinas, Brazil

Abstract

A virtual accelerator is being developed for Sirius, the new 4th generation synchrotron light source being built in Campinas, Brazil [1]. The virtual accelerator is an on-line beam simulator which is integrated into EPICS control system. It consists of a command line interface server with a channel access (CA) layer and with an in-house developed tracking code library written in C++ for efficiency gain. The purpose of such server is to facilitate early development and testing of high level applications for the control system.

INTRODUCTION

Sirius, the new storage ring at the Brazilian Synchrotron Light Laboratory (LNLS), will use EPICS as its control system. Most of the development of the high level applications (HLAs) will take place next year and will be conducted by the accelerator physics group. In the meanwhile, a few client applications have already been implemented to allow analysis of the choices in software development frameworks [2] that were made.

To be able to test and integrate the above-mentioned HLAs in the control system (CS), it was decided that a virtual accelerator (VA) with channel access server layer (CAS) for EPICS should be developed, implemented and made available as soon as possible. The idea is that having a VA allows for early development of control system software, be it for input-output controllers (IOCs) or HLAs.

The VA implements functionalities that can provide simulated process variables (PVs) on the CS that have not been made available yet, thus creating a mock-up CS environment in which early software development is possible. This test environment with VA also has the potential to speed up project development by partially parallelising implementations of applications that consume data from each other.

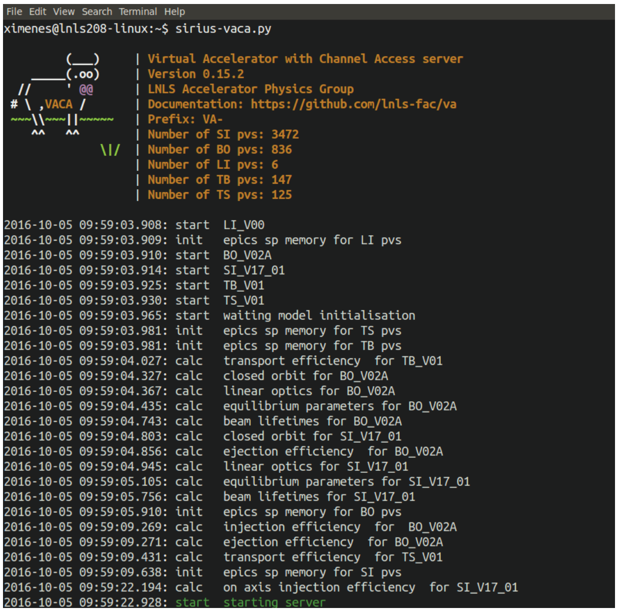
Commissioning training is also possible using HLAs in the control system provided with the virtual accelerator.

VIRTUAL ACCELERATOR

From the onset it was decided that the virtual accelerator would be composed of two parts: the first, a back-end machine application implementing a simulated virtual accelerator with a channel access server layer (VACA) and the second, a set of front-end virtual IOCs (vIOCS) with which other CS applications are supposed to interact.

Accelerator properties such as beam current and position, injection losses, power supply and RF subsystem setpoints, and so on, are simulated nominally in VACA. The virtual IOCS, on the other hand, encapsulate all PVs that represent the interface between the VA and the rest of the CS. They also add simulated fluctuations to accelerator properties and implement device-dependent parameters, such as excitation curves of the magnets or BPM calibration parameters.

The advantage of this approach is that virtual IOCs can be gradually replaced by their corresponding real IOCs when they become available, and without having to modify core simulation code, since it is implemented separately in VACA.



```
ximenes@lnls208-linux:~$ sirius-vaca.py

        (___)       | Virtual Accelerator with Channel Access server
   _____(.oo)       | Version 0.15.2
  //      ' @@      | LNLS Accelerator Physics Group
 # \ ,VACA /        | Documentation: https://github.com/lnls-fac/va
 ~~~\\~~~||~~~~~    | Prefix: VA-
    ^^   ^^         | Number of SI pvs: 3472
               \|/  | Number of BO pvs: 836
                    | Number of LI pvs: 6
                    | Number of TB pvs: 147
                    | Number of TS pvs: 125

2016-10-05 09:59:03.908: start  LI_V00
2016-10-05 09:59:03.909: init   epics sp memory for LI pvs
2016-10-05 09:59:03.910: start  BO_V02A
2016-10-05 09:59:03.914: start  SI_V17_01
2016-10-05 09:59:03.925: start  TB_V01
2016-10-05 09:59:03.930: start  TS_V01
2016-10-05 09:59:03.965: start  waiting model initialisation
2016-10-05 09:59:03.981: init   epics sp memory for TS pvs
2016-10-05 09:59:03.981: init   epics sp memory for TB pvs
2016-10-05 09:59:04.027: calc   transport efficiency  for TB_V01
2016-10-05 09:59:04.327: calc   closed orbit for BO_V02A
2016-10-05 09:59:04.367: calc   linear optics for BO_V02A
2016-10-05 09:59:04.435: calc   equilibrium parameters for BO_V02A
2016-10-05 09:59:04.743: calc   beam lifetimes for BO_V02A
2016-10-05 09:59:04.803: calc   closed orbit for SI_V17_01
2016-10-05 09:59:04.856: calc   ejection efficiency  for BO_V02A
2016-10-05 09:59:04.945: calc   linear optics for SI_V17_01
2016-10-05 09:59:05.105: calc   equilibrium parameters for SI_V17_01
2016-10-05 09:59:05.756: calc   beam lifetimes for SI_V17_01
2016-10-05 09:59:05.910: init   epics sp memory for BO pvs
2016-10-05 09:59:09.269: calc   injection efficiency  for  BO_V02A
2016-10-05 09:59:09.271: calc   ejection efficiency  for BO_V02A
2016-10-05 09:59:09.431: calc   transport efficiency  for TS_V01
2016-10-05 09:59:09.638: init   epics sp memory for SI pvs
2016-10-05 09:59:22.194: calc   on axis injection efficiency  for SI_V17_01
2016-10-05 09:59:22.928: start  starting server
```

Figure 1: Screen printout of a command line terminal showing a running instance of VACA with the display of its banner and useful information.

VACA

The virtual accelerator with channel access is written in Python3. The choice of a high level programming language allows for rapid development of the intricate functionalities the virtual accelerator has to provide. Python language works as a binding layer between the two main core modules: the CA server and the tracking simulation code. The Python package PCASPy [3] is used to write the CA server module. PVs implemented in VACA have a prefix "VA-" indicating that they represent virtual process variables. For the simulation module, a library called trackcpp [4], that has been developed at LNLS by the accelerator physics group staff, was reused. It is a C++ library of beam dynamics calculations and tracking routines closely based on, and tested against, Tracy [5] and Matlab AT [6] passmethods. This library is converted to a python package using Swig3.0 [7], thus conveying fast and optimised routines that can be conveniently called within Python.

[*] ximenes.resende@lnls.br

ISBN 978-3-95450-189-2

HIGH LEVEL APPLICATIONS FOR SIRIUS

I. Stevani[*], N. Milas, X. R. Resende, L. N. P. Vilela
Brazilian Synchrotron Light Laboratory (LNLS), Campinas, Brazil

Abstract

Has been decided that Sirius will use EPICS as its distributed control system and this year the development of its high level applications started. Three development frameworks were chosen for building these applications: CS-Studio [1], PyQt [2] and Matlab Middle Layer [3] (MML). Graphical user interfaces (GUI) and machine applications have already been designed and implemented for a few systems using CS-Studio and PyQt: slow orbit feedback, lifetime calculation and top-up injection. Specific Sirius data structures were added to the MML scripts in order to allow for EPICS communication through LabCA [4].

INTRODUCTION

Sirius is a synchrotron light source facility based on a 4th generation storage ring that is presently under construction at LNLS in Campinas, Brazil. The project initiated in 2008 with the first lattice studies and in the beginning of 2015 the building construction started. Machine commissioning is expected to start mid 2018 [5]. By this time, indispensable control systems for beam stability – such as tune measurement, orbit correction, injection – should be ready for use, and so should their high level applications (HLAs).

To achieve this goal, the accelerator physics group started developing these applications this year and should intensify its activity with HLA next year, after most of the group's work related to machine components' specifications – that is currently taking most of the group efforts – is done. A virtual accelerator with channel access server (VACA) [6] has been implemented to emulate the real machine, thus providing a testbed environment for high level software development.

APPLICATIONS FRAMEWORK

Figure 1 shows a schematic view of the application framework Sirius will use for HLA development. On a lower level, services are categorized in two groups: IOCs and machine applications. Both of them are channel access (CA) servers, the first providing control access to machine hardware while the second providing machine services. Machine applications developed so far were written in Python using PCASpy [7]. Virtual IOCs were implemented using EPICS Database on top of VACA in order to provide simulated fluctuations for machine parameters. EPICS V4 is another option for implementing machine applications that will be studied and discussed next year.

On top of Fig. 1 are the client applications. For basic process variable monitoring and settings with graphical user interface capabilities, CS-Studio was chosen due to its simplicity and fast learning curve. Most of the control system interface is expected to be implemented with CS-Studio. For software that requires elaborate logic, algorithms will either be implemented in the IOC level with simple driving GUI clients or using other options, such as PyQt or MML.

The control system can benefit from various support applications for storing and organizing system's parameters, IOC software, device and PV name lists, and so on. The plan is to use a few of the applications in development under the DISCS collaboration effort [8, 9]. The device naming and configuration modules of DISCS are currently being tested and used in the HLA development framework. Other DISCS modules, such as its logbook, IOC and machine save/restore services, will soon be tested as well.

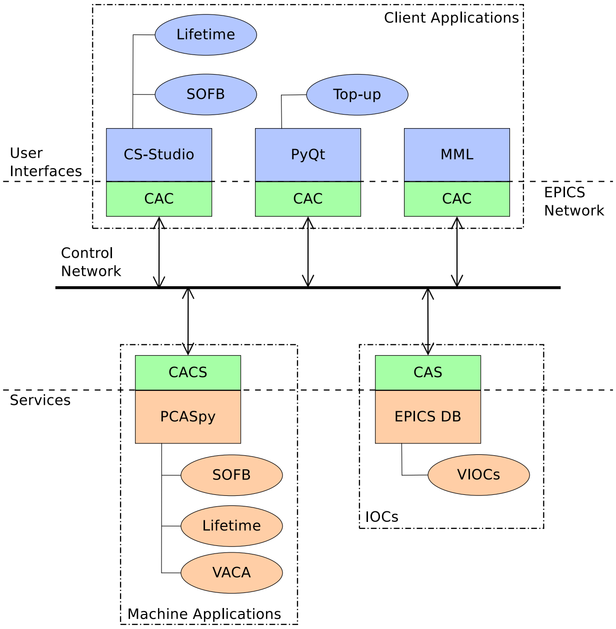


Figure 1: Applications framework schematic.

IMPLEMENTED APPLICATIONS

Some prototypes were built on these platforms listed before to help the developers familiarize with their functionalities and evaluate if they are suitable for Sirius requirements. The HLAs are presented next.

Slow Orbit Feedback

The slow orbit feedback (SOFB) system software is an example of a HLA that has already been written for Sirius. It consists of two modules: a machine application that reads BPMs and controls orbit correctors and a user graphical interface application.

The first one is responsible for all data processing and control, such as orbit/response matrix measurements and

[*] isabella.stevani@lnls.br

ISBN 978-3-95450-189-2

BEAMLINE SUPERVISORY SYSTEM USING A LOW-COST SINGLE-BOARD COMPUTER

G. T. Semissatto[†], F. H. Cardoso, H. F. Canova, J. Souza
LNLS, Campinas, Brazil

Abstract

Sirius is the new accelerator facility, under construction at the LNLS (Brazilian National Synchrotron Light Laboratory) site, in Campinas, São Paulo. The new machine is a 3 GeV, low emittance storage ring designed to accommodate up to 40 experimental stations. During beamline operation, supervisory systems are an important tool to provide information about machine status and beamline operation modes for the beamline´s users. A modern TV based broadcast system was developed to meet this application, using low-cost single board computers with an interface to EPS/PPS system. The details about hardware, software configuration, user's requirements as well suggestions on further improvements, will be presented.

INTRODUCTION

At UVX, the current machine at LNLS, the machine status broadcasting system is based on antiquated cable television topology. Furthermore, beamline status is monitored only using audible sounds (at hutch armed status) and light indicators. This system has become obsolete and is no longer manageable, with an additional drawback regarding the support for people with color vision deficiency (such as Daltonism). A new and modern supervisory system is envisioned for Sirius, to provide readable warnings, useful information and customized sounds for beamline users.

A broadcast system, based on web browsers and LED TV´s is proposed. The system displays the machine status (such as storage ring current, beam lifetime, machine energy, top-up) and beamline operations modes (photon beam status: beam on, beam off, imminent beam, hutch armed, etc).

The supervisory system was developed using a low-cost single board computer, the Beaglebone Black [1], integrated with the Personnel Protection System (PPS) and Equipment Protection System (EPS), both based on industrial programmable logic controllers (PLCs). Beaglebone Black (see Figure 1) is a powerful low-cost single board computer equipped with a Sitara ARM Cortex-A8 processor running at 1 GHz. The board provides 512 MB of RAM, two 46-pin expansion connectors (with much general-purpose I/O pins), on-chip Ethernet, a microSD slot, a USB host port and HDMI video interface. The board also runs many distributions of embedded Linux providing support for a variety of programming languages.

By means of Python scripts and web browsers, the system was designed for low cost, high flexibility and expandability. This also allows for a efficient way to create fully customized warning messages for users.

BEAMLINE SUPERVISORY SYSTEM

System overview

During normal operation, the system provides machine status (storage ring parameters) using a web-based application. The operation group, based in the storage ring control room, can configure this information. Besides the storage ring information, the beamline´s users have the need to know the photon beam status, the beamline operation mode and any other information required for that specific experimental station, such as lasers, high pressure cells, robots, etc.

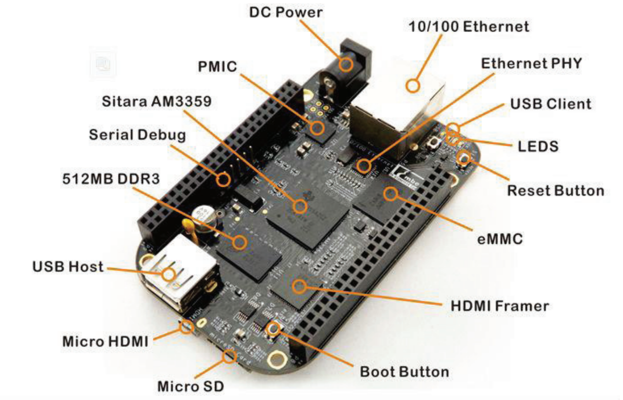


Figure 1: Beaglebone Black.

The beamline supervisory system flowchart is shown below (see Figure 2). The PPS is an engineered interlock system that monitors the various devices installed in the beamline, for personnel safety and provide emergency beam shutdown. PPS is based on PLC's that monitor the positions of the front-end shutters, beamline shutters and beamline hutch doors. This system also defines all the safety functions to avoid any accidental exposure to radiation inside the beamline hutch.

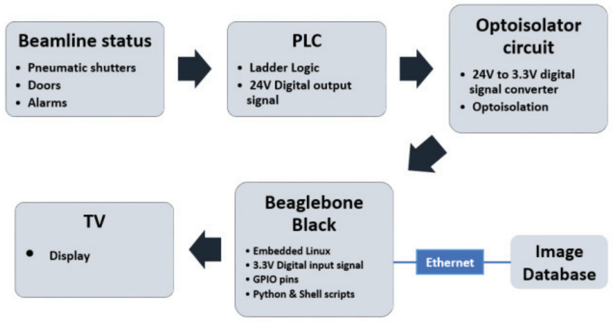


Figure 2: Supervisory system stages.

[†] guilherme.semissatto@lnls.br

VDE - VIRTUAL DOCUMENTATION ENVIRONMENT

H. F. Canova*, D. H. C. Araujo, M. P. Donadio, R. R. Geraldes
LNLS, Campinas, Brazil

Abstract

At LNLS hundreds of motors are used at the beamlines to move parts, equipment or full systems, according to different profile, synchronization and accuracy requirements. Historically, the documentation of motion axes of the LNLS beamlines was either done only at the moment of their installation and commissioning, or not properly done at all. Thus, after some time, keeping track of changes and performing maintenance could turn out to be very challenging, and there was the clear need of some solution to ensure that every change in motors would be reflected in their documentation. In 2012 the migration of the beamlines control system to the EPICS (Experimental Physics and Industrial Control System) [1] platform pushed the development of a new documentation system. In a first version, it consisted of a smart spreadsheet that generated the EPICS configuration files automatically. Later the spreadsheet evolved to a web-based system the VDE - Virtual Documentation Environment, which allows the beamlines staff to change the motion axis parameters without the need of a deep knowledge about EPICS and ensures the complete motion axis documentation intuitively. Also, changes in motors will not work in EPICS if the documentation is not updated, guaranteeing the link between documentation and the real system.

INTRODUCTION

Currently, around 700 motorized mechanisms are installed at LNLS beamlines, being used, for instance, to perform sophisticated optical alignment of mirrors and monochromators and position samples. These motorized mechanisms can be abstracted as motion axes, which are composed of motors (mostly stepper motors), gear boxes, transmission elements and encoders, when necessary. All these axes are integrated into the EPICS distributed control system of the beamlines and endstations.

The most common motion controllers at the LNLS beamlines are Galil [2], Parker [3] and IMS [4] devices whose EPICS IOCs (Input/Output Controller) run in a dedicated National Instruments PXI chassis [5], for the main beamline components, or in a Virtual Machine, in the case of separated endstations.

DPM

In 2012, with the need to document the motorized axes of the beamlines, until then based only on mechanical drawings, and to facilitate the software configurations, by providing all the relevant mechanical parameters, a standard configuration tool, the so called DPM, has been implemented. It was based on two spreadsheets, one being used as a library and the other, the master spreadsheet for the beamline. In the first, there were records of models and manufacturers, for motors, drivers and gearboxes, for instance, whereas the latter concentrated the information of each axis of the installation in a line on the spreadsheet. It also had the necessary information to perform all the needed calculations and determine the primary parameters of the axes as speed and resolution, for example. After all the axes were correctly registered in the master spreadsheet, the DPM was able to generate the necessary configuration files for the EPICS IOCs using VBA macros in the spreadsheet. Finally, all the configuration files were stored in text format and a LabVIEW application was used to transfer the file to the IOC server by FTP.

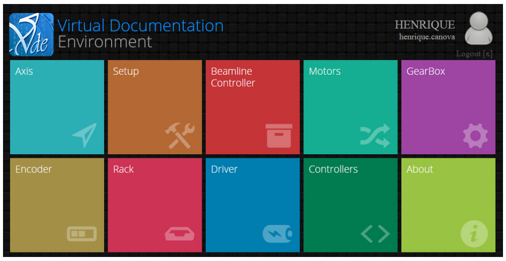


Figure 1: VDE home screen.

VDE

With the need to make a more robust tool, in order to avoid errors during the operation, and to add more features, the VDE was created. Allowing for greater versatility a web-based platform running on an Apache server, and a relational MySQL database is used. The main language of the system, to access the database and to manipulate the data, is PHP/JavaScript, whereas the pages designs are based on HTML5 and CSS3.

The VDE consists of a series of different web pages, each one giving access to each part of a motor axis and its respective entries. The home screen (Fig. 1) shows all the available options in the platform, with even though some of them are for the administrators only.

The system is integrated with the LNLS Active Directory System to provide a way for the user to login using the institutional username and password from any computer in the campus. Thereby, this enables the VDE to set different access levels to restrict access to certain sets of motors for the desired users. For example the home screen, viewing access to the registered common equipment is allowed for general user, however, only admins can edit or register new items. Naturally the administrators also have access to all axes in the database.

[*] henrique.canova@lnls.br

ISBN 978-3-95450-189-2

USING TKINTER OF PYTHON TO CREATE GRAPHICAL USER INTERFACE (GUI) FOR SCRIPTS IN LNLS

D. B. Beniz[†], A. M. Espindola[‡], Brazilian Synchrotron Light Laboratory, Campinas, Brazil

Abstract

Python is being widely used to create scripts which cover different necessities in computational scenario. At LNLS (Brazilian Synchrotron Light Laboratory) we successfully developed Python scripts to control beamlines operations, including a case of GUI (*Graphical User Interface*) creation using Tkinter [1] for one of LNLS beamlines, DXAS (*Dispersive X-ray Absorption Spectroscopy*) [2]. In this article its motivation and some implementation details will be presented.

MOTIVATION

The decision to use Python to build a GUI was based on the previous experience with such programming language in LNLS. There is a library package, called Py4Syn [3], developed in LNLS with Python version 3.4 and in use to control beamline devices, like motors and detectors, and to operate a sequence of actions to perform specific experiments by synchronization of a set of such devices and storing of collected data into files formatted in columns to facilitate their analysis. Controllers of such devices have software abstractions, IOCs (*Input/Output Controllers*), developed in EPICS (*Experimental Physics and Industrial Control System*) [4] which resources, PVs (*Process Variables*), are available in the laboratory network via CA (*Channel Access*) [5] protocol. Python offers packages to control such resources, with PyEPICS [6], to perform mathematical calculations and data matrix manipulation, with NumPy [7], and to display data as graphics, with Matplotlib [8], this way it facilitated the development of Py4Syn.

Once we had the tool to elaborate scripts to orchestrate synchrotron beamline experiments, Py4Syn, the new challenge was to offer a GUI that helped users to inform scripts parameters, control and monitor their execution. LNLS adopted CS (Control System) Studio [9] as the tool to monitor and operate EPICS IOCs. It is a great option to monitor and interact with EPICS PVs, however, it is not recommended by their developers to control complex scripts in Python. We tried to use CS Studio to control Py4Syn scripts but the performance was unsatisfactory. Then, we decided to build a GUI in Python, once the scripts were written using that language, and Tkinter arose as a good start as it is the standard GUI package of Python and we found a large number of tutorials and code examples in the Internet. The first experience in LNLS with Tkinter to build graphical interfaces for Python scripts was with DXAS beamline, which was being reformed between 2014 and 2015.

ARCHITECTURE OVERVIEW

In Figure 1 the overview of current software architecture solution for control system of DXAS is presented.

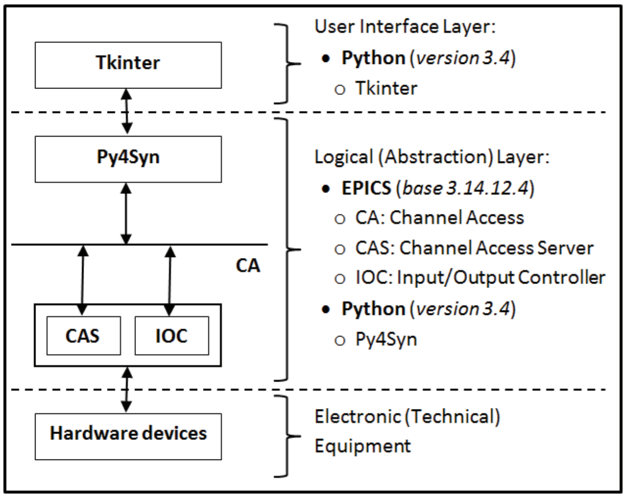


Figure 1: Overview of DXAS Solution Architecture.

Electronic (Technical) Equipment

At lower level of control system are the typical technical devices present in synchrotron beamlines. In fact, they have their own controllers which receive instructions, via serial (RS232/RS248) or Ethernet connection, for example, and then command the equipment. Some devices present in DXAS beamline of LNLS are:

- Galil DMC-4183: motor controller
- Parker OEM750: motor controller
- Heidenhain MT 2501: optical encoder
- Keithley 6485: picoammeter
- Kepco BOP: power supplier
- OMRON E5CK: digital controller of furnace
- LakeShore 331: temperature controller of cryogenic cooling system
- Stanford SR570: low noise current preamplifier
- Princeton Instruments PyLoN: CCD camera

Logical (Abstraction) Layer

Over the devices controllers is the first abstraction of them, build in EPICS, with correspondent IOCs for each one of the devices. Those devices of the same manufacturer and model share the same IOC program, but run in individual instances. Basically, a set of instructions to get information or to send a command to devices is organized in those IOCs as PVs, where each PV is a record, or piece of data, with some attributes to format, configure or simply return related information.

[†] douglas.beniz@lnls.br
[‡] alexey.espindola@lnls.br

DEVELOPMENTS OF THE 'CERBERUS' LASER INTERLOCK AND HAZARD DISPLAY SYSTEM AND ASSOCIATED DESIGN TOOL

D. A. Pepler[†], R. Bickerton, A. Tylee, STFC Rutherford Appleton Laboratory, Didcot, UK

Abstract

Following on from the successful implementation of 'Cerberus' a comprehensive laser interlock / control and hazard display system [1, 2] on the Vulcan High-Power Laser at the Central Laser Facility (CLF) [3], the last few years have seen the safety system become a CLF standard and its use extended to many different laser systems and laboratories within the department.

This paper will provide an overview of the system, its most recent enhancements and in particular the developments of a design tool and the potential for this system to be used in other fields.

INTRODUCTION

Vulcan is a large, complex and Petawatt class (10^{15} W) neodymium-doped glass laser system. It has a footprint of two Olympic-sized swimming pools with some 250 m of beamlines from its initial suite of optical pulse sources through to a target. The short timescale (10^{-12} to 10^{-9} s) laser pulses pass through many optical components that are situated in a number of rooms which can be kept isolated (and which are then effectively controlled autonomously) or coupled together with remotely-controlled and pneumatically driven shutters that link two or more rooms together. All entry and exit points, laser enclosures and sources and the interconnecting shutters are part of the interlock system. Any potential increase in the level of hazard (e.g. enabling a laser source) requires simultaneous human operator authorisation key action from all affected areas. Reducing the level of hazard, for example by closing a shutter, can be achieved at any time with a single touch. Unauthorised entry or detection of a fault condition will automatically shut the affected areas down rendering them safe by disabling all local hazards and closing any links to other areas.

SYSTEM DESCRIPTION

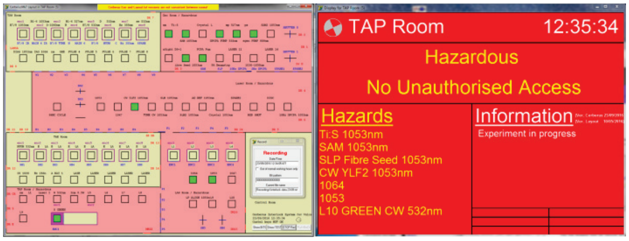
In Greek mythology Cerberus is generally depicted as a three-headed guard dog and the laser interlock and control system was partially given that name because of its guarding nature and its threefold constituent parts – a control Programmable Logic Controller (PLC), a safety PLC and a Hazard Information Display system.

The control PLC provides the operator with multi-page touchscreen display that allows for the indication and control of various items such as laser hazard, shutters and beam-stops.

The safety PLC conforms to the international standard IEC 61508 [4] at Safety Integrity Level 2 (SIL2) and provides the underlying safety by constantly monitoring the state of the system and either permits or denies a particular action to take place on the basis of the required and present level of authorisation.

Within each room of the laser facility the Hazard Information Display comprises of a local PC with a dual monitor. At a glance, the first monitor gives operators an overview of the safety state of the whole system and the second monitor shows the specific laser hazards present within the room. Screenshots from these are shown in Fig. 1 below.

The overview screen is located at a rooms central control point whereas the hazard display is also duplicated in multiple locations around the room. Both screens are also duplicated at the entry point with the laser hazard display having a clear green or red colouration to denote whether the system is safe (no laser hazards) or not. For Vulcan, laser hazards would commonly be continuous or pulsed invisible near-IR lasers operating at 1053 nm and which if present require special protective eyewear to be worn.



TAP Room 12:35:34

Hazardous

No Unauthorised Access

Hazards

Ti:S 1053nm
SAM 1053nm
SLP Fibre Seed 1053nm
CW YLF2 1053nm
1064
1053
L10 GREEN CW 532nm

Information

Experiment in progress

Figure 1: Images from the Hazard Information Displays showing left) the system overview screen, and right) the individual area's laser hazard warning display.

Apart from the display function, the Cerberus application also has the ability to record all activity and changes within the interlock system in a text-based log file. This provides a way of checking for unexpected anomalies or out-of-normal-working-hours activity. This log can also be played back in real or enhanced time.

SYSTEM DEVELOPMENTS

Because all laser hazards are enabled through Cerberus controls and normally would turn the hazard display red demanding that goggles be worn, the inclusion of newer visible (532 nm) alignment lasers that are predominately eye-safe necessitated a separate indication. In the case where *only* eye-safe lasers are in use a new mode was implemented to complete the 'traffic light' warning notification. This is demonstrated in Fig. 2 where only a 532 nm laser is enabled and the hazard display is green apart from an orange band displaying the yellow / black flashing message 'Hazardous at Focus'.

This approach was seen as visually clearer than having the whole screen orange so as to provide good discrimination between orange and red. If a non-eye-safe laser were to added then the display would revert to the standard red 'Hazardous' indication.

[†] dave.pepler@stfc.ac.uk

ISBN 978-3-95450-189-2

OPEN HARDWARE AND COLLABORATION

J. Serrano, CERN, Geneva Switzerland

Abstract

Open Source Hardware (OSHW) follows the lead of Free and Open Source Software (FOSS) and has similar goals: ensuring developers can share their work without artificial hurdles, improving quality through peer review, avoiding vendor lock-in and providing for a fair playground in which projects can thrive and accommodate contributions without compromising their long-term future. The paper introduces OSHW and then attempts to answer a number of questions: (i) what are the perceived benefits and issues of OSHW, in general and in the context of public research facilities? (ii) what is new with respect to FOSS? (iii) what makes OSHW projects succeed or fail? The paper uses real examples of OSHW projects and practice throughout – mostly CERN-related because they are as good as any other and well known by the author – and concludes with some thoughts about what the future holds in this domain.

INTRODUCTION

What is Open Source Hardware? The word *"open"* is often abused. Some people consider it good marketing to assign this label to a product they have designed, without much concern to what is actually meant by it. Having identified this issue quite early on, the Open Source Hardware Association (OSHWA) published an official definition [1] of Open Source Hardware (OSHW) in 2010. It is inspired by the definition of Open Source Software [2] and the four freedoms of Free Software [3]. As such, it focuses on the freedoms granted to users of OSHW designs, which include the right to study the design documents, modify them, share the modifications, build hardware based on the designs, and commercialize the results. The definition is important insofar as it enables efficient communication by clarifying what OSHW actually means.

With the definition under our belts, let's move on and discuss the OSHW phenomenon. More and more designers share their design documents on the Web, using licenses which comply with the OSHW definition. Why? Because designing non-trivial hardware has become easier in the last years, opening this domain to a larger subset of the population. Things which are easy and useful to do have a special relationship with freedom. If a world power decided to forbid travel to Jupiter, few people would complain. If, on the other hand, Spanish citizens were suddenly forbidden to cross the French border, there would understandably be strong protests. This is why it is important to guarantee that people who want to share their designs can do so within a robust legal framework which guarantees a number of important freedoms.

As we will see throughout this article, much of what is happening in OSHW and many of the issues we confront in its practice have a direct counterpart in the Free and Open Source Software (FOSS) world. The democratization of access to computing in the late 80's and early 90's explains to a large extent the emergence of Free Software. We see here again the need to provide a solid conceptual and legal setting for a freedom when it becomes easy to exercise by many. FOSS is a reference and a source of inspiration for many OSHW practitioners.

There are many examples to illustrate the revolution in empowerment that has happened in hardware design in the last few years. Consider the design of a GPS watch. It used to require the efforts and talents of a small or medium-size enterprise. Today it can be done by a few friends staying after work over a few months [4], even if they restrict themselves to using only FOSS tools in the development. The availability of simple, powerful, ready-made, modular electronic boards has enabled an even bigger community of integrators. They use hardware from companies such as Adafruit, Sparkfun and Arduino – which have themselves grown to a considerable size – and add simple customizations to generate a variety of innovative products.

For reasons of legitimacy and focus, this article deals mostly with the particular case of electronics development in the framework of big public scientific institutions, but most of its reasoning and conclusions should apply directly elsewhere. The existence of very successful commercial and non-commercial OSHW organizations and projects tells us that this paradigm is useful. We try to see what contexts favor these developments and take a critical look at perceived advantages and disadvantages of OSHW.

PERCEIVED BENEFITS

As one would expect, the perceived benefits of FOSS apply quite clearly to OSHW:

Reuse. People don't spend their time and money re-inventing the wheel, and those resources then become available to add quality to a project and innovate. Reusing existing designs is also a very powerful risk management technique. Of course, proprietary design can also be reused, but in a necessarily reduced scope. Proprietary designs benefit from fewer pairs of eyes scrutinizing them and finding bugs. They also tend to be more geared to solving a particular problem and are therefore harder to use in a new setting. Finally, in a design reuse scenario, proprietary designs often need dedicated and costly legal efforts to guarantee the licensee is granted a sufficient degree of freedom.

Avoidance of vendor lock-in. All other things being equal, vendor lock-in is never in the interest of the user of a given technology. OSHW makes lock-in virtually impossible. On the other hand, vendor lock-in is often part of the commercial strategy of proprietary design

ISBN 978-3-95450-189-2

PandABox: A MULTIPURPOSE PLATFORM ADAPTED FOR MULTITECHNIQUE SCANNING AND FEEDBACK

Yves-Marie Abiven[†], Jérome Bisou, Guillaume Renaud, Frédéric Ta
Synchrotron SOLEIL, Gif-sur-Yvette, France

Isa Servan Uzun, Michael Abbott, Andrew Cousin, Tom Cobb
Diamond Light Source, Oxfordshire, United Kingdom

Abstract

Synchrotron SOLEIL and Diamond Light Source are two third generation light sources located respectively in France and the UK. In 2015, both facilities started the collaboration project "PandA" to overcome technical limitations of SPIETBOX at SOLEIL and Zebra at Diamond as well as to manage obsolescence of the products. The collaboration enables both institutes to share the technical leadership on hardware, firmware and software developments. The initial objective is to achieve multi-channel encoder processing to synchronize motion systems and data acquisition during experiments addressing simultaneous and multi-technique scanning. However, its design based on Xilinx Zynq SoC is thought to be powerful and modular in terms of firmware as well as for hardware. This flexibility permits envisaging derivative applications and interfacing to different third party hardware. This paper details the organization of this collaboration, status of the ongoing project in terms of hardware and firmware capabilities and the results of the first tests at both sites.

PROJECT STATUS

The first step on the project confirmed that the organisation of the project was efficient, sharing leadership between institutes. After few months agreeing specifications in term of hardware and firmware, the last 10 months have been intense, focusing on the design. On one side SOLEIL designed the 3 main boards which are integrated in a 19" rack. On the other side Diamond developed the firmware and software to interface PandABox to EPICS or TANGO control system, up to being able to configure the system and make acquisition. At each step of the design both institutes worked together towards reviewing and validating designs. The First prototype presented in Figure 1 was delivered in August. The prototype hardware was tested successfully along with initial firmware and software releases at both sites during September, and only a few hardware modifications were identified. Update of the electronic and mechanic designs were completed in October and second prototypes were ordered with expected delivery in January.

On the firmware side, Diamond delivered first version of the firmware allowing both institutes to start hardware tests: Processor, FPGA, SFP, FMC and all inputs and outputs. Communication via a TCP server permits to remotely work with the platform: Reading on-board sensors, setting and acquiring encoder position or generating triggers on TTL outputs. All standard configurations required for first application are available. Today the functional tests of the firmware are ongoing and already promising.

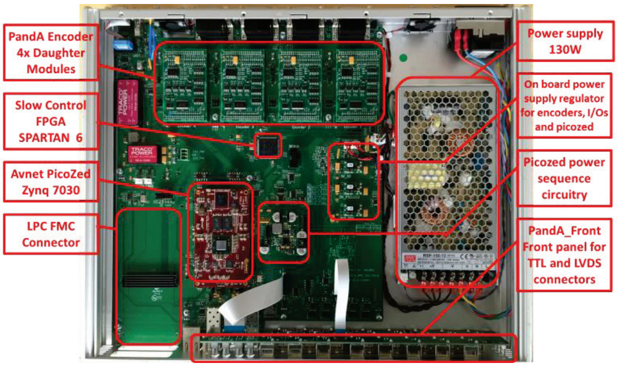

PandA Encoder 4x Daughter Modules
Slow Control FPGA SPARTAN 6
Avnet PicoZed Zynq 7030
LPC FMC Connector
Power supply 130W
On board power supply regulator for encoders, I/Os and picozed
Picozed power sequence circuitry
PandA_Front Front panel for TTL and LVDS connectors

Figure 1: PandABox inside view.

HARDWARE ARCHITECTURE

The hardware architecture is designed to meet the requirements for simultaneous and multi-technique scanning with support for a wide range of encoder protocols. The architecture presented in Figure 2 was developed to be modular and flexible in order to be open for a maximum numbers of applications. The platform is built with:

I/O interfaces

- Multi-Channel TTL and LVDS I/Os for synchronous
- triggering and clocking.4-Channels of encoder interface I/Os, supporting Incremental, Absolute encoder protocol (AquadB, SSI, EnDat and BiSS). Integrated on the platform as a mezzanine boards allowing this functionality to be updated.
- A fully compliant Low-Pin Count FMC slot for interfacing to analog and digital off-the-shelf boards or custom I/O modules.
- JTAG allowing programming and debugging for Zynq and Spartan-6 during prototyping.
- Serial port, giving a terminal for linux administration.

Communication interfaces

- 3-Channels of SFP Gigabit Transceiver interface for UDP triggering, Timing System, Diamond Communication Controller or custom high-speed serial connectivity.
- A Gigabit Ethernet connectivity for control systems integration and high-speed data acquisition.

[†] yves-marie.abiven@synchrotron-soleil.fr

ISBN 978-3-95450-189-2

NEW CONTROLS PLATFORM FOR SLAC HIGH-PERFORMANCE SYSTEMS[*]

Till Straumann, R. Claus, J. M. D'Ewart, J.C. Frisch, G. Haller, R.T. Herbst, B. Hong, U. Legat, L. Ma, J.J. Olsen, B.A. Reese, L. Ruckman, L. Sapozhnikov, S.R. Smith, J. Vasquez, M. Weaver, E. Williams, C. Xu, A. Young, SLAC, Menlo Park, California, USA

Abstract

The 1 MHz beam rate of LCLS-2 precludes the use of a traditional software solution for controls of "high-performance systems" which operate at this rate, such as BPMs, LLRF or MPS. Critical algorithms are ported into FPGA logic and administered by ordinary PCs via commodity Ethernet. SLAC has developed a controls architecture which is based on FPGA technology interconnected by 10G Ethernet and commercially available ATCA shelves. A custom ATCA carrier board hosting an FPGA, memory and other resources provides a "common platform" for many applications which can be implemented on AMC cards which are plugged into the carrier. A library of firmware modules including e.g., timing, history buffers and reliable network communication together with corresponding software packages complement the common platform hardware and provide a standardized environment which can be employed for a variety of high-performance applications across the laboratory.

INTRODUCTION

The LCLS-2 XFEL is currently under construction at SLAC. A superconducting linac supplies electron bunches at a rate of up to 1 MHz. Several diagnostics, controls and protection systems must be able to resolve individual bunches and process data in real-time. A conventional control system based on computers with peripheral devices and software based algorithms is – at the current state of the art – not capable of meeting the necessary throughput and latency requirements (considering that the actual data rate some subsystems have to handle is much higher than the beam rate of 1 MHz).

Therefore, the bulk of processing must be implemented in programmable logic, leveraging a high level of parallelism in hardware. SLAC has developed a new platform which is based on the following technologies:

- ATCA [1] form-factor and commercial shelves.
- IPMI management [1].
- 10 G Ethernet communication.
- FPGA on generic ATCA carrier.
- Application-specific AMC cards.

An library of firmware components and a "common-platform" software framework complement this versatile and powerful instrumentation and controls platform.

Since the hardware design has already been presented elsewhere [2,3] this paper shall focus on the communication protocols and software framework.

HARDWARE OVERVIEW

The platform employs a COTS ATCA shelf that provides standardized mechanical support, power, cooling, management and interconnect. The backplane's fabric interface is configured in a dual-star topology supporting up to 10 Gbps per channel (with a 40 Gbps upgrade path). A block diagram of the system is depicted in Fig. 1.

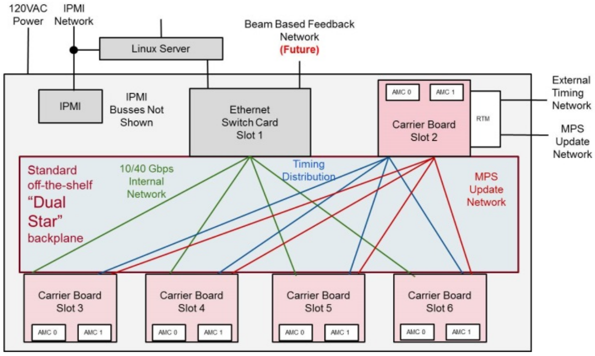


Figure 1: Common Platform System Diagram.

One star is concentrated at a 10 Gb Ethernet switch that provides the main connectivity between ATCA boards as well as an external Linux server computer (Fig. 1) which is responsible for interacting with the firmware.

The second star is used to broadcast timing data and to gather MPS (machine-protection) information that is further aggregated by the custom general-purpose carrier board (as described in the next paragraph) in slot 2'. Timing and MPS connectivity to the outside is provided by a custom RTM.

One core element is the custom carrier board that hosts a Xilinx Kintex Ultrascale XCKU040 or -060 FPGA and 8 GB of DDR3 memory. The FPGA's SerDes interfaces are connected to the Fabric interface as well as four AMC bays. These bays can receive single- or dual-wide, full-height AMCs. Additional LVDS and high-speed I/O is available via the zone-3 connector to RTMs.

A variety of AMC cards (commercial and custom) can be plugged into the carrier. A common use case are high-speed ADCs for applications like BPMs or bunch-length monitors. Placing analog components on well-shielded AMC cards ensures that EMI effects can be minimized.

[*] Work supported by the US Department of Energy, Office of Science under contract DE-AC02-76SF00515

OPEN HARDWARE EXPERIENCE ON LNLS' BEAM DIAGNOSTICS

G. B. M. Bruno*, D. O. Tavares, H. A. Silva, F. C. Sant'Anna,
J. L. Brito Neto, L. M. Russo, L. A. Martins, S. R. Marques
LNLS, Campinas, Brazil

Abstract

LNLS Beam Diagnostics Group has decided to adopt and develop open-hardware technologies for most of its projects, partnering with other institutes and companies to design and build a complete RF BPM electronics, from the BPM pick-up to the operator interface. That decision resulted in advancements and learning, bringing new technologies, flexibility and knowledge, but also brought some hardships and new challenges. This paper details the history, advantages and difficulties of this open hardware approach.

INTRODUCTION

Sirius is the new synchrotron light source currently being built by LNLS in Brazil, with 518.4 m circumference and expected to be installed in late 2017, for commissioning in mid 2018 [1]. A total of 255 units of in-house-developed BPM electronics will be built in order provide feature-rich data acquisition system for button and stripline BPM pick-ups of the storage ring, booster and booster to storage ring transfer line.

The Sirius digital BPM electronics consists of 4 main pieces of hardware: a standalone RF front-end electronics for analog signal conditioning (RFFE), a digitizer mezzanine card for analog-to-digital conversion (FMC ADC), a FPGA-based carrier board acting as a digital back-end for acquisition, signal processing and communication (AMC FMC Carrier - AFC), as depicted in Fig. 1 and a COTS MicroTCA.4 crate providing a wealth of services for the digital back-end and its mezzanine cards, including power supply, cooling, crate management, a CPU host and shared trigger and clock lines for all crate slots. The crate hosts several digital back-end boards (up to 10), whereas each digital back-end board hosts 2 digitizer mezzanine cards. Each digitizer has 4 ADC channels thus being able to digitize the set of 4 analog signals provided by each BPM pick-up.

Apart from the COTS crate and its infrastrucutre components, all electronics developed by LNLS for the Sirius RF BPM electronics are open hardware, licensed under the CERN OHL license [2] and available through the CERN Open Hardware Repository [3]. The following sections will give a brief motivation for the adoption of open hardware, in general and in the context of Sirius electronics development, and then will describe the evolution of three open hardware projects: RFFE, FMC ADC and AFC.

THE OPEN HARDWARE APPROACH

There are several different definitions and intents associated with the names "open hardware" and "open source hardware", which are, sometimes, conflicting ones. The most common meaning, borrowed from the free software and open source software initiatives, requires the user to have access to the design files, being allowed to do modifications, to distribute and to use the project without restrictions [4]. Public release of the work may be required by some definitions, such as the one defined by the Open Source Hardware Association (OSHWA) [5], while some licenses, such as CERN OHL [2] may only require documentation to be sent to users who acquire the products.

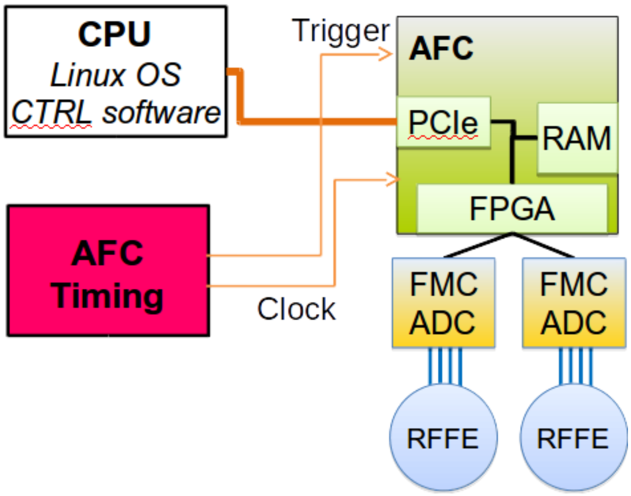


Figure 1: Simplified scheme for Sirius BPM electronics.

An important and frequently misunderstood aspect of the open hardware definition is its use in commercial applications. Most open hardware definitions do not prohibit commercial use of the project, in fact, most of them explicitly allow commercial use, indistinctively of application or profit intention. In effect, open hardware advocates usually do not consider restrictions on commercial use to be legitimate on open hardware projects [5, 6]. All definitions, licenses and projects discussed in this work allow for unrestricted commercial use.

The motivations for adopting open hardware often includes better collaboration among users and suppliers, reducing of work duplication on similar projects, and allowing for quick advance on technologies by improving existing designs. For science instrumentation, open hardware may be preferred as a mean to allow or facilitate experiment duplication and to fully document and submit the results to peer review. An open hardware adopter is also less dependent on a specific company, as several different companies may produce and support the same open hardware design, or variations of it.

[*] gustavo.bruno@lnls.br

ISBN 978-3-95450-189-2

EMBEDDED CONTROL SYSTEM FOR PROGRAMMABLE MULTI-PURPOSE INSTRUMENTS

M. Broseta, J. Avila-Abellan, S. Blanch-Torné, G. Cuní, D. Fernández-Carreiras, O.Matilla, J. Moldes, M. Rodríguez, S. Rubio-Manrique, J. Salabert, X. Serra-Gallifa,
ALBA Syncrothron, Cerdanyola Vallès, Spain

Abstract

At the ALBA's Computing Division, we have started the development of a high-performance electrometer, the ***Em#*** project, as a versatile and full customizable equipment. It is based on a SPEC board (simple PCIe FMC carrier) with customizable FMC cards and an SBC (Single Board Computer), altogether built in a single cost-optimized instrument. The whole device is designed to provide a wide range of functionalities to fulfill unique and complex experiments by means of configuration changes instead of having specific instruments.

Within the controls software development group, we started the development of a full embedded control software, based on a Linux OS that communicates with the SPEC's FPGA using the PCIe bus. This approach enables the integration of complex operations and functions in real time to higher software layers, as well as the local control, setup and diagnostics via an integrated touch-screen display controlled by the I2C protocol. For a user-level control, the system provides an SCPI API (Standard Commands for Programmable Instruments) allowing an easy integration to any control system. This paper describes the design process, main aspects of the data acquisition and the expected benefits during the integration in the Control System.

INTRODUCTION

High accuracy low current readout is an extensively demanded technique used in 3rd generation synchrotrons. They comprise a need both for diagnostics and data acquisition in today's photon labs. In order to tackle the problem of measuring from various sources of different nature and magnitude synchronously, while remaining flexible at the same time, ALBA developed years ago a 4 independent channel electrometer, the Em, based on trans-impedance amplifiers with high resolution ADC converters integrated and an Ethernet communication port.

The new ***Em#*** is the evolution of the 4-channel electrometer Em widely used in at ALBA since 2011. This new product solves a few limitations and provides new functionalities to make it more versatile and customizable.

ELECTRIC DESIGN

The ***Em#*** project uses a Simple PCI Express FMC Carrier (SPEC [1]) board to command a FPGA Mezzanine Card (FMC), designed to work as a 4-channel ADC and transfer the processed data to a Single Board Computer (SBC), the Intel NUC DE3815 [2], using its high-speed serial computer expansion bus (PCIe).

The SPEC, it is a cost-optimized design developed by the CERN under the Open Hardware Licence (OHL) that mainly works as FMC carrier. It is powered by Xilinx Spartan 6 FPGA [3] for custom gateware designs using High-speed serial connectivity, a PCIe interface and an FMC slot.

This hardware configuration allows ***Em#*** implement its own FPGA control code, to manage a 4-channel ADC converter in the FMC card. The communication and data sharing with the main control software in the SBC are also part of the control code routines implemented in the FPGA. But apart form the SPEC and SBC boards, there are other hardware boards included in this equipment. Figure 1 shows the hardware diagram block of the ***Em#*** project.

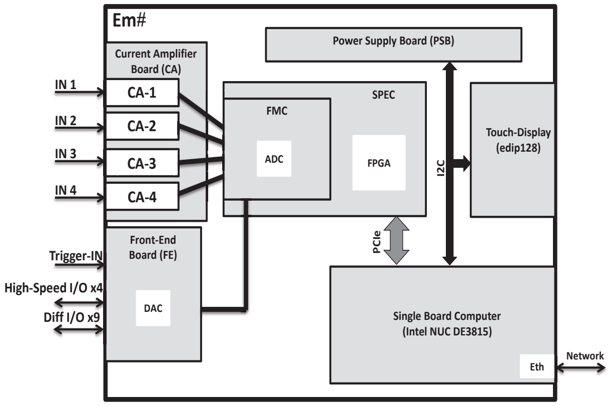


Figure 1: Em# hardware diagram block.

The Current Amplifier board (CA) contains the circuitry needed to communicate and control the 4 current amplifiers (CA-X). The Front-End board (FE) manages the trigger input and the different IO ports (4 High-Speed I/O ports and 9 Differential I/O ports). The Power Supply Board (PSB) supplies the equipment with different voltages needed by each module or board. A Touch-Screen (edip128) monitors the status and lets a general configuration of the equipment.

SOFTWARE DESIGN

The software development has been divided into three software projects, all together distributed in a single software package:

- The Linux OS
- The gateware (FPGA software)
- The main control software in the SBC (ALIN)

GATEWARE AND SOFTWARE FRAMEWORKS FOR SIRIUS BPM ELECTRONICS

L. M. Russo*, J. V. F. Filho, LNLS, Campinas, SP, Brazil

Abstract

The Brazilian Synchrotron Light Laboratory (LNLS) is developing a BPM system based on the MicroTCA.4 standard comprised of AMC FPGA boards carrying FMC digitizers and an AMC CPU module. In order to integrate all of the boards into a solution and to support future applications, two frameworks were developed. The first one, gateware framework, is composed of a set of Wishbone B4 compatible modules and tools that build up the system foundation, including: PCIe Wishbone master; FMC digitizer interfaces; data acquisition engines and trigger modules. The gateware also supports the Self-Describing Bus (SDB), developed by CERN/GSI. The second one, software framework, is based on the ZeroMQ messaging library and aims to provide an extensible way of supporting new functionalities to different boards. To achieve this, this framework has a multilayered architecture, decoupling its four main components: (i) hardware communication protocol; (ii) reactor-based dispatch engine; (iii) business logic, comprising of the specific board functionalities; (iv) standard RPC-like interface to clients. In this paper, motivations, challenges and limitations of both frameworks will be discussed.

INTRODUCTION

Sirius is a new 3 GeV synchrotron light source under construction in Brazil, with a 0.27 nm.rad natural emittance and 518 meters circumference. The beginning of machine installation is scheduled to the end of 2017 [1].

In this context, a BPM electronics system has been specified, designed and developed by the Beam Diagnostics team at LNLS, reaching its final phase of long-term testing and hardware manufacturing in the following year [2].

As the system was developed from scratch, employing new high-performance data acquisition and communication technologies (e.g., MicroTCA.4, FPGA, FMC, PCIe) [3], the need for an FPGA gateware and software infrastructure frameworks emerged. For that matter, it was sought the use of consolidated codebases and collaborative development through an open source approach. Examples of this were the initial collaboration with the Warsaw University of Technology, the use of a community-driven set of repositories aimed at building generic software/gateware modules from the OHWR collaboration [4] and the use of projects such as the ZeroMQ messaging library [5], the CZMQ High-Level C Binding library [6] and the Malamute Messaging Broker [7] which leveraged many years of development.

In the next sections, the requirements and the details of the these two frameworks, licensed under the copyleft GPLv3 and LGPLv3 licenses, will be described.

GATEWARE FRAMEWORK

The gateware framework consists of a set of modules, mainly written in VHDL, that interconnect with each other and to external interfaces. It can be used as a basis to other designs. The general architecture is as depicted in Fig. 1.

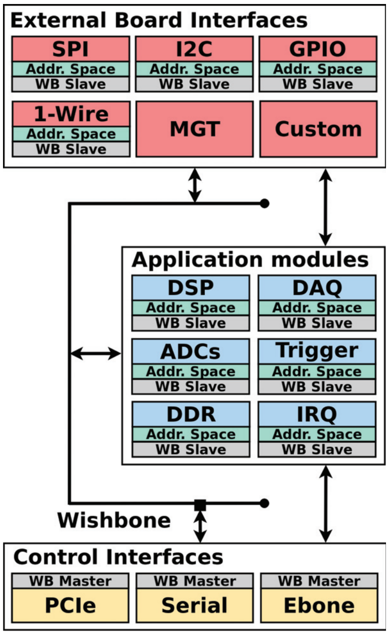


Figure 1: Gateware Framework Architecture.

General Description

The first basic component of the framework is the *Control Interfaces*, as shown in Fig. 1, and it encompasses standard communication interfaces to a controlling node, acting as a Wishbone Master to the gateware side. Currently, 3 types are supported: PCIe Gen1 (for Xilinx Artix7 FPGA) and Gen2 (for Xilinx Kintex7 FPGA), with accompanying linux driver with PIO, single buffer and scatter-gather DMA; RS232 Syscon for simple serial communication supporting autobaud generation; preliminary support for Ethernet MAC + Etherbone [8] for UDP communication and userspace software library.

The second layer, *Wishbone Infrastructure*, is a set of Wishbone modules and functions that provide the general interconnection between controllable modules (i.e., that can receive/transmit configuration parameters and/or low-bandwidth data) and address space enumeration with SDB [9] support. This is the standard adopted within the framework for consistently interfacing with all *Control Interfaces*.

The third layer, *Application Modules*, is where all of the framework functionalities reside, comprising components from third-parties and LNLS Beam Diagnostics team. A

[*] lucas.russo@lnls.br

ISBN 978-3-95450-189-2

A FRAMEWORK FOR DEVELOPMENT AND TEST OF xTCA MODULES WITH FPGA BASED SYSTEMS FOR PARTICLE DETECTORS[*]

T. C. Paiva, L. A. Ramalho, A. A. Shinoda, Faculdade de Engenharia de Ilha Solteira, UNESP, Brazil
A. Cascadan[†], V. F. Ferreira, M. Vaz, Núcleo de Computação Científica, UNESP, Brazil

Abstract

This work describes a framework to develop firmware for ATCA carrier boards with FPGA. It is composed of an ATCA IPMI protocol implementation for environmental monitoring and control, and a companion XVC protocol implementation for remote FPGA configuration and system debugging. A study case is also presented of the development of a setup to validate a Level 1 Tracker Trigger System proposed for CMS at HL-LHC.

INTRODUCTION

The ATCA standard, originally created for the telecommunication industry, has recently aroused the interest of other fields. In the physics community the ATCA is already been used [1, 2], due to the high level of support to system monitoring and control it provides [3]. As an example, the ITER experiment developed a modular control and data acquisition systems designed on ATCA platform [4]. Also, the LBNL synchrotron built an ATCA readout system with an ATCA processor blade which performs image descrambling and formatting [5]. And, recently, the ATLAS Cathode Strip Chamber (CSC) back-end readout system has been upgraded to ATCA environment with high speed links, commercial Pigeon Point IPMC and Timing Trigger and Control (TTC) I/O for synchronization [6].

ATCA standard is also the choice of the AM+FPGA group to the development of a Level 1 Tracking Trigger (L1TT) system for the Compact Muon Solenoid (CMS) operating in the High Luminosity LHC [7]. Sao Paulo Research and Analysis Center (SPRACE), as part of this international collaboration, is in charge of building a demonstration framework to validate those R&D prototypes. It consists of two ATCA shelves with custom ATCA carrier boards, named Pulsar 2b [8]. One of them implements the Data Sourcing System, which is an emulator for the Outer Tracker detector electronics output, and the other contains the Pattern Recognition System being proposed.

The L1TT electronics, which will be physically inaccessible during the LHC runs, requires safe and reliable operation, remote configuration and JTAG tests of all its FPGA devices. These requirements are attended by our framework, as it follows Intelligent Platform Management Interface (IPMI) [9] and Xilinx Virtual Cable (XVC) [10] specifications, and also it uses Pulsar 2b ATCA carrier boards. The following will describe the implementation of this system and its application in the demonstration setup.

ATCA AND IPMI

The ATCA is a standard defined by PICMG [3] that specifies a chassis system (shelf) intended to provide a comprehensive and reliable environment for hosting carrier boards and their extensions, such as mezzanines and transition modules. It defines also the concept of Hardware Platform Management (HPM) consisting of a distributed control system, which relies on the IPMI specification. Controllers collect information from different types of sensors spread in the system and take actions, like speeding up cooling fans or shutting down modules, to ensure a safe environment for the electronic boards. HPM also enables hot swap operations to allow replacement of electronic units with the system powered on. There are three types of controllers in this architecture:

Shelf Manager Controller (ShMC): it is the central element in the shelf, located in the Shelf Manager (ShMC) board, that gathers information from the installed hardware, generates alarms and controls power supply and fan speed.

Intelligent Platform Management Controller (IPMC): it is installed on each carrier board and is a local HPM agent, directly conneted to ShMC.

Modular Management Controller (MMC): it is the simplest management agent residing in the extension modules of the carrier boards, like Advanced Mezzanine Card (AMC) and Rear Transition Modules (RTM), that is only able to execute basic commands sent by the IPMC.

The HPM management elements connect to each other using the Intelligent Platform Management Bus (IPMB) as a physical layer for the IPMI communication [11]. IPMB-0 stands between the ShMC and IPMC cards and IPMB-L connects IPMC and MMC devices. The Figure 1 shows the controllers and the connection betweeen them inside a shelf.

The starting point of the framework was the Pulsar 2b carrier board with an IPMC card already used for simple tasks but with no support for IPMI operations. The previously chosen proprietary real time operating system, RTX from Keil, prevented the use of open source projects as base of the IPMI solution, like CoreIPM, and it was decided for an implementation from the scratch, with a very minimalist approach. It proved necessary support for the hot swap

[*] This material is based upon work supported by the São Paulo Research Foundation (FAPESP) under Grant No. 2013/01907-0 and by funds provided by the cooperation agreement with PADTEC S/A under FUNDUNESP Grant No. 2215/2013.

[†] cascadan@ncc.unesp.br

ISBN 978-3-95450-189-2

UVX CONTROL SYSTEM: AN APPROACH WITH BEAGLEBONE BLACK

S. Lescano, INSA Lyon, Villeurbanne, France
A. R. D. Rodrigues, E. P. Coelho, G. C. Pinton, J. G. R. S. Franco*, P. H. Nallin,
Brazilian Synchrotron Light Laboratory (LNLS), Campinas, Brazil

Abstract

UVX is a 1.37 GeV synchrotron light source that has been in operation by the Brazilian Synchrotron Light Laboratory (LNLS) since 1997. Its control system, which was completely developed in-house, has received some upgrades lately in order to get around issues from aging, improve performance and reduce maintenance costs. In this way, a new crate controller was designed. It is based on BeagleBone Black single-board computer (SBC) [1], a cheap open hardware and community-supported embedded Linux platform that will be adopted for some control system applications in Sirius [2], the upcoming brazilian light source. In this paper, we describe an overview of the design and results obtained.

INTRODUCTION

As shown in [3], the control system of our machine is organized in three levels. The lowest one comprises 3U VME-like crates with I/O cards and a local controller module, responsible for managing these cards and communications to the upper level. We name this low-level control system based on crates with I/O cards as LOCO (from LOcal COntroller).

Since the start of UVX operation, I/O cards are the same. Most of them has only TTL digital pins and analog inputs and outputs with 12 or 16-bit resolution. There are also specific boards for reading Pt100 temperature sensors or counting pulses, for instance. However, LNLS Controls Group developed three generations of local controller boards, as shown in the next section.

UVX LOCAL CONTROLLERS TIMELINE

In a nutshell, the goal while developing a local controller for UVX control system is always to build a reliable, low-cost and general-purpose module to manage the crates. Since the beginning of control system implementation, another important feature is always present in our controller designs: units of a given model run the same program, a universal software that contains routines to read and write all types of I/O boards and performs all possible operations specified by high-level applications. Running local controllers with a unique software simplifies maintenance tasks and the embedded software development process. With an architecture like that, only high-level applications know the equipments controlled by each crate. Also, the application protocol used in communications to controller crates was specified by LNLS engineers and has never suffered drastic changes.

First Generation: Z80 and Serial Communication

First version of UVX local controller board was based on a Z80 microprocessor. Its software, named PSICO, was written in assembly language. Controller's communication interface was designed internally and is based on RS-485. Although these boards are in operation until today, they are treated as a legacy system, as we don't provide updates for the embedded software anymore and many of their electronic components are obsolete.

Second Generation: eZ80 and Ethernet

Aiming the adoption of a widely used communication interface standard (Ethernet), a new version of the local controller board was developed in the early 2000s [4]. Zilog eZ80F91 microcontroller was the hardware platform chosen. The embedded software was rewritten in C and renamed to PROSAC. Some of these boards are still in operation in UVX storage ring systems, despite the lack of updates for this design in the past years.

Third Generation: SBC with FreeBSD

Because second generation local controller was too much dependent on its hardware platform (an eZ80 microcontroller with built-in Ethernet MAC), the design of a new controller was started. At that time, Controls Group engineers wanted to experiment with the emerging world of embedded Linux platforms too.

An industrial-class single-board computer (Advantech PCM-4153F) was picked. Local controller software (PROSAC) was completely rewritten in C, taking into account libraries such as Pthread (POSIX threads) and the presence of an abundant secondary memory, used for storage of waveforms for special applications. Routines to read and write I/O cards were implemented with the mapping of SBC's PC/104 bus into LOCO crate backplane bus lines.

Although we have moved to an environment which is not hard real-time with this design, in practice all requirements for controller operation in UVX were satisfied. Today we have many of these local controllers under operation, notably those which interfaces to UVX storage ring high current power supplies.

Various flavours of Unix and Linux operating systems were tested for use with Advantech SBC. FreeBSD surpassed all the others because tests revealed that it was more responsive while performing synchronized operations over power supplies (beam energy ramp and magnets cycling).

This local controller based on Advantech single-board computer is the most failsafe we ever made. Machine operators always say that.

[*] guilherme.franco@lnls.br

ISBN 978-3-95450-189-2

openMMC: AN OPEN SOURCE MODULAR FIRMWARE FOR BOARD MANAGEMENT

H. A. Silva[*], G. B. M. Bruno, LNLS, Campinas, Brazil

Abstract

openMMC is an open source firmware designed for board management in MicroTCA systems. It has a modular architecture providing decoupling between application, board and microcontroller-specific routines, making it useful as a base for many different designs, even those using less powerful controllers. Despite being developed in a MicroTCA context, the firmware can be easily adapted to other hardware platforms and communication protocols. The firmware is based on the FreeRTOS operating system, over which each monitoring function (sensors, LEDs, Payload management, etc) runs its own independent task. The OS, despite its reduced footprint, also provides numerous tools for reliable communication among the tasks, controlling the board efficiently.

INTRODUCTION

LNLS Beam Diagnostics team is currently developing the Sirius' Beam Position Monitor (BPM) electronics and has adopted the MicroTCA.4® standard by PICMG [1] in its boards designs [2].

The main board on the electron BPM system is the AMC FMC Carrier (AFC) [3], which is a general purpose FPGA board that hosts up to two mezzanine cards in FMC form factor. Those smaller cards carried by AFC can be implemented to have many different applications (e.g. fast digitizers, SFP modules, data pre-processing modules, RS485 communication). In the BPM application, fast digitizer FMCs will be used in order to read the BPM analog signals.

AMC boards, as the application boards are called in the MicroTCA system, must have implemented a Module Management Controller (MMC), which usually is a microcontroller responsible for monitoring the board health and acting as a communication channel between the system manager and application using Intelligent Platform Management Interface (IPMI) [4].

openMMC was created to be an open source (using GPLv3 license) modular and generic firmware, easily portable to other platforms. It runs over FreeRTOS, which gives the developer a wide set of tools to implement complex monitoring functions or advanced hardware control. Given its modular independent structure, it is possible to use the firmware in applications outside MicroTCA environment with little effort, changing the communication protocol in its lower layers, for example.

The project development is being versioned in a GitHub repository [5], using pull requests and issues tracking as its main collaboration tools.

MOTIVATION

The development of openMMC started after some unfruitful tests with the available open source MMC implementations. Those firmwares were developed to run on specific target boards, requiring a substantial effort in order to port them to AFC's hardware and begin a functional evaluation process.

After some attempts, it was clear that porting the code basically meant to rewrite it from scratch, given that its low level driver and application functions were deeply intertwined.

The hardware flexibility offered by MicroTCA was not being accompained by its MMC firmware architecture, since each board implementation had to recreate the managing firmware. openMMC was thought to be the hardware independent firmware that could meet this need.

FreeRTOS

FreeRTOS [6] is a popular open source real time operating system for embedded controllers that has already been ported to a wide range of CPU architectures.

It features a preemptive scheduler that allows the application code to run multiple tasks in parallel with a single core. The scheduler decides which task will run based on its priority. More important tasks are always executed first, whilst blocking the lower priority ones. If one or more tasks are on the same priority level, a round-robin time slice method is applied, ensuring that all of them are executed within its time limits.

Most of OS-native tools used in communication between tasks are also implemented on FreeRTOS (e.g. semaphores, software timers, queues). The developer can also use some unique functions provided, such as direct-to-task notifications, event groups and co-routines.

Except for task creation, all tools on FreeRTOS can be stripped from its compilation, reducing resource usage, thus enabling use of cheaper and lower-power microcontrollers.

FreeRTOS project uses a modified GPLv3 license, which allows the user to implement its application on top of the OS without having to publish proprietary code [7].

FIRMWARE STRUCTURE

The firmware was structured in order to be easy to upgrade and port to different boards and controllers. Therefore four different abstraction layers are implemented: *Application*, *Hardware Abstraction*, *Port* and *Driver*, arranged as in Fig. 1.

Application

The Application layer holds high-level tasks responsible for deciding which action will be taken based on the in-

[*] henrique.silva@lnls.br

ISBN 978-3-95450-189-2

IMPLEMENTATION OF A PRECISION LOGARITHMIC AMMETER

W. R. de Araujo*, M. R. Bissiano, G. Paulino, D. Galante, LNLS, Campinas, SP, Brazil

Abstract

A precision ammeter is in development for the acquisition of sensor signals such as photodiodes, gold mesh (by photoelectron effect) and ionization chambers. One of the problems of conventional ammeters is the automatic scale selection, which hinders many measurements performed in ample energy ranges. The ammeter in development is based on a different methodology than present on most commercial systems, using a logarithmic amplifier. This choice can provide a logarithmic response output in the range of pico to milli-amperes. The electronic board is in development by Brazilian Synchrotron Light Laboratory (LNLS), and it is being installed and tested at the Toroidal Grating Monochromator (TGM) Beamline.

BASIC OF LOGARITHMIC AMPLIFIER

A logarithmic amplifier (LogAmp) is a device that can express the output as a logarithmic function of the input, either in electrical current or voltage. This device is commonly used in telecommunications for compression of voice and video signals, and here it is explored its potential for measuring and monitoring currents produced in light sensors. [1]

The most common LogAmp uses the exponential curve of a bipolar junction transistor (BJT) to convert an input current in a logarithmic output voltage. This class of LogAmp usually operates on unipolar inputs and it is very sensitive to temperature variations. The most simplified model is composed by a NPN transistor connected at the negative feedback of an operational amplifier, as shown outlined in Figure 1. The transfer function of this circuit is:

$$V_{out1} = -V_{BE} \approx Vt \ln\left(\frac{I_{IN}}{I_s}\right) \qquad (1)$$

Where I_s is the reverse saturation current and V_t is the thermal voltage. The first is a constant dependent of the construction parameters, materials, and, as consequence, temperature of operation. The thermal voltage is a universal parameter of transistors, and depends only on the temperature. The approximation shown above can be used due to the fact that the input current in the OpAmp is approximately zero.

A first solution to reduce the dependence of temperature is introducing a second LogAmp, which acts like a reference to the first through a subtracting circuit, as shown in the Figure 1.

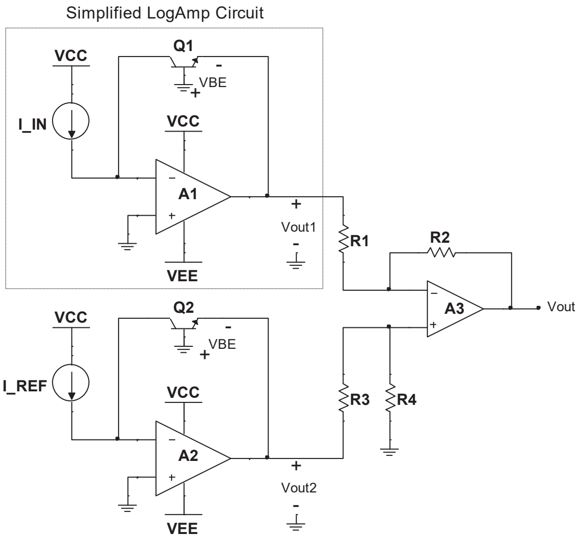


Figure 1: A referenced current LogAmp is shown above. A simplified model of a LogAmp circuit is outlined. The subtraction of two simple LogAmps with the third OpAmp (A3) eliminates the temperature dependence generated by I_s.

The general transfer function of the above circuit is given by:

$$V_{out} = V_{LOG} = V'_y \ln\left(\frac{I_{IN}}{I_s}\right) - V'_y \ln\left(\frac{I_{REF}}{I_s}\right)$$

$$V_{LOG} = V_y \log\left(\frac{I_{IN}}{I_Z}\right) \qquad (2)$$

Where V_y is a function of V_t and the gain attributed of the resistor in the subtractor circuit, and I_z is a conversion name to I_{ref}. This approach is able to eliminate the temperature dependence caused by I_s because the transistors have the almost identical properties and are in close thermal contact for proper cancellation. However, V_y is proportional to V_t, and so, it is still sensitive to temperature variations. By adding subsequent temperature-compensation circuitry this dependency is virtually eliminated (normally, an additional OpAmp amplifier stage with a resistive temperature detector [RTD], or similar device, is incorporated as part of the gain) [1]. Several commercial integrated circuits have embedded some type of temperature compensation which can stabilize the final gain V_y of the system.

DEVELOPMENT OF THE CONCEPT

For the development of the Logarithmic Ammeter (Log-Ammeter), it was designed an electronic based on a commercial LogAmp IC, with a range of 200 dB. This circuit has an internal reference current of 100 nA and a temperature stabilizing embedded in the chip. The system

* william.araujo@lnls.br

OPERATION EXPERIENCE AND MIGRATION OF I/O CONTROLLERS FOR J-PARC MAIN RING

N. Kamikubota[#], S. Yamada, K.I. Sato and N. Yamamoto, J-PARC, KEK & JAEA, Japan
T. Iitsuka, T. Aoyama and S. Yoshida, Kanto Information Service, Tsuchiura, Japan
H. Nemoto, ACMOS Inc., Ibaraki, Japan

Abstract

In 2008, when we started beam commissioning of J-PARC MR, I/O controllers (EPICS IOC) were consisted of about 80 pieces of VME-bus computers and a few PC-based controllers. The total number of IOCs was 80-90.

In 2016, we have non-VME IOCs: a) Yokogawa F3RP61 (Linux-based CPU with PLC I/O modules), b) virtual ioc (EPICS IOC on a virtual machine), and c) commercial micro-server. The total number of IOCs is around 170.

Histories and characteristics of VME and non-VME IOCs are described in this report. In addition, based on operation experience since 2008, reliability of VME-bus computers is discussed.

INTRODUCTION

J-PARC (Japan Proton Accelerator Research Complex) is a high-intensity proton accelerator complex, located in Ibaraki, Japan. It consists of three accelerators: a) 400-MeV Linac (LI), b) 3-GeV Rapid Cycling Synchrotron (RCS), and 30-GeV Main Ring (MR) [1-3]. The control system of J-PARC was developed using the EPICS (Experimental Physics and Industrial Control System) toolkit [4-5].

J-PARC MR started beam operation in 2008. Since then, the beam power of MR has been improved year by year. In June, 2016, the power reached 425kW (Figure 1). We continue beam commissioning toward the design goal, 750kW, and more [6].

To support commissioning activities, the MR control system has been often required to add new beam-diagnostic devices, or follow upgrade of existing power supplies, and so on, in a limited period. Thus, flexibility and extension ability are very important.

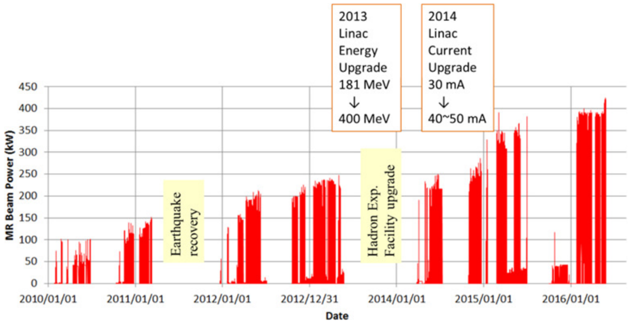


Figure 1: Improvement of MR beam power until 2016.

CHROCICLE OF IOC IN J-PARC MR

Initial Design of IOC for J-PARC MR

The control system for J-PARC MR was constructed in 2006-2008. At the time, a VME-bus was understood to be robust and highly reliable, compared with other platforms. Thus, we selected VME-bus computer as a primary I/O controller (hereafter IOC). About 80 pieces of VME-bus IOCs were introduced in 2008.

However, in our history, non-VME-type IOCs were also introduced. Even in 2008, a PC-based controller was introduced. Then, a PLC-type Linux-based CPU in 2009, an IOC running on a virtual machine in 2011, and a commercial micro-server ("Saba" IOC) in 2015, were introduced. The numbers of MR IOCs between 2008 and 2016 are summarized in Figure 2.

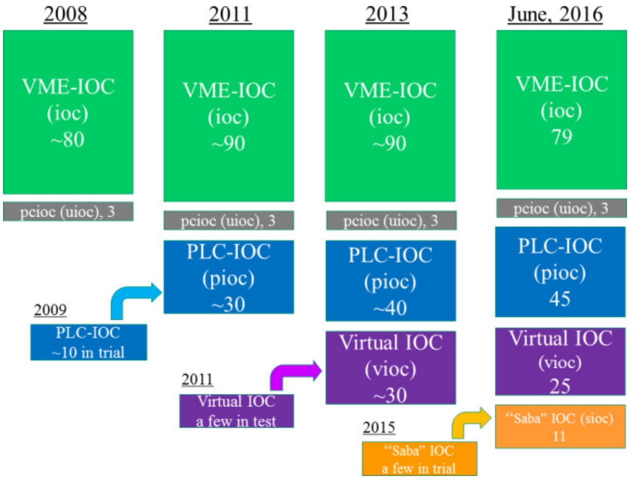


Figure 2: Numbers of MR IOCs between 2008 and 2016.

VME-IOC and PC-based IOC

In 2008, three (later four) models of VME-bus computers were used in MR operation: (a) the model, VMIC V7807 (after 2009, V7865), for DAQ of network-based digitizers (Yokogawa WE7000) [7], (b) the model, Sanritz SVA041, for controlling ladder-logic PLC (Yokogawa FA-M3) [8], and (c) the model, VMIC V7700, for VME-bus timing modules [9]. The specs of four models are given in Table 1. For all the models, we selected Intel-based chip, in order to run Linux OS. Front-end devices underneath VME-IOCs are shown in Figure 3.

One model of vacuum controllers has no external control interface but a RS485 port. It was not supported

[#] norihiko.kamikubota@kek.jp

ISBN 978-3-95450-189-2

TIMING AND SYNCHRONIZATION AT FRIB[*]

M. Konrad[†]
Facility for Rare Isotope Beams, Michigan State University, East Lansing, MI 48824, USA

Abstract

The Facility for Rare Isotope Beams (FRIB) requires a timing system for distributing a common time base for time-stamping data as well as for triggering actions of multiple devices distributed over the machine. Three different technologies are used to accomplish these goals. An event system based on commercial off-the-shelf hardware made by Micro-Research Finland provides time-stamps and triggers to more than 500 fast data-acquisition devices like beam diagnostics electronics, LLRF controllers, and machine protection nodes. This system is also used to distribute FRIB's complex beam pulse patterns with event rates of more than 50 000 events per second. For many hundred devices which require a lower timing accuracy like programmable logic controllers and computers the Precision Time Protocol (PTP) is used. Additionally the Network Time Protocol is used for legacy devices that do not support PTP, yet. We describe the architecture of the FRIB timing system and how the different timing subsystems are synchronized. We also describe how FRIB's beam pulse patterns are generated.

INTRODUCTION

FRIB [1] is a project under cooperative agreement between US Department of Energy and Michigan State University (MSU). It is under construction on the campus of MSU and will be a new national user facility for nuclear physics. Its driver accelerator is designed to accelerate all stable ions to energies >200 MeV/u with beam power on the target up to 400 kW [2]. The FRIB driver linac requires a timing system for distributing a common time base for time-stamping data as well as for triggering actions of multiple devices distributed over the facility. Commissioning of the front-end as well as the core components of the timing system is currently underway. The remaining parts of the accelerator and the timing system are planned to be commissioned over the next two years.

ARCHITECTURE

The client devices connected to the FRIB timing system have very different timing requirements. They can be divided into the three accuracy classes listed in Table 1. Due to the large number of timing clients in each of the classes the use of different technologies helps to reduce cost. Devices in the "high" accuracy class are connected to an event system that broadcasts its events over a fiber network. This system is based on commercial off-the-shelf products by Micro-Research Finland [3]. For the "medium" accuracy class the Precision Time Protocol (PTP version 2 as defined by the IEEE 1588-2008 standard) is used. In the last years this protocol has gained a lot of popularity for industrial automation applications and is widely supported by state-of-the-art programmable logic controllers. Additionally the Network Time Protocol (NTP) is supported. Compared to NTP the more modern PTP offers features like more efficient multicast messaging, cleaner handling of leap seconds as well as fault tolerance. This is why PTP is preferred over NTP even for clients in the "low" accuracy class. NTP is used only for devices that do not support PTP.

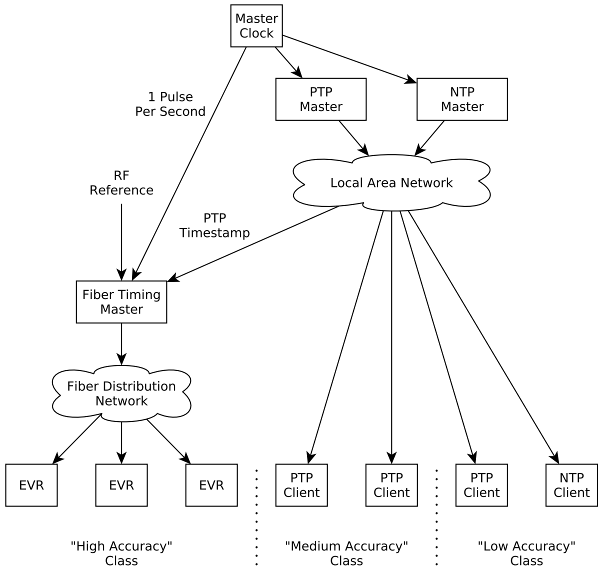


Figure 1: The FRIB timing system relies on three different technologies: A fiber-based event system as well as the Ethernet-based Precision Time Protocol (PTP) and Network Time Protocol (NTP). The three subsystems are synchronized to a common master clock.

The event system, the PTP timing system and the NTP timing system can be considered three independent subsystems of FRIB's timing system. All three subsystems are synchronized to a common master clock to ensure data which has been time-tagged by devices connected to different timing subsystems, can be correlated (see Fig. 1).

EVENT SYSTEM

The event system is the most complex and most critical part of the FRIB timing system. In the following we will give a more detailed description of this subsystem.

[*] Work supported by the U. S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661

[†] konrad@frib.msu.edu

ISBN 978-3-95450-189-2

PROCESSING SPE FILES FROM PRINCETON INSTRUMENTS DURING DATA ACQUISITION IN LNLS

D. B. Beniz[†], Brazilian Synchrotron Light Laboratory, Campinas, Brazil

Abstract

DXAS (*Dispersive X-ray Absorption Spectroscopy*) [1] beamline of LNLS (*Brazilian Synchrotron Light Laboratory*) uses a Princeton Instruments CCD, PyLoN [2], to acquire spectra of materials under analysis. Such detector produces an SPE binary file which can be read by a Python script, WinspecUtils.py [3], extracting intensities information on a 2D matrix for each acquired frame and then processing them as a NumPy [4] array.

Using that, a procedure to analyse partial data while the experiment is being performed in DXAS beamline was developed in Python language for their experiments. In this article we will focus on XMCD (*X-ray Magnetic Circular Dichroism*) analysis, describing its motivation and main aspects of its implementation.

MOTIVATION

XMCD experiments in DXAS involve a complex setup of equipment. Characteristics of materials under analysis also contribute to a succeed experiment. The fact is that only after hours of spectra acquisition and more hours of data analysis such results are achieved, and then it could be too late to go back and change something on experiment environment or even in the sample itself.

With that in mind, a way to pre-analyse XMCD results during the experiment could save time and effort, helping to make a decision to make something different and maybe remove, or mitigate, injurious interferences on the experiment even before the end of scheduled time of beam usage by the researcher.

The option by Python language to elaborate scripts to achieve such aim was the natural decision in LNLS since it is being recently used in the laboratory at almost all their beamlines to orchestrate the control of devices during experiments, using an internally developed Python package called Py4Syn [5], which is also available for external community. Looking on the Internet by a Python tool that read SPE files, the first step on data analysis, WinspecUtils.py was found and its test was pretty satisfactory. So, we decided to adopt it and develop a procedure to process the data array extracted from the SPE file during the spectra acquisition.

DATA FLOW TO CALCULATE XMCD

Each XMCD cycle is a combination of eight spectra, or frames, which are acquired by PyLoN CCD when the sample were submitted to a magnetic field with a specific sequence of directions, as here: [+ - - + - + + -], where "+" represents the positive direction of magnetic field and "-" the negative one. To guarantee the synchronization between the applied magnetic field direction and the moment where a spectrum is acquired by the CCD, all the process is controlled by the same script, also in Python, using Py4Syn, which operates the power supplier connected to the magnetic coil and send the electric pulse to trigger the CCD acquisition.

At the end of a cycle, that is, every time a set of eight spectra are acquired, the XMCD is calculated. Figure 1 illustrates the data flow across the main data processing boxes.

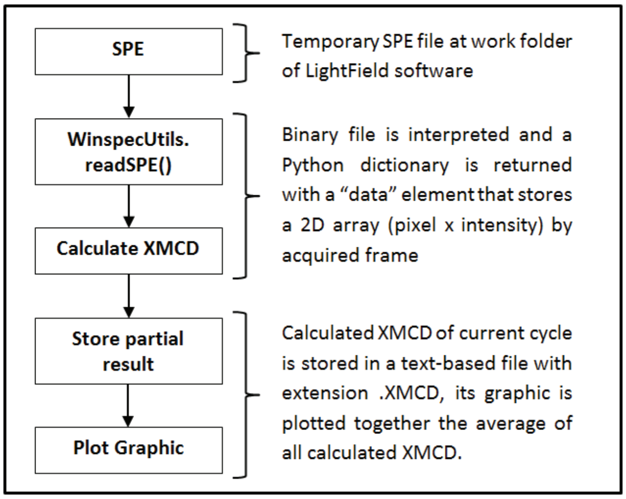

SPE
WinspecUtils.
readSPE()
Calculate XMCD
Store partial
result
Plot Graphic
Temporary SPE file at work folder of LightField software
Binary file is interpreted and a Python dictionary is returned with a "data" element that stores a 2D array (pixel x intensity) by acquired frame
Calculated XMCD of current cycle is stored in a text-based file with extension .XMCD, its graphic is plotted together the average of all calculated XMCD.

Figure 1: Data Flow to Calculate and Plot XMCD.

WinspecUtils.readSPE()

Based on the well formatted SPE binary file, and Princeton Instruments definition of each type and length of data on its header, readSPE() method of WinspecUtils.py script unpack each piece of data from it and return a Python dictionary with this structure:

```
spedict = {  'data':[],
             'IGAIN': pimaxGain,
             'EXPOSURE': exp_sec,
             'SPEFNAME': spefilename,
             'OBSDATE': date,
             'CHIPTEMP': detectorTemperature,
             'COMMENTS': comments,
             'XCALIB': xcalib,
             'ACCUMULATIONS': accumulations  }
```

The most important to us at this moment is the 'data' element. It will store in the first dimension a set of spectra, or frames, and of each spectra a 2D matrix of pixel x intensity.

[†] douglas.beniz@lnls.br

HIGH LEVEL SOFTWARE FOR THE COMMISSIONING OF THE EUROPEAN XFEL

Lars Fröhlich*, Christopher Behrens, Bolko Beutner, Maria Elena Castro-Carballo,
Winfried Decking, Olaf Hensler, Raimund Kammering, Torsten Limberg, Sascha Meykopff,
Matthias Scholz, Jan Szczesny, Josef Wilgen, DESY, Hamburg, Germany
Enrico Allaria, Giuseppe Penco, Elettra–Sincrotrone Trieste, Basovizza, Italy

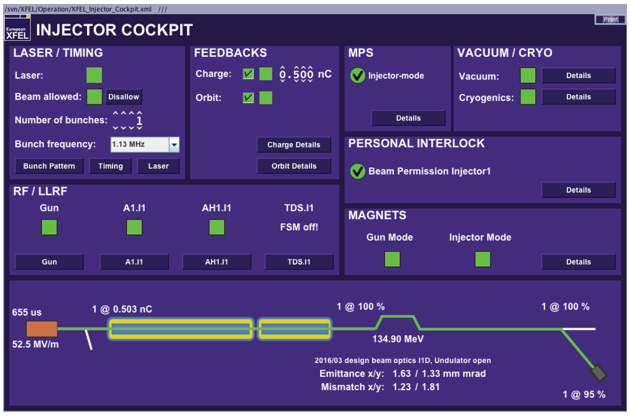
Abstract

The European X-Ray Free-Electron Laser (XFEL) will generate extremely short and intense X-ray flashes from the electron beam of a 2.1 km long superconducting linear accelerator. The commissioning and operation of the accelerator relies heavily on high level software for the automatization of measurements and procedures. The paper gives an overview of the ongoing work and highlights some new measurement techniques.

INTRODUCTION

The European X-ray Free-Electron Laser (XFEL) is a research facility that has been constructed in collaboration between the European XFEL Facility GmbH[1] and DESY[2] in Hamburg, Germany [1–4]. The main component of the facility is a superconducting linear accelerator (linac) that delivers an electron beam with particle energies up to 17.5 GeV and average beam power up to ~600 kW into several long undulator sections. In these sections, the electrons generate extremely brilliant X-ray pulses at wavelengths down to 0.05 nm. These light pulses are distributed to several beamlines and end-stations for photon science experiments. Extensive infrastructure, including a cryogenic plant, has to be operated to allow the XFEL to work. An overview of the control system architecture for the entire facility is given in [5]. The 130 MeV injector has already been fully commissioned with beam, and the commissioning of the entire machine will start soon.

The XFEL is a system of considerable complexity, and operating it smoothly requires a high degree of automatization. We therefore aim to offer high level abstractions for all important machine parameters and to develop user friendly tools for typical physical and technical tasks in the control room. This paper gives a brief overview of our evolving control system landscape, reports on our experience from the commissioning of the injector, and highlights a few applications that have enabled us to perform unprecedented measurements of the beam emittance across the bunch train with a new measurement techniques.



/svn/XFEL/Operation/XFEL_Injector_Cockpit.xml ///

INJECTOR COCKPIT

LASER / TIMING
Laser: 🟩
Beam allowed: 🟩
Number of bunches: 1
Bunch frequency: 1.13 MHz

FEEDBACKS
Charge: ☑ 🟩 0.500 nC
Orbit: ☑ 🟩

MPS
✓ Injector-mode

VACUUM / CRYO
Vacuum: 🟩
Cryogenics: 🟩

PERSONAL INTERLOCK
✓ Beam Permission Injector1

RF / LLRF
Gun 🟩
A1.I1 🟩
AH1.I1 🟩
TDS.I1 FSM off!

MAGNETS
Gun Mode 🟩
Injector Mode 🟩

655 us
52.5 MV/m
1 @ 0.503 nC
1 @ 100 %
134.90 MeV
1 @ 100 %
2016/03 design beam optics I1D, Undulator open
Emittance x/y: 1.63 / 1.33 mm mrad
Mismatch x/y: 1.23 / 1.81
1 @ 95 %

Figure 1: Screenshot of the “injector cockpit” *jddd* panel.

FUNDAMENTALS

No less than four main control system protocols are in use at the XFEL: DOOCS [6, 7], EPICS [8, 9], TINE [10, 11], and Karabo [12]. A lot of work has already been invested to improve the interoperability of the first three protocols [13], so that the problem of network communication across protocol boundaries is less daunting today than it was several years ago. Most of the remaining cross-protocol effort is focused on creating an interface with Karabo.

We are operating a central *configuration database* as a network service. It stores a complete list of beamline components and associated information such as calibration data. This helps to avoid inconsistencies in the configuration of distributed servers [14].

In large parts, XFEL controls follow a “rich server–thin client” philosophy. We are trying to implement high level abstractions and advanced data processing at the server level so that many user interfaces do not need to be programmed but can be configured with our *jddd* [15, 16] user interface builder (Fig. 1). Even applications requiring more complex interaction or more advanced plotting capabilities become simpler to write and easier to maintain with this approach.

Almost all of the control system servers for the machine are written in C++ and deployed on Linux systems. We have created a multitude of libraries to facilitate easy access to various accelerator components and to the central database, for numerical and image analysis tasks, for the calculation of optical functions, and for particle tracking. GUI applications are deployed on MacOS, Windows, and Linux desktops and written in Matlab, Java, or Python. Toolboxes and libraries for these languages are available as well.

[*] lars.froehlich@desy.de

[1] European X-Ray Free-Electron Laser Facility GmbH, Albert-Einstein-Ring 19, 22761 Hamburg, Germany

[2] Deutsches Elektronen-Synchrotron DESY, Notkestraße 85, 22607 Hamburg, Germany

AUTOMATION OF THE MAGNETIC FIELD MEASUREMENTS OF THE AIR COILS BY MEANS OF THE MOVING WIRE SYSTEM

M. Yakopov[†], S. Abeghyan, S. Karabekyan, J. Pflueger, European XFEL Hamburg, Germany
Z. Zhao USTC/NSRL, Hefei, Anhui, People's Republic of China

Abstract

To ensure self-amplified spontaneous emission process the undulators, of a used for this, must not deflect the electron beam from its orbit. The possible deflection of the electron beam, introduced by undulator, must be corrected by means of two air coils. These air coils which are installed from both sides of the undulator, must eliminate not only the deflection angle, but also the displacement between electron beam trajectory and the orbit. For European XFEL 182 air coils are necessary. The magnetic field of each air coil was measured, to determine a conversion coefficient used by the control system. To minimize the measurement time an automated procedure has been developed and implemented. This paper describes the measurement setup, technical implementation method and automation procedure.

INTRODUCTION

At both ends of each undulator, used by European XFEL [1], gap dependent steering errors may occur due to unavoidable magnet imperfections. These errors lead to the errors in the 1st and the 2nd Field Integral. One of the important tasks during the commissioning of the undulators in the XFEL magnetic hutches is the exact measurement of the gap dependency of upstream and downstream kicks as a function of the gap. For the compensation of these errors an air coil corrector is foreseen on either end. The serial production of the air coils has been successfully done by an industrial supplier. There are 91 air coils with 12 mm gap and 91 air coils with 32 mm gap, due to geometry of the vacuum chamber, where they supposed to be installed. In order to precisely operate air coils, both software and hardware solution have been integrated into the undulator control system [2], which is based on the Beckhoff automation technology. The software requires the conversion coefficients between currents and steering strengths of the different coils. For precise determination of the conversion coefficients of all air coils an automated test stand has been built based on the moving wire (MW) technique [3]. The conversion coefficients obtained in this way have been implemented in the undulator control system. In addition the distribution of the first field integral across the air coil gap, a crosstalk of the vertical (*By*) component of produced magnetic field to the horizontal (*Bz*) component and vice versa have been measured. These measurements were a part of the commissioning procedure of the air coils.

AUTOMATED MEASUREMENT SETUP

Description of the Setup

The centerpiece of the measurement setup is the MW measurement system [4]. It is an automated system aimed to the measurements. A LabVIEW program is used to control the measurements, perform the data analysis and manage the data storage. The MW system is operated by a XPS – Newport controller which communicates with the control PC through the TCP/IP network protocol. The measurement setup also includes a water cooling system for the air coils. The functional diagram of the test setup is shown in Fig. 1. A hardware controller has been created for switching ON and OFF the power supplies of the air coils. The controller was operated by XPS via a GPIO interface. The current setting of the two power supplies has been adjusted manually to 1 A and controlled by Keithley multimeters with ±5 μA accuracy for both *By* and *Bz* magnetic field components.

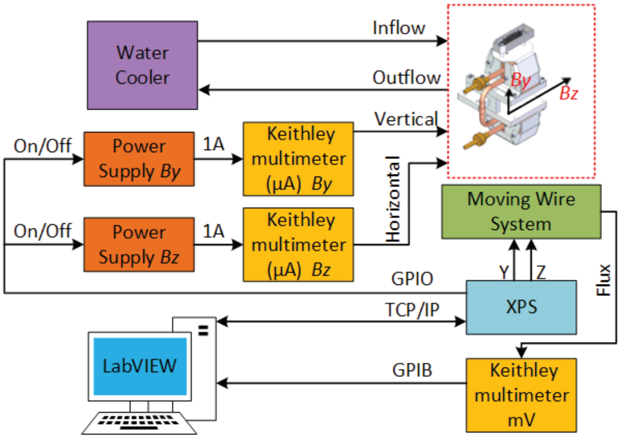


Figure 1: Functional diagram of the magnetic field measurement test setup of the air coils.

Functionality of the setup

The distribution of the first field integral of *By* and *Bz* components across the gap of the air coils have been measured using the developed Labview software. Before starting the measurements, the air coils have been installed on a specially designed holder, and the front edge of the wire has been positioned at the geometrical middle point of the air coil gap, in X and Y directions. Figure 2 shows an air coil installed on the MW bench ready for measurements.

[†] mikhail.yakopov@xfel.eu

ISBN 978-3-95450-189-2

DIAGNOSTICS AT JINR LHEP PHOTOGUN BENCH

M. A. Nozdrin*, N. I. Balalykin, V. F. Minashkin, G. D. Shirkov, JINR, Dubna, Russia
S. Weisse, DESY, Zeuthen, Germany

Abstract

The photoinjector electron beam quality strongly depends on the laser driver beam quality. For laser beam diagnostics, a "virtual cathode" system was realized at the photogun bench located in the Veksler and Baldin Laboratory of High Energy Physics (LHEP) of the Joint Institute for Nuclear Research (JINR). The system allows one to image laser beam profile at the cathode. For imaging, the AVINE software suite developed in DESY Zeuthen is used. Equipment for emittance measurement using the slit method was installed. The original emittance calculation software EmCa was created and tested with laser beam.

INTRODUCTION

Photogun bench of JINR LHEP [1, 2] aims to develop and improve "transmissive" photocathodes to increase quantum efficiency, lifetime and to decrease vacuum requirements. "Transmissive" photocathode is the development of the "Hollow" photocathode conception developed at JINR [3, 4]. It consists of micron-sized metal mesh or quartz/sapphire plate with thin-film coating: either metal or semiconductor.

Beam diagnostics, both electron and laser, is important in order to investigate cathode characteristics. To accomplish this, a set of diagnostics subsystems is installed at the bench.

BENCH EQUIPMENT

Main bench elements (Fig. 1) are DC photogun with the maximum voltage of 30 kV, focusing magnet with correction windings, diagnostics and driver laser. Bench beamline vacuum is less than 10^{-8} torr. A faraday cup is used for electron bunch charge measurement. Laser pulse energy is measured by an Ophir Nova II power/energy meter equipped with PE25 pyroelectric sensor.

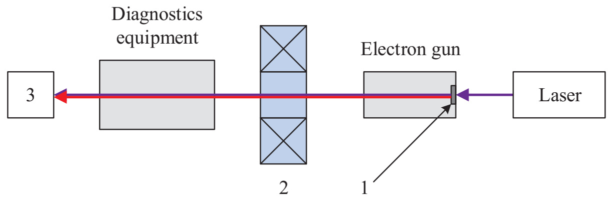


Figure 1: Photogun bench scheme: 1 - "transmissive" photocathode; 2 - focusing magnet with correction windings; 3 - beam dump.

Lasers

The main laser driver currently is LS-2134 by LOTIS TII (Minsk, Belarus). Also LS-2151 model was used, LS-2132UTF model is being commissioned now. As next step, the new—photoinjector—bench is being constructed. It will use the unique laser system [5, 6] by IAP RAS (Nizhny Novgorod, Russia) and will have a maximum electron energy of 400 keV. Key parameters of the aforementioned laser systems are listed in Table 1.

Emittance Measurement Equipment

Emittance is one of the main particle beam parameters, its measurement is one of the key photocathode investigation tasks. Therefore, a slit emittance measurement system is being developed at the bench. It consists of the following equipment:

- slit mask, 1 mm thick tungsten plate with 9 50 µm slits located at 3 mm distance from each other;
- vacuum chamber with 2-position pneumatic actuator where the slit mask is installed;
- scintillator screen;
- high sensitivity CCD camera Prosilica GC1380 for beam imaging;
- compressor with 7 atm maximum pressure for pneumatic actuator operation.

Imaging Equipment

A high sensitivity Gigabit Ethernet CCD camera AVT Prosilica GC1380 [7], together with a lens Kowa LM50JCM, is used at the bench. Camera parameters are given in Table 2. Two external trigger inputs are provided: isolated (trigger latency 5 µs, jitter ±0.5 µs) and non-isolated (trigger latency 3.7 µs, jitter ±20 ns). As potential differences are not expected on the local bench setup and due to absence of powerful noise sources, the non-isolated channel is used.

Table 1: Lasers Parameters

Parameter [units]	LS-2134	LS-2151	LS-2132	IAP RAS
Wavelength [nm]	266	266	266	260–266
Micropulse length [ps]	15,000	75	5,000	8–12
Micropulse energy [µJ]	15,000	5,000	30,000	1,85
Macropulse length [µs]	—	—	—	800
Micropulses in macropulse	—	—	—	8 000
Macropulse repetition rate [Hz]	—	—	—	10

[*] nozdrin@jinr.ru

ISBN 978-3-95450-189-2

ORBITKORREKTUR, A JAVA CLIENT FOR TRANSVERSE ORBIT CORRECTION IN PETRA-III

G. K . Sahoo#, F. Brinker, F. Wedtstein, DESY, Hamburg, Germany

Abstract

PETRA-III is a 3rd generation synchrotron light source dedicated to users at 14 beam lines with 30 instruments in the Max von Laue Hall and 10 beamlines in newly constructed Extensions in North and East which are under installations. PETRA-III is operated with several filling modes such as 60, 480 and 960 bunches with 100 mA or 40 bunches with 90 mA at an electron beam energy of 6 GeV. Transverse orbits are corrected to a reference orbit, which is based on BBA measurements taking into account requirements of the beamlines and results of the dispersion correction. Histograms from 244 BPMs are displayed by means of this Java Client OrbitKorrektur. The orbits may be corrected using the Effective Correction Method with a few correctors or can be corrected using the SVD Method with proper singular Eigen values using theoretical response matrices from an Optic Server of used Optics. In this application additional features are implemented for better observation and analysis of orbits. Furthermore it can also be used to do many additional jobs, such as showing corrector set currents, loading golden and reference orbits, local bumps, displaying the first turn data during commissioning etc.

INTRODUCTION

PETRA-III [1] is a 3rd generation synchrotron light source commissioned with electron beam energy of 6 GeV and 100 mA stored current at betatron tune values of 37.12 and 30.28 in 2009. It has a large circumference of 2304 m which is considerably larger than any existing light source around the world. The machine consists of arcs and several straight sections. The so-called long straights which have a length of 108 m are located in the North, East, South and West. In between two long straights are arcs and a short straight section with a length of 64.8 m. The magnetic structure is a simple FODO lattice. The part that extends from the middle of one long straight to the middle of the adjacent short straight is the basic building block of the machine. Since this section is just one eighths of the machine it is called an octant. The magnetic arrangement of one octant is mirror reflected at the middle of the short straight. Electrons are injected in the South-East (SE) and travel clockwise around the machine. The octant extending from North-East to East was modified breaking the fourfold symmetry and is called as new octant (built in the building *Max von Laue Hall*). It consists of nine Double Bend Achromat cells (DBA). Eight of them provide space for one 5 m or two 2 m long insertion devices (ID).

The two 2 m IDs are inclined towards each other by 5 mrad. This scheme allows operating two independently tunable undulators in a single straight section with beam paths sufficiently separated for individual beam line optics. The ninth straight section is suitable for the installation of an insertion device up to a length of 20 m. The horizontal beam emittance is 1 nm.rad while a coupling of 1% amounts to a vertical emittance of 10 pm.rad. The machine is dedicated to users for experiments from 14 beam lines with 30 end-stations in the Max von Laue Hall. The storage ring has been further modified in 2014 in the North and East quadrants to incorporate 10 new beam lines including a Super Luminescent beam line from dipole radiation. PETRA operates with several filling modes, such as 480 and 960 bunches with a beam current of 100 mA or in the Timing mode with 40 or 60 bunches with 90 – 100 mA. The new light source needs careful magnet alignments, orbit stabilization and closed orbit and dispersion corrections. A prerequisite for the small vertical emittance are tight tolerances on the spurious vertical dispersion that is mainly created by the misalignment of the sextupoles. The closed orbit deviations result from field errors arising from magnetic element positioning errors. The most severe effects come from misalignment of quadrupole magnets, where the resulting dipole field is proportional to both gradient and alignment errors. The closed orbit distortions have been simulated with the alignment tolerances and field errors. Since the integrated quadrupoles strength in the new octant is two to three times higher than in the old octants the alignment requirements are accordingly tighter.

For the initial design condition and without magnet errors the particles move along the so-called "Design Closed Orbit" with designed energy. Generally this orbit is centred in the magnets and Beam Position Monitors (BPMs). The offsets of the BPMs adjacent to quadrupoles are measured with a Beam Based alignment (BBA) to allow the beam passing through the centre of the quadrupoles. Variations of the initial conditions, alignments and magnet field errors lead to a deviation from the "Design Closed Orbit" and oscillate around a distorted closed orbit. Off energy particles take shorter or longer paths around this closed orbit as per the deviation from the designed energy. Transverse orbits are corrected to a reference orbit, which is based on BBA measurements taking into account requirements of the user beamlines and results of the dispersion correction.

# Gajendra.Kumar.Sahoo@desy.de

ISBN 978-3-95450-189-2

FAST ORBIT FEEDBACK AT DELTA*

P. Hartmann, A. Althaus, D. Rohde, D. Schirmer, G. Schuenemann†, P. Towalski†, T. Weis, S. Khan
DELTA, TU Dortmund University, Germany

Abstract

At the electron storage ring DELTA, studies of a fast orbit feedback integrating Libera Electron and Bergoz MX-BPM electronics were conducted. A review of the project and its results is given.

INTRODUCTION

Global orbit feedback systems that correct orbit distortion in a frequency range between 1 and 500 Hz have become a standard at modern synchrotron light sources [1]. They are typically based on FPGA-driven beam position monitoring (BPM) electronics as for example the I-Tech Libera Brilliance [2].

At the 1.5 GeV synchrotron light source DELTA [3] at the TU Dortmund University, the storage ring as well as the booster synchrotron are mainly equipped with Bergoz MX-Beam Position Monitors [4]. At strategic positions in the storage ring, ten MX-BPMs have been replaced by I-Tech Libera Electron/Brilliance BPM electronics which are capable of measuring the beam position turn-by-turn. The Libera BPM electronics are shipped with a digital interface that intergrates seemlessly into DELTA's control system EPICS [5] while the MX-BPMs provide the measured beam position as an analog voltage that is at present digitized by 12-bit analog-to-digital converters (ADC) and fed over CAN bus into the EPICS control system of the accelerator. A slow control system based global orbit feedback (latency > 1 s) compensates for slow beam motion caused by insertion device drifts and thermal drifts.

Fast beam motion

Beam motion at frequencies above 1 Hz has been identified by turn-by-turn measurements from Libera BPM Electronics (see Fig. 1). While high-frequency distortion is mainly caused by the line frequency and its harmonics, distortions at 5.3 Hz, 10.6 Hz and 12.5 Hz were identified as girder resonances using acceleration sensors on top of magnets and on the floor (see Fig. 2) [6].

LOCAL ORBIT FEEDBACK

Suppression of fast beam motion has been investigated in a first step with a fast local orbit feedback based on two I-Tech Libera BPMs and four correctors (see Figs. 3 to 5) [6]. The Libera devices use a code developed at the Diamond Light Source called the Diamond Communication Controller (DCC) [7]. This code runs on a Virtex-II-Pro FPGA contained in the Libera BPM electronics. It distributes measured BPM position data at a rate of 10 kHz between all participants linked to a dedicated optical fiber network.

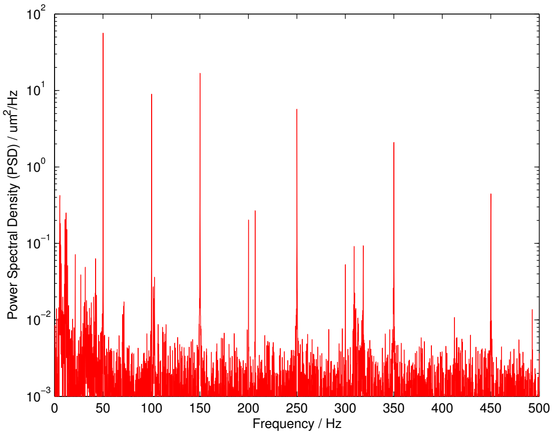


Figure 1: Vertical beam motion at BPM42.

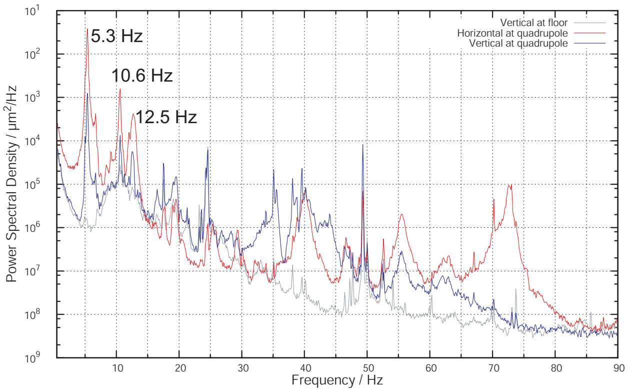


Figure 2: Magnet motion due to girder resonances.

In order to compute orbit corrections and to apply them to fast power supplies and corrector magnets, we implemented the DCC on a Virtex-II-Pro FPGA mounted on the Digilent XUP-Virtex II Pro evaluation board (XUPV2P) [8]. This board was connected to the fiber network and was thus able to receive BPM data at a 10 kHz data rate. In addition, we developed software for orbit correction based on a PI regulator and a driver for fast magnet power supplies (see Fig. 4).

With an optimum set of parameters we were able to reach an upper frequency limit for orbit correction slightly above 350 Hz (see Fig. 5). Distortion at higher frequencies is amplified instead of being damped. This frequency limit is attributed to latency of the beam measurement and regulation process. Beam motion at 50 Hz was damped to -20 dB of the original amplitude, while at 150 Hz the damping still was -10 dB.

[*] Work supported by BMBF, FKZ 05P09PERB5 and Forschungszentrum Jülich, contract no. COSY/FAIR-114

[†] Affiliation changed

ISBN 978-3-95450-189-2

HIGH-LEVEL APPLICATION DEVELOPMENT AND PRODUCTION INFRASTRUCTURE AT TRIUMF

E. Tikhomolov, Y. Bylinskiy, A. C. Morton[*], T. Planche, T. Tateyama, J. Lee, P. Jung[†]
TRIUMF, Vancouver, BC, Canada

Abstract

TRIUMF users and operators use a number of high-level applications (HLAs) written in different languages, with complicated graphical user interfaces, to carry out tasks related to delivering ion beams with required characteristics and to process data from TRIUMF's EPICS-based and legacy cyclotron control systems. Some applications have been developed by the EPICS community, and some at TRIUMF. These applications run on different production computers and are developed on different machines. This model no longer satisfies TRIUMF's needs because of the growing number of applications, the long times required for data processing on current machines, the lack of real- time visualization of beam properties and so on. New infrastructure for HLA development has been implemented to address these issues and is working reliably with room for further expansion.

MOTIVATIONS

TRIUMF doesn't have a dedicated group of software developers tasked with immediate response to issues that arise during day-to-day beam delivery. Such issues are resolved by operators and physicists themselves. Thus, a flexible and simple ("user-friendly") software development environment was the main request when it was decided to set up High-Level Application (HLA) development and production infrastructure. At the same time, developers come with different experience, backgrounds and their own favorite tools for development. Thus, the HLA environment should provide some "default set" of tools which are rather common and compatible with other widely-used tools.

HLA SERVERS SET UP AND THEIR DIFFERENT ROLES

The projects that are under development at TRIUMF usually have three components: the code itself (in a number of programming languages), documentation and information, and input/output data. Thus, the common project can be represented as a point in a "three-dimensional space" where each component defines an axis, as illustrated in Fig. 1.

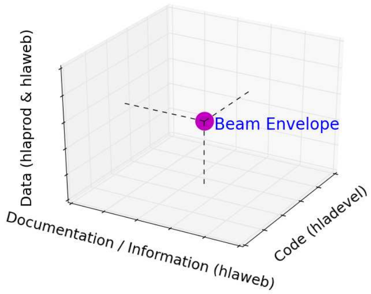


Figure 1: Common HLA project in "3D space".

Some projects are just data projects which process experimental data (from TRIUMF's 520 MeV Cyclotron and Isotope Separator and Accelerator facility, or ISAC) for use with third-party applications. Each component has a corresponding location and directory structure: the code is developed on a development server (hladevel), input/output data are saved on production servers (hlaprod and hlaweb), and documentation and information are maintained on a web server (hlaweb [1]).

To fulfil these tasks rack-mounted Dell servers were chosen [2]. All three servers have identical directories and software. This reduces maintenance time and allows one to easily switch from one server to another in the event of hardware failure. This approach also allows the maintainer to easily expand the infrastructure by cloning the existing servers, and redistributing the load. The differences between the servers are defined only by their different roles. Figure 2 shows these roles and the interactions between servers and user machines and control systems. The HLA development server hladevel and production server hlaprod are accessible only from the TRIUMF network. The web server hlaweb is accessible via the Internet and users can ssh to their accounts. Users have scratch directories (public_html) which may be used e.g. for presenting results at meetings. To reduce security threats it is not possible to connect to any TRIUMF computers from user accounts on hlaweb.

The data and user's scratch directories on the development server, hladevel, are synchronized (using rsync) with hlaweb every 4 hours. Data directories from hlaprod are also synchronized to hladevel. This duplication of data helps to avoid data loss. Nightly backups are done only on hladevel both on local USB drives (for fast restoration) and to a remote location (using TRIUMF's Amanda system [3]).

For version control Git [4] is used. Public and private remote repositories are located on hlaweb and hladevel, respectively. For projects which are expected to involve large number of inter-lab developers Github [5] may also be used.

CentOS 7 was chosen as the operating system for the HLA servers. This is already used at TRIUMF and supported by TRIUMF's Core Networking and Computing Group. To reduce the time spent on system administration the standard tools and applications provided with CentOS 7, at installa-

[*] Present affiliation: FRIB. MSU, East Lansing, MI, USA
[†] Present affiliation: Waterloo University, Waterloo, ON, Canada

STATUS OF THE NSLS-II LLRF SYSTEM*

C. Marques†, J. Rose, N. Towne, B. Holub, F. Gao, G. Wang
NSLS-II, BNL, Upton, NY 11973, U.S.A

Abstract

The NSLS-II RF system uses an in-house FPGA based low level RF (LLRF) solution called the Cavity Field Controller (CFC). The CFC directs the amplitude and phase for the high power RF and directly influences beam acceleration and stability. In this paper we discuss a logically embedded network analyzer in situ with the digital feedback loop controlled via a MATLAB or EPICS interface. The embedded NA was used to evaluate the RF feedback stability and influence of the feedback parameters on the beam. We will also discuss diagnostics tools to investigate longitudinal beam dynamics and other functionality embedded into the FPGA fabric. Future development of the CFC implementation and hardware upgrades will also be discussed.

INTRODUCTION

The Field Programmable Gate Array (FPGA) based CFC was developed as a common hardware platform for all NSLS-II RF systems. The reconfigurable logic on the FPGA allows the CFC to run in multiple modes of operation. For example, open loop feedforward (FF), closed loop feedback (FB) and a combination of the two. In addition to operating modes the CFC is able to run diagnostics, cavity tuner loop adjustments, data buffering, event handling including interlocks and real-time data acquisition simultaneously. The CFC includes eight 500 MHz RF inputs and one 500 MHz RF output along with multiple IO which can be seen in Figure 1. More information about the digital controller can be found elsewhere [1].

CFC FUNCTIONS

One of the primary functions of the CFC is a feedback control loop of the cavity field. A simplified schematic of the feedback loop and other CFC logic can be seen in Figure 2. The feedback loop includes digitized vector signal detection provided by dual 14-Bit ADC LTC2299s at 80 MHz (40 MHz/channel) and can be seen in the schematic by the $I, Q, -I, -Q$ sequence just after the cavity pickup. The $I, Q, -I, -Q$ notation comes from the fact that the 50 MHz IF is sampled at half the clock frequency or 40 MHz. The digitized sequence is then corrected for feedback path latency of approximately 1.1 ms using a phase rotation provided by the $Kp(I)$, $Kp(Q)$ multipliers and where the total Kp magnitude is $|Kp| = \sqrt{Kp(I)^2 + Kp(Q)^2}$. Digital signal processing using a host-front end is used to calculate and properly adjust the loop phase rotation. The summed I and Q pair is multiplied by the integral gain factor Ki and subsequently summed. It is worth noting that the integral gain Ki is dependent on the proportional gain Kp and hence so is the corner frequency or bandwidth of the feedback loop. The summed error signal produced in the integral leg of the feedback loop not only provides for error correction but also synthesizes the output drive signal for the CFC.

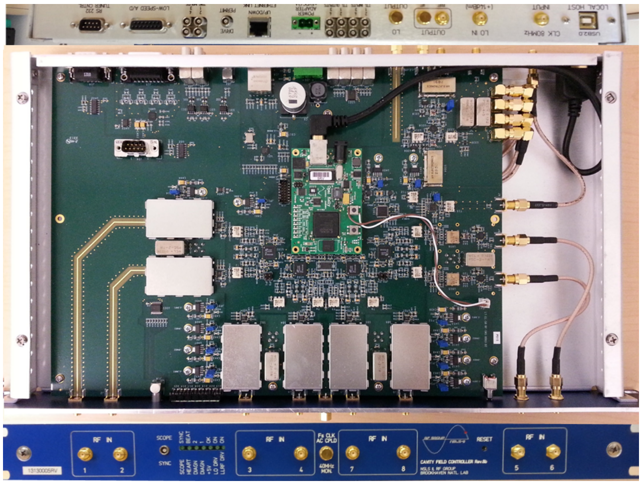


Figure 1: Picture of the CFC. The top and bottom of the picture shows the different IO on the back and front panels respectively.

Network Analyzer

The logically embedded vector network analyzer produces a direct digital synthesized (DDS) stimulus which is summed with the output drive of the controller. The network analyzer module performs a fourier analysis on the CFC readbacks to produce response functions of those readbacks normalized to the DDS output. Much like an S_{21} measurement on a conventional network analyzer except in this case we have several inputs including the feedback loop and longitudinal beam motion from a summed beam position monitor (BPM). Using a narrow bandpass filter of 540 kHz with a center frequency of 499.68 MHz, we eliminate any problematic signals including the charge induced by an empty bunch train from the storage ring fill pattern and we are able to measure a unambiguous beam response. Figure 3 shows the amplitude response function of the cavity field and BPM. The figure shows that for a fixed integral gain Ki and DDS amplitude, the proportional gain Kp influences both the beam and cavity response to the stimulus. For higher Kp gain the effect of the beam from perturbations is decreased and suggests higher stability and lifetime. This effect is also shown by the narrowing of the peak at the synchrotron frequency for increasing Kp values. Figure 3 also shows

* Work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-SC0012704.
† cmarques@bnl.gov

ISBN 978-3-95450-189-2

OVERVIEW OF SOME FEEDBACK- & CONTROL SYSTEMS AT SYNCHROTRON SOLEIL

C. Engblom[*], F. Alves, F. Blache, D. Corruble, A. Dawiec, M. Diop, N. Hubert,
S. Kubsky, F. Langlois, P. Marchand, N. Jobert, G. Renaud, Y.M. Abiven
Synchrotron SOLEIL, Saint-Aubin, France
T. Stankevic, MAXIV, Lund, Sweden

Abstract

This paper gives an overview of some feedback & control systems at Synchrotron SOLEIL that are in use or in development today. Beam stability is crucial and adressed in all SOLEIL aspects; Fast Orbit Feedback is a multi-input multi-output control system made to stabilize beam position perturbations in the low- & high frequency band. In addition, active RF cavities are used to maintain stable beam energy & spread as well as keeping electron density even throughout the storage ring. Beam stability also comes from feedforward non-linear control in particle trajectory compensation on both sides of electromagnetic undulators. On some beamlines, multi-actuator piezos or pneumatics are used to regulate photon flux to keep within detector operating range; a method to maximize the photon flux while still keeping below detector damage thresholds. Currently in development & at the sample stage level, the Nanoprobe Project collaboration (MAXIV & SOLEIL) focuses on sample stabilization during step- & fly- scans which is realized through multi-axis nano-positioning with high- & low- frequency closed-loop control implementing interferometer feedback &/or compensation tables.

INTRODUCTION

For the past couple of years Synchrotron SOLEIL has, in a collaboration project with MAXIV, been constructing an endstation prototype capable of 2D- and 3D tomography scans on the nanometric scale. Such a project not only relies on passive thermal stability and vibration but also on feedback systems. These systems are not only crucial in the endstation setup but also relies on the stability provided by a series of feedback systems that resides in the storage ring, entry and exits of insertion devices, radiofrequency systems and photon flux regulating devices to provide stable photon beam positioning and flux to beamline endstations. In long beamlines, such as Nanoscopium [1] in Synchrotron SOLEIL or NanoMax [2] in MAXIV, endstations particularly specialize in scanning x-ray microscopy on the nano-scale.

As such, this paper provides an overview of some of the existing feedback systems in Synchrotron SOLEIL with a focus on the progress of the Nanoprobe project.

BEAM ORBIT STABILITY SYSTEM

Stable photon flux to the beamlines is in part provided by ensuring electron beam position and angle stability in the storage ring. For this reason the Global Orbit Feedback System (GOFB) is necessary against environment perturbations in the long term (hours to a day, ex: thermal effects, sun & moon tides), medium term (seconds to minutes, ex: moving crane, insertion devices), and short term (less than a second, ex: booster cycling operations, ground vibrations) [3]. The frequency spectrum of the noise at SOLEIL Synchrotron has been shown to reside in the range from DC→150 Hz [3].

The GOFB correction algorithm is based on Singular Value Decomposition (SVD) where an inverse response matrix, R^{-1}, together with the orbit error, ΔU_{BPMi}, is used to calculate the correction currents, ΔI_{corrj} [3]. Multiplication is shown in Eq. 1, the matrix dimensions of R^{-1} are here denoted as $M \times N$ and directly corresponds to the number of Beam Position Monitors, N, and number of current actuators, M.

$$\begin{bmatrix} \Delta I_{corr\ j} \\ \Delta I_{corr\ j+1} \\ \vdots \\ \Delta I_{corr\ j+M} \end{bmatrix} = R^{-1} \cdot \begin{bmatrix} \Delta U_{BPM\ i} \\ \Delta U_{BPM\ i+1} \\ \vdots \\ \Delta U_{BPM\ i+N} \end{bmatrix} \qquad (1)$$

The GOFB is divided into two systems [3]:

- The Slow Orbit Feedback (SOFB), with correction rate and bandwidth limited to 0.1 Hz
- The Fast Orbit Feedback (FOFB), with correction rate to 10 kHz and efficient up to a few hundred Hz.

The two systems contain their own sets of correctors and can run inpendently of each other. This can cause interference if their control-frequencies are overlapping which is why there are three approaches that can be used [3]:

1. 'Deadband' method (Fig. 1), a frequency deadband is introduced between the two systems guaranteeing complete independency from each other. The issue with this method is that the deadband needs to be sufficiently large and if there are components creating disturbances inside the deadband spectrum they can't be corrected for.
2. 'FOFB only' method (Fig. 2), run the FOFB system alone for all frequencies down to DC. This approach, due to the weakness of fast correctors, have limits on correction amplitudes and might saturate the correctors.
3. 'FOFB/SOFB interaction' method (Fig. 3), the SOFB corrects low-frequency disturbances while predicting

[*] christer.engblom@synchrotron-soleil.fr

ISBN 978-3-95450-189-2

A FAST, CUSTOM FPGA-BASED SIGNAL PROCESSOR AND ITS APPLICATIONS TO INTRA-TRAIN BEAM STABILISATION

G. B. Christian*, N. Blaskovic Kraljevic, R. Bodenstein, T. Bromwich,
P. N. Burrows, C. Perry, R. Ramjaiwan, J. Roberts[1]
John Adams Institute for Accelerator Science at the University of Oxford, Oxford, UK
[1]also at CERN, Geneva, Switzerland

Abstract

A custom 9-channel feedback controller has been developed for low-latency applications in beam-based stabilisation. Fast 14-bit ADCs and DACs are used for high-resolution signal conversion and a Xilinx Virtex-5 FPGA is used for core high-bandwidth digital computation. The sampling, and fast digital logic, can be clocked in the range 200 to 400 MHz, derived from an external or internal source. A custom data acquisition system, based around LabVIEW, has been developed for real-time control and monitoring at up to 460 kbps transfer rates, and is capable of writing and reading from EPICS data records. Details of the hardware, signal processing, and data acquisition will be presented. Two examples of applications will also be presented: a position and angle bunch-by-bunch feedback system using strip-line beam position monitors to stabilise intra-train positional jitter to below the micron level with a latency less than 154 ns; and a phase feedforward system using RF cavity-based phase monitors to stabilise the downstream rms phase jitter to below 50 fs with a total latency less than the 380 ns beam time-of-flight.

INTRODUCTION

Many modern particle accelerators and future colliders require the generation and preservation of low emittance beams with a high degree of stability. Future electron-positron collider designs, such as the International Linear Collider (ILC) [1] and the Compact Linear Collider (CLIC) [2], call for beam spot sizes of 5 nm and below at the interaction point (IP), in order to maximise the luminosity. In order to achieve the design luminosity, 2 x 10^{34} cm^{-2} s^{-1} in the case of ILC, in the presence of ground motion and facilities noise, a fast feedback is envisaged, operating at the interaction point, to correct the incoming position error of one beam with respect to the other, within the duration of the bunch train. Prototypes of such systems have been developed by the Feedback On Nanosecond Timescales (FONT) group. Initially, a purely analogue system was developed for room temperature RF cavity based linear collider designs, such as the NLC, and achieved a system latency of 23 ns, on a 56 ns duration bunch train at the KEK Accelerator Test Facility (ATF) in 2005 [3]. Following the choice of superconducting RF for the ILC, with ~1000 bunches separated by ~500 ns, a digital IP feedback system prototype was developed, using a custom FPGA-based digital feedback controller. This has been tested and used extensively at ATF, and has been employed in the beam stabilisation efforts at the ungraded ATF extraction line, ATF2. The ‘FONT5’ feedback controller has also found applications in other beam stabilisation systems requiring low latency, for example, the CLIC drive beam phase feedforward demonstration at the CLIC Test Facility (CTF3). Details of the feedback controller, including data acquisition, will be presented as well an overview and key results from the applications above.

FONT5 FEEDBACK CONTROLLER

Figure 1 shows the PCB and front panel of the FONT5 digital feedback controller. The board is based around a Xilinx Virtex-5 FPGA (XC5VLX50T) [4], with a maximum speed of 550 MHz and 2160 Mb integrated block memory. The board has nine analogue input channels using 14-bit ADCs [5], capable of digitising up to 400 MSPS. Only the most significant 13-bits are connected to the FPGA however, in order to reduce routing congestion, and hence ease timing closure in the FPGA fabric. The ADCs have a low-latency (3.5 clock cycles) making them very suitable for fast feedback applications. The nine channels are arranged as three banks of three, with each bank sharing a common ADC clock. Offset DACs are provided to trim the ADC pedestals. The board also features four 14-bit DACs [6], with a maximum conversion speed of 210 MHz and 0.5 clock cycle latency. As for the ADCs, only the upper 13-bits are connected to the FPGA.

An on-board 40 MHz oscillator is provided for clocking slow logic and ancillary functions, as well as a fast comparator for an external system clock, usually in the range 200 to 400 MHz. A fast system clock can either be sourced externally, with an optional PLL-based jitter filter, or synthesised internally using a digital clock manager. Two programmable-level digital inputs are provided, which are normally used for trigger inputs, as well as several buffered and non-buffered I/Os. Communication to the FPGA is made via an RS-232 connection, running at up to 460.8 kbps. 128 7-bit control registers are used to communicate commands and variables to the FPGA, and up to 1024 samples per channel can be stored in Block RAM and transmitted via a UART, alongside read-backs of the control registers and status bytes. ADC data is displayed and saved to file using custom DAQ software written in LabVIEW. This software can also set control registers, and load RAM tables on the FPGA. Data and settings can be published as EPICS process variables, and infor-

[*] glenn.christian@physics.ox.ac.uk

ISBN 978-3-95450-189-2

HARMONY: A GENERIC FPGA BASED SOLUTION FOR FLEXIBLE FEEDBACK SYSTEMS

X. Serra-Gallifa[†], J. Avila-Abellan, M. Broseta, G. Cuni, D. Fernandez-Carreiras, O. Matilla, A. Ruz, ALBA-CELLS Synchrotron, Barcelona, Spain

Abstract

Feedback and complex acquisition systems usually need real-time interaction among instruments with microsecond's time response. These implementations are hard to achieve with processors but feasible using FPGAs. There are some cases, such as synchrotron beamlines, where high flexibility and continuous tuning are also required, but the implementation of multiple full-custom FPGA designs are extremely time-consuming. Harmony is a solution based in FPGA that offers, via high level programming, a unique framework with common time base, data acquisition, storage, real-time processing, data sharing and diagnostic services designed to implement flexible feedback systems. It is based in two interconnected buses: Self-Describing Bus, developed at CERN/GSI under OHWR license, that communicates with Control System; and Harmony Bus which creates a bus framework where different modules can share timestamped data capable of pre-programed events generation. The first version of Harmony is already successfully being used in Em# project which objective is the development of a performant four-channel electrometer.

ELECTROMETER'S DEVELOPMENTS AT ALBA

During the phase design of the beamlines of ALBA the large number of four channel electrometer needs lead Computing Division to the development of a medium performance electrometer oriented to diagnostic applications [1]. The simplicity of integration in the beamline Control System was one of its main goals. The project was a success with more than 40 units installed and a very good performance of the current amplifier. Inherent analog capabilities extended the use of the electrometer to more complex requirements. Soon it was agreed that a new development project to take advantage of the current amplifier performance was needed to improve different aspects [2, 3]. The main points focused on:

- Improving the signal to noise rate increasing the digital resolution up to 18 bits and the sampling rate.
- The possibility to perform current measurements under voltage biased conditions up to 1 kV and also to avoid ground loops.
- Capability of implementing real-time feedback systems.
- Minimizing the obsolescence problems by adopting a flexible and modular architecture based in standards.

HARMONY: A FLEXIBLE FEEDBACK SOLUTION

From the beginning of the project, it was clearly seen that the new electrometer (Em#) should be easily adaptable to future needs of beamlines experimental stations. However, there is always a trade-off between performance and flexibility. Since the Em# is mainly designed to fulfil the requirements of the Beamlines in a Synchrotron (although is applicable in many other environments) a detailed analysis of the latencies needed by the different "players" was carried out. It states that:

- Low current measurements (bellow µA) have inherently due to its high gain stages and parasitic capacitors maximum bandwidths of few kHz.
- Encoder or trigger readouts can reach frequencies of few MHz.
- Actuators and motor reaction times are higher than hundreds of microseconds.
- Data acquisition timestamped in the microsecond range enables the correlation of different instruments' data during the analysis.

Therefore the solution targeted feed-back systems at frequencies up to 10 kHz, and consequently the FPGA technology was chosen among different options including several microprocessors.

Actually, the performance of modern FPGAs is largely sufficient for the required feedback speeds, with clock frequencies allowing hundreds of clock cycles to process inputs and produce outputs. This enabled a FPGA architecture with independent blocks sharing data. It was also considered that the use of a bus for data sharing design would optimize the desired flexibility for multiple applications.

Development in Workgroup

In previous developments at ALBA, the electronics engineers developed the hardware, gateware (FPGA software) and firmware; once everything was tested it was delivered to software engineers who integrate the instrument in the control and data acquisition system and developed the user interface. In order to overcome this issue, this new project incorporated the contributions from the software engineers earlier in the schedule, reducing the number of late requirements to the hardware and working in parallel shortening the whole duration of the project.

This approach involved a clear definition of the project and its interfaces from beginning. A crucial help to

[†] xserra@cells.es

ISBN 978-3-95450-189-2

CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY AT FRIB[*]

M. Konrad[†], D. Maxwell, G. Shen
Facility for Rare Isotope Beams, Michigan State University, East Lansing, MI 48824, USA

Abstract

Development of many software projects at the Facility for Rare Isotope Beams (FRIB) follows an agile development approach. An important part of this practice is to make new software versions available to users frequently to get feedback in a timely manner. Unfortunately building, testing, packaging, and deploying software can be a time consuming and error prone process. We will present the processes and tools we use at FRIB to standardize and automate this process. This includes use of a central code repository, a continuous integration server performing automatic builds and running automatic test, as well as automated software packaging. For each revision of the software in the code repository the continuous delivery pipeline automatically provides a software package that is ready to be released. The decision to deploy this new version of the software into our production environment is the only manual step remaining. The high degree of reproducibility as well as extensive automated tests allow us to release more frequently without jeopardizing the quality of our production systems.

INTRODUCTION

FRIB [1] is a project under cooperative agreement between US Department of Energy and Michigan State University (MSU). It is under construction on the campus of MSU and will be a new national user facility for nuclear physics. Its driver accelerator is designed to accelerate all stable ions to energies >200 MeV/u with beam power on the target up to 400 kW [2]. Commissioning of the front-end is currently underway and the remaining parts of the accelerator are planned to be commissioned over the next two years.

FRIB's controls group strives to support commissioning and operation by rolling out bug fixes and new features as fast as possible. To make this happen we are following principles of agile software development which include iterative, incremental and evolutionary development and a short feedback and adaption cycle. Unfortunately this approach can be slowed down considerably by the fact that building and deploying control-system software can be a complex and error prone process that often requires considerable manual work by experts. In the following we will describe how we speed up the build and deployment process for FRIB's controls software by following continuous integration (CI) and continuous delivery (CD) principles.

CONTINUOUS DELIVERY

The process of building and deploying software generally consists of a series of tasks that can be thought of as a pipeline. Figure 1 shows the steps of a typical software build and deployment process as it has been implemented at FRIB. Each task in this pipeline is carried out after the preceding step has been completed successfully. In the following we will describe each of these steps in detail.

Revision Control

All source code required to build software for FRIB's accelerator control system is stored on a central Git [3] repository server. The repository server is running Atlassian Bitbucket Server [4] which, in addition to basic Git server functionality, provides a web interface, pull requests and branch permissions.

Our Git work flow largely follows the Gitflow [5] approach which requires developers to implement new features or bug fixes on feature branches allowing them to work on their feature without the risk of breaking other developer's build. If however the number of feature branches becomes too high and feature branches live for too long merge conflicts are becoming more likely. To reduce time-consuming conflict resolution we are following the practice of CI which requires feature branches to be merged into a shared mainline frequently. CI principles generally recommend branches to be merged at least once a day. In our experience many controls projects have a rather low rate of change or a very low number of developers making a life time of a few days feasible with an acceptable risk of running into merge conflicts. For critical code we use pull requests as a tool to facilitate code reviews. The goal still remains the same: Code reviews and possibly required rework should be performed timely so that feature branches can be merged into mainline as fast as possible.

A significant amount of FRIB's control system software is developed in collaboration with other laboratories with its source code being tracked in a an upstream repository on the Internet. In this case our CI server mirrors the mainline branch of the upstream repository into a branch in our local repository on a regular basis. Following CI principles, we are merging upstream changes into our own mainline branch as soon as possible to keep merge conflicts with our feature branches to a minimum. In general we are aiming to keep the number of FRIB-specific modifications to a minimum. Instead we prefer to contribute our improvements back to the upstream project. This reduces the risk of merge conflict in our repository and thus reduces the maintenance effort in the long run. At the same time this approach allows us to fix critical bugs in our local repository until they are fixed upstream.

[*] Work supported by the U. S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661

[†] konrad@frib.msu.edu

ISBN 978-3-95450-189-2

EXPERIENCE GAINED DURING THE COMMISSIONING OF THE UNDULATOR CONTROL SYSTEM AT THE EUROPEAN XFEL

S. Karabekyan[†], S. Abeghyan, J. Pflueger, M. Yakopov, European XFEL, Hamburg, Germany

Abstract

The European XFEL is a fourth-generation light source, which will start the operation in spring 2017. Three undulator systems - SASE 1, SASE 2 and SASE 3 - will be used to produce photon beams. For operation of all undulator systems, in total 91 undulators have been produced and commissioned. SASE 1 and SASE 3 undulator systems, consisting of total 56 undulator cells, have been installed and prepared for the operation in the tunnel in spring and summer 2016. SASE 2 will be installed by the end of 2016. This paper describes the commissioning process of the whole undulator control system and reports about the experience gained over the entire duration of undulator control system commissioning.

INTRODUCTION

Commissioning of the undulator control system was a multilayer task and was carried out in several steps. The strategy of commissioning was to test all hardware components at least once before those components would be installed in the tunnel. The same is true for the software development. Each release of software has been tested first using a simulation software and then on the undulator system test setup, before using it for the real undulator system.

COMMISSIONING OF THE HARDWARE

Undulator System Test Setup

An undulator system consists of an array of up to 35 undulator cells installed in a row in the tunnel along the electron beam. It consists of up to 35 undulator segments and intersections. The system is controlled by a central control node (CCN). It is installed in the control room, which is located about 1 km away from the undulator system. CCN communicates with the undulator cells over optical fibers, and the communication between individual cells is implemented using copper Ethernet and EtherCAT cables. Two media converter racks (MCR) installed from both sides of the system are used to convert signals from copper carriers to optical fiber carriers and vice versa. These two MCR racks are necessary for implementation of the redundant ring topology used for control of the undulator system.

It was obvious that all the envisaged components were to be tested before installation in the tunnel. For this purpose, an undulator system test setup was built in the undulator hall (see Fig. 1). The only difference between the real undulator system and test setup was the amount of undulator cells, which was reduced to four in the test setup. This test setup allowed to test the complete hardware components before installation in a tunnel. It also allowed developing the global control system software three years ahead of the installation of the system in the tunnel.

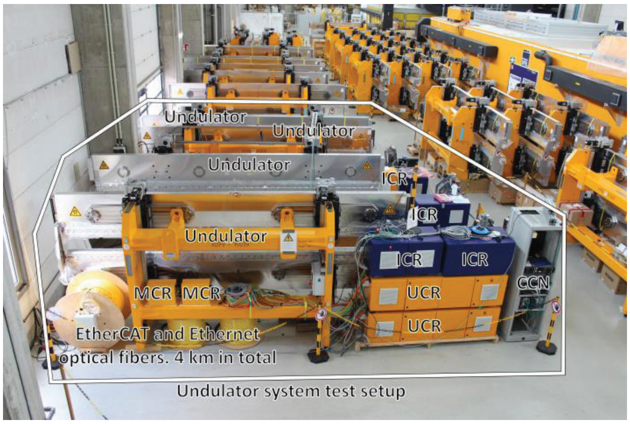

Undulator
Undulator
Undulator
Undulator
ICR
ICR
ICR
ICR
UCR
UCR
CCN
MCR
MCR
EtherCAT and Ethernet optical fibers. 4 km in total
Undulator system test setup

Figure 1: Undulator system test setup in the hall.

Control Components For Undulator Segments

The undulator control system needed to be commissioned before the magnetic commissioning of the undulator. This process is described in details in reference [1]. The undulator control hardware contains components installed on the undulator frame, as well as in the undulator control rack (UCR). The rack is connected to the undulator using a cable bundle. The first operation of the undulators and UCRs took place at the companies producing the undulator frame. It included only the operation of the motors, encoders and limit switches installed on the frame.

For the magnetic commissioning of the undulator it was necessary to bring it into the magnetic measurement hutch. After placing it in the hutch, the undulator had to be connected to the control rack installed on the rooftop of the hutch. Control of all undulators introduced to the hutch can be carried out using the same rack. Nevertheless, it was decided to commission the undulator with the assigned control rack. During this commissioning, a complete set of tests was carried out. This strategy allowed to check the whole undulator control system before installation in the tunnel and helped to discover of hardware-related problems in more relaxed situation on about 5% of the system.

Intersection Control Components

The intersection control components consist of the hardware installed on the Quadrupole Mover (QM) and Phase Shifter (PS) as well as Intersection Control Rack (ICR). The phase shifters have been produced by three different companies. For production and adjustment pur-

[†] suren.karabekyan@xfel.eu

ISBN 978-3-95450-189-2