PCaPAC 2018

International Workshop on Emerging Technologies and Scientific Facilities Controls

October 16-19, 2018
Hsinchu, Taiwan

Topics

- Control Systems
- User Interfaces and Tools
- Hardware Technologies and Component Integration, System Modeling and Automation
- Data Acquisition and Data Storage, Data Analysis
- Infrastructure and Networking, Management of IT Projects, Cyber Security

Key Dates

- Abstract submission：March 15 - May 15, 2018
- Early bird registration：June 1 - August 31, 2018
- Paper Submission：June 17 - October 10, 2018

indico.nsrrc.org.tw/e/pcapac2018

pcapac2018@nsrrc.org.tw

Program Committee

Michael Abbott (Diamond)
Yves-Marie Abiven (SOLEIL)
Arman Arkilic (BNL)
Reinhard Bacher (DESY)
Luciano Catani (INFN)
Jean-Michel Chaize (ESRF)
Matthias Clausen (DESY)
Mark Clift (AS)
Philip Duval (DESY)
Richard Farnsworth (APS)
Pravin Fatnani (RRCAT)
Pete Jemian (ANL)
Norihiko Kamikubota (KEK)
Takashi Kosuge (KEK)
Chang-Hor Kuo (NSRRC)
Ralph Lange (ITER)
Oscar Matilla (ALBA)
Wolfgang Mexner (KIT)
Tim Mooney (APS)
Sarbajit Pal (VECC)
Mark Plesko (Cosylab)
James Rezende Piton (LNLS)
Lucas Sanfelici (LNLS)
Till Straumann (SLAC)
Daniel De Oliveira Tavares (LNLS)
Jonah Weber (LBL)
Glen Wright (CLS)
Dirk Zimoch (PSI)

國家同步輻射研究中心
National Synchrotron Radiation Research Center

Sponsors：MOST 科技部 Ministry of Science and Technology

經濟部國際貿易局
Bureau Of Foreign Trade

PCaPAC 2018

Emerging Technologies and Scientific Facilities Controls

October 16-19, 2018 Hsinchu, Taiwan

國家同步輻射研究中心
National Synchrotron Radiation Research Center

Dear PCaPAC 2018 Participant,

Welcome to PCaPAC 2018 and to Hsinchu!

The 12th International Workshop on Personal Computers and Particle Accelerator Controls (PCaPAC 2018) is taking place in Hsinchu city, Taiwan, from October 16th to 19th, 2018. The conference is organized and hosted by the National Synchrotron Radiation Research Center, a host of JACoW based conference for the first time.

The mission of PCaPAC 2018 is to keep the traditional aspects of PCaPAC as well as to focus on emerging technologies. This is reflected in PCaPAC's new official name "International Workshop on Emerging Technologies and Scientific Facilities Controls". PCaPAC 2018 covers the topics as shown below.

- Control System and Component Integration
- GUI Technologies and Frameworks, User Interfaces and Tools for Operations
- Hardware Technologies and Custom Hardware, Feedback Control and Tuning, Timing and Synchronization, and Functional Safety
- Data Acquisition and Data Storage, Data Analysis
- Infrastructure and Networking Management of IT Projects, Cyber Security

The scientific program includes keynote talks, contributed orals, poster-in-pills and poster sessions. The keynote talks include the topics of "Internet of Things", "Big data/Data mining", and "AI (Deep learning/Machine learning)". Four tutorials are also available before the meeting. Participants could learn updated technologies in a nutshell.

On behalf of the Local Organizing Committee, we welcome around 100 contributors to PCaPAC 2018, to Hsinchu, to Taiwan. I would like to thank you all for your support to the program, the program committee for their suggestions, the JACoW team, and especially each member of the local organizing committee for their devotion and hard work.

I sincerely wish a successful PCaPAC 2018 and all attendees having a joyful stay in Taiwan and a great appetite in Taiwanese cuisine!

Yung-Sen Cheng

Chair of Local Organizing Committee, PCaPAC 2018

Instrumentation and Control Group, Light Source Division, NSRRC

International Program Committee

Michael Abbott (Diamond)
Yves-Marie Abiven (SOLEIL)
Arman Arkilic (BNL)
Reinhard Bacher (DESY), chair
Luciano Catani (INFN)
Jean-Michel Chaize (ESRF)
Matthias Clausen (DESY)
Mark Clift (AS)
Guifré Cuní (ALBA)
Philip Duval (DESY)
Richard Farnsworth (APS)
Pravin Fatnani (RRCAT)
Norihiko Kamikubota (KEK)
Takashi Kosuge (KEK)
Chang-Hor Kuo (NSRRC)
Ralph Lange (ITER)
Wolfgang Mexner (KIT)
Tim Mooney (APS)
Sarbajit Pal (VECC)
Mark Plesko (Cosylab)
James Rezende Piton (LNLS)
Lucas Sanfelici (LNLS)
Till Straumann (SLAC)
Daniel De Oliveira Tavares (LNLS)
Jonah Weber (LBL)
Glen Wright (CLS)

Local Organizing Committee

Yung-Sen Cheng, chair
Kuo-Tung Hsu, co-chair
Chih-Yu Liao
Chun-Yi Wu
Chih-Hsien Huang
Wen-Huei Chang
Yin-Tao Chang
Becky Chu
Kuo-Hwa Hu
Diana Lin
Pei-Chen Chiu
Jenny Chen
Yu-Chi Lin
Hong-Zhe Chen

Editorial Board

Yung-Sen Cheng (NSRRC)
Pei-Chen Chiu (NSRRC)
Lu Li (IMP/CAS)
Christine Petit-Jean-Genaz (CERN)
Volker RW Schaa (GSI)

Contents

Preface i
Foreword iii
Committees iv
Contents v

Papers 1
WEC1 – The Do's and Don'ts in Process Controls - Lessons Learned Over 35 Years 1
WEC2 – Status of the TPS Control System 6
WEC3 – Overview and Status of the SHINE Control System 11
WEC4 – HEPS Controls Status Update 14
WEC5 – !CHAOS General Status Report 17
WEP01 – EtherCAT Driver and Tools for EPICS and Linux at PSI 22
WEP02 – BLISS - Experiments Control for ESRF Beamline 26
WEP03 – A General Solution for Complex Vacuum System Controls 30
WEP04 – Developing and Validating OPC-UA Based Industrial Controls for Power Supplies at CERN 35
WEP05 – Wireless Internet of Thing Application in the TLS 38
WEP06 – Data Archiving and Visualization of IRFEL 41
WEP07 – Innovative Graphical User Interfaces Development: Give the Power Back to Users 44
WEP08 – A Universal System Based on WebSocket and JSON for the Employment of LabVIEW External Drivers 47
WEP09 – SwissFEL Electron Beam Diagnostics Tools and their Control System Components 51
WEP10 – Update on the Status of the FLUTE Control System 54
WEP11 – EtherCAT Solution for the TPS Control System 57
WEP13 – Control System Upgrade for the FFAG Accelerator Complex at KURNS 60
WEP14 – Vacuum Control System for the Taiwan Photon Source 63
WEP15 – Recent Development of the RIKEN RI Beam Factory Control System 66
WEP16 – CMS ECAL Detector Control System Upgrade Plan for the CERN Large Hadron Collider Long Shutdown II 69
WEP17 – Extending the Remote Control Capabilities in the CMS Detector Control System with Remote Procedure Call Services 73
WEP19 – TINE Release 5.0 : A First Look 77
WEP21 – Injection Control of the TPS 80
WEP22 – The Lens Effect in the Secondary Emission Based Systems of Joint Searching in EBW 83
WEP23 – Control System Using EPICS Tools at TARLA LINAC 85
WEP24 – The State Machine Based Automatic Conditioning Application for PITZ 88
WEP25 – Evolution and Convergence of Alarm Systems in Tango 92
WEP26 – The Development of Vacuum Gauge Monitoring and Control System using Web2CToolkit 96
WEP27 – Development of the Malfunctions Detection System at Vepp-2000 Collider 99
WEP28 – Development of Software for Accessing the Vepp-2000 Collider Facility Archiving System 102
WEP30 – Design of Reliable Control with Star Topology Fieldbus Communications for an Electron Cyclotron Resonance Ion Source at RIBF 105
WEP31 – Design of PLC Temperature Flow Acquisition System Based on EPICS 109
WEP32 – The Design and Development of an Auto-conditioning SRF Cavities Software 111
WEP33 – Introduction of Ciads Control System 113
WEP34 – The Control System based on EPICS for the Experimental Target Prototype 116
THCA1 – Quest for the New Standard PSI IOC Platform 119
THCA2 – Development of MicroTCA-based Low-level Radio Frequency Control Systems for cERL and STF 124
THCA3 – A Feedback/Feedforward System at the TPS and its Component Performance 127
THCA4 – Development of a Network-based Timing and Tag Information Distribution System for Synchrotron Radiation Experiments at SPring-8 131
THCA5 – Rethinking PLCs: Industrial Ethernet for Large-Scale Real-Time Distributed Control Applications 136
THCB1 – ACOP. NET : Not Just Another GUI Builder 139
THCB2 – Development of ACOP .NET STARS Transport Layer 144
THCB3 – Improving Web2cHMI Gesture Recognition Using Machine Learning 148
THCB4 – Leveraging Internet of Things Developments for Rapid Prototyping of Synoptic Displays 151

THCB5 – Intelligent Controls and Modeling for Particle Accelerators and Other Research and Industrial Infrastructures . . . 156
THP02 – Current Status of the RAON Machine Protection System Development . . . 160
THP03 – Marvin Update ' the Robotic Sample Mounting System at the Embl-Hamburg . . . 163
THP04 – Real-time and Detailed Provision of J-PARC Accelerator Operation Information from the Accelerator Control LAN to the Office LAN . . . 167
THP05 – Development and Current Status of KURAMA-II . . . 170
THP06 – Reliability Improvement for the Insertion Device Control in the TPS . . . 173
THP07 – Long-term Stability Observed by Electron BPM and Photon BPM for Taiwan Photon Source . . . 176
THP08 – Design and Implementation of Stepper Motor Control of the LINAC High Power RF System Based on FPGA . . . 179
THP10 – Collimator Motion Control System Upgrade for Medical Linear Accelerator Project at SLRI . . . 183
THP12 – Upgrading the Synchronisation and Trigger Systems on the Vulcan High-Power Nd:glass Laser . . . 187
THP13 – The Development of a FPGA Based Front End Safety Interlock System . . . 189
THP14 – Development of the New Spill Control Device for J-PARC MR . . . 192
THP15 – Design and Implementation of FPGA Based Protection System for Beam Acceleration in Linear IFMIF Prototype Accelerator . . . 195
THP16 – Remote Waveform Access Supports with EPICS for TPS and TLS Control Systems . . . 198
THP17 – Injection and Extraction Timing Controls at SuperKEKB Damping Ring . . . 201
THP18 – Operational Experience of the Digital LLRF Control System at the Booster Ring of Taiwan Photon Source 204
THP19 – Timing System for Multiple Accelerator Rings at KEK e^+/e^- Injector LINAC . . . 207
THP20 – Study of Energy Saving Operation for the TLS Booster Power Supply . . . 210
THP21 – Development of Triggered Scaler to Detect Miss-Trigger . . . 213
THP22 – Guaranteeing the Measurement Accuracy in Em#* . . . 216
THP23 – Further Improvements in Power Supply Controller Transient Recorders for Post-Mortem Analysis of BPM Orbit Dumps at PETRA-III . . . 220
FRCA1 – New Collaborative Approach to Scientific Data Management with NOVA . . . 224
FRCA2 – Development of a Task-Oriented Chatbot Application for Monitoring Taiwan Photon Source Front-End System . . . 228
FRCB1 – Ultra Fast Data Acquisition in ELI Beamlines . . . 230
FRCB2 – Design and Construction of the Data Warehouse Based on Hadoop Ecosystem at HLS-II . . . 233
FRCB4 – The Application for Fault Diagnosis and Prediction of Power Supply Control Device on BEPCII . . . 236
FRCC1 – FPGA-based Image Processing System for Electron Beam Welding Facility . . . 239
FRCC2 – Continuous Beam Scanning Intensity Control of a Medical Proton Accelerator Using a Simulink Generated FPGA Gain Scheduled Controller . . . 242
FRCC3 – CERN Supervision, Control and Data Acquisition System for Radiation and Environmental Protection 248
FRCC4 – Maintenance and Optimization of Insertion Devices at NSLS-II Using Motion Controls . . . 253

Appendices **257**
List of Authors . . . 257
Institutes List . . . 262
Participants List . . . 268

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC1

THE DO's AND DON'Ts IN PROCESS CONTROLS LESSONS LEARNED OVER 35 YEARS

M. Clausen, T. Boeckmann, J. Hatje, O. Korth, M. Moeller, J. Penning, H. Rickens,
B. Schoeneburg, Deutsches Elektronen Synchrotron DESY, 21220 Hamburg, Germany

Abstract

Designing, implementing and maintaining process control systems for cryogenic plants requires different viewpoints compared with those in machine controls. 24/7 operations for more than a year is a basic requirement. Hardware and software must be designed to fulfil this requirement. Many projects are carried out with industrial partners. Companies specify the process control logic which gets implemented by the local DESY team. Responsibilities, time tables and milestones must be clearly defined in such a case. Several cryogenic installations have been equipped with state of the art process control systems for cryogenic controls. Where the last one being the European XFEL. In the course of time commercial and open source systems were implemented and maintained. Control loops were basically always implemented in front end controllers running the real-time operating system VxWorks and EPICS as the control system toolkit. The approach to use PLCs will be discussed as an alternative approach. Large installations like the European XFEL require good project planning. Our success story will finalize our look back and initiate our look forward.

PROCESS CONTROLS AT DESY

In 1982 process controls for cryogenic systems was implemented in hardware PID controllers. Only a few engineers had the knowledge how to operate such a system. A failure over night was a night mare because no diagnostics were installed.

Over the years archive systems were installed. Even alarm systems found their way into cryogenic control systems because 24/7 operations required immediate action if some conditions were suspicious.

Over the years all of the cryogenic processes (cryogenic plants and cryogenic distribution systems) are controlled by process controllers. Some of them went through two basic refurbishments. In the end all cryogenic process controls at DESY and the XFEL are implemented in EPICS front end controllers – so called Input Output Controller (IOC).

This paper will describe a subset of the experiences we gained of the years. Namely: PLC integration; testing of new equipment and project management issues.

TO USE OR NOT TO USE PLCS IN PROCESS CONTROLS

PLCs can be really useful – there is no doubt. The question is: 'Where should PLCs be used?'

PLCs for Machine Interlocks

A very prominent usage for PLCs is the area of hardware interlocks. In former times this kind of hardware protection was implemented in hard wired logic. This logic slowly moved into intelligent controllers and finally into PLCs. These interlocks are well defined. They are thoroughly tested and should not be altered. Implementing this logic in a PLC is a no-brainer and used basically for every system in place.

PLCs as Data Concentrator

The next occasion where PLCs find their way into control systems is the usage as a data concentrator. There are a lot of different I/O signal types which can be connected via specific signal-conditioning modules to PLC type of communication controllers. Depending on the vendor theses are PLCs with I/O modules or intelligent communication controllers with I/O modules which can be programmed like a PLC.

In both cases the controller or the PLC will function as a data concentrator and will not be used for control functions in the field. Typically these controllers will be connected to a field bus like Profibus or CAN or (real-time) Ethernet.

Communication with the Process Controller

The communication between the data concentrator and the process controller will run through one of the field busses mentioned above. Therefore the process controller has to implement the necessary driver for that type of connection.

The communication must be configured for each individual I/O signal as if the signal would be connected to the process controller directly. This kind of configuration will be typically limited to one channel for each I/O signal. Basically this is a one to one representation.

Reliability

Reliability is a strong argument in favor of PLCs. For sure PLCs are known for their reliable runtime behavior. They run in thousands of instances and the code is (should be) thoroughly tested. All of the hardware components are nearly mill proof and reliable as well. Crashes of PLCs are not known to the author. If the same kind of application is run on an EPICS IOC one would have to put the same constraints on software and hardware on the EPICS implementation. A Windows or Linux operating system will hardly reach the level of reliability which a PLC based OS will provide. A 'real' real-time operating system like VxWorks is tuned for reliability and will work differently from OS's which require for instance hard

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC1

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

drives. The structural blocks of a PLC controls logic are not custom made. These are designed for reliability. The same implementation on an IOC would run as reliable as long as basic control blocks are used. Custom made EPICS 'records' or subroutines can easily cause distortions in the runtime behavior. Last not least the hardware requirements on an IOC must be comparable to those of a PLC. PLCs do not use the most powerful CPUs which need active cooling. They also come without hard drive. 'No moving parts on a reliable IOC'. This is a string requirement on reliable IOC.

Taking the above requirements into account there's basically no good reason why an EPICS based IOC should not run as reliable as a standard PLC. Examples show EPICS IOCs which have run for several years without interruption.

Redundancy

If the requirements on reliability go even further it is possible to run PLCs as well as IOC in a redundant setup. Both implementations are available.

Flexibility

Before discussing flexibility one has to agree on the setup being used. In this discussion the setup will consist of a PLC, an EPICS IOC and a console application on a PC. The next condition is the location of the process controls software. Where does the logic reside? On the PLC side or the IOC side?

Control Software on the PLC

A very obvious location could be the PLC. The logic will be 'close' to the I/O and take advantage of the reliable implementation of the PLC. The implementation of the controls logic will be straight forward because many companies specify their requirements in the controls logic in 'sort of' PLC language. Thus the implementation in the PLC is nearly 1:1. Once the logic is implemented it is a strong requirement that _all_ of the parameters in the controls logic shall be made available up to the operator console. This implies that the data exchange between PLC and IOC must be configured in a way to make that happen. Even basic control block in a PLC can be pretty complex. Between 30 and 60 different properties are possible. In addition precautions must be taken to allow not only for read but also for write operations. This way one control block on the PLC side may result in about one hundred records on the IOC side.

Besides the obvious overhead in the sheer number of records to be defined the configuration- and change management adds another level of complexity.

Last not least runtime diagnostic and runtime access to the controls implementation must be taken into account when control software shall run on the PLC. A simple example might explain the difference: If the controlled value of a control loop shall be changed from one sensor to another it would be a no-brainer on an EPICS IOC and is possible from the console level. Even if the sensor resides on another IOC it would still be possible to change the record name online. There's no way to do that on the fly on a PLC based implementation.

Control Software on the IOC

The arguments on the previous topic already describe the disadvantages of PLC based implementation in which we ran at DESY. The flexibility of running process control software on IOC or IOC type of controllers is the basis for running cryogenic controls at DESY and now for the European XFEL for the last 25 years. 24/7 operations require the maximum flexibility wih process controllers permanently running. Loading new software during operation is not an option. The existing implementation must provide all the flexibility the operators need to survive unforeseen situation. Any process value with all it's properties must be available on the console level without prior configuration. Operator or at least process engineers must be able to manually change (nearly) any property in the control system to continue operation by any means.

Besides this flexibility requirement one has to discuss the effort to configure both systems. In the latter case the I/O must be configured in the data concentrator or communication processor or PLC. This is the same in both approaches. The difference is the communication between communication processor and the IOC. In this case only the basic data including status and error conditions must be exchanged between both sides. This can be implemented in a well-structured way. The implementation of the controls logic would use record structures which are similar to the PLC block structures. Any property of these so called EPICS records will be available by default. No configuration will be necessary. In addition it is possible to add so called sequence- or state notation programs in a high level language. Implementing process control logic on the IOC side has clearly many advantages.

Process Engineer and Controls Engineer

Having mentioned the advantage of process control software on the IOC side it is still not always obvious that this is the only 'correct' approach. In the end it comes to people: Their experience and practical and theoretical background. A controls engineer with a strong PLC background who gets involved with process engineering will have the tendency to implement the controls software on the PLC side. A process engineer with no experience on the implementation of controls software might have the tendency to choose the more flexible approach.

The DESY/ XFEL Approach

DESY has a long history in implementing process control software on IOC type front end controllers. Initially a commercial control system was in place. Later that was replaced by EPICS IOCs. Reliable IOCs are running on Compact PCI CPUs which are powered by redundant power supplies and diskless without active cooling. Many of the IOCs are running redundantly. On the other hand we run so called 'soft IOCs' on virtual Linux machines implemented in a Hyper-V cluster on a set of 4 powerful computers. – The best of both world one may say. Of

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC1

course DESY also runs PLCs. But mostly for hardware interlocks. Some PLCs though run 'black box' software from the vendor of the controlled hardware. This is not always the best approach.

TESTING CONTROLS APPLICATIONS

A thorough test of controls applications requires a very elaborate implementation of test code. This can easily reach a level of complexity which is higher than the software which shall be tested. Especially testing the dynamic behavior of the controlled process can be very time consuming and will consume resources which are typically not available in this phase of a project.

Before thinking of the best of all worlds – namely dynamic testing – the test environment should cover the very basic requirements.

Test Environment: Simulation of Input Channels

A very basic requirement is to be able to simulate the input channels of the system. This can be implemented by setting the input channels into 'simulation mode'. Another approach is to let the input channels read from dedicated simulation channels instead of the connection to the 'real' I/O hardware (which will not be available for software commissioning). In both cases two tools should be configured:

Save/ restore for the Simulation Channels

Since we are living in a real world changes to he software will be necessary. These changes will probably require restarting the software. Procedures have to run through again (and again). To ease this process it will be useful to store the intermediate values in a save/ restore tool. This way an intermediate state can be recovered easily and the throughput of the test procedure will be more productive.

Graphical User Interface for all Channels to be Tested

Synoptic displays should be available for each channel from the very beginning. This is important to display not only the value but to get access also to all of the other properties of the channel. The preparation of groups of 16 channels using so called 'faceplates' is a well-established procedure at DESY. Faceplates for each type of I/O signal and different types of control loops have been created. They are configured by scripts in groups of 16.

Full Integration Test End to End

A final test from the sensor to the operator panel – or synoptic display should always be carried out. This final step will ensure that the data path from the sensor to (PLC to) IOC with all the necessary configurations the control system channel names and the graphical representation of the values – like STRING or DOUBLE values – will be tested and verified.

Archiving for Later Evaluation of the Test Results

Setting up the archive system from the very beginning is a goody but not absolutely necessary. Is is useful for bookkeeping and creating archive plots for the final acceptance protocols.

Setting up Alarms

Some signals will generate alarms which shall be used for soft-interlocks. It is useful to have the alarms configured upfront for checking the results in the alarm retrieval tools.

Logging in State Notation Programs

State notation – or sequencing programs do not per se offer logging of the status and states. It should be implemented from the very beginning to get messages especially on state changes. These should be grouped into messages just for commissioning and other messages which will be used for the operators during normal operations. Commissioning messages should be disabled or deleted after commissioning to avoid message floods.

PROJECT ORGANIZATION

Working in a scientific organization implies that the final design of equipment is shifted to the last minute. This opens the chance to provide the best technical design to the users. Besides the positive aspect of providing the latest/ greatest technical design it surely also has its drawbacks.

What kind of lessons could be learned from such an approach? The experience shows that working on the final design of the control system – and finishing it - before the final design of the controlled equipment is defined will cause a lot of frustration. Changes are necessary in the control system on the hardware side as well as the configuration and even the synoptic displays. Two approaches are possible to overcome frustrating experiences.

Just In Time Development

It is possible to wait with the final design of the control system until the final design of the controllable components has been approved. This will make sure that you implement what is really necessary. All the necessary control channels will be available but just in time does not necessarily mean – just in time –ready-! Waiting until the last minute could cause problems implementing the control logic and synoptic displays in time. You'll run into a phase of the project where everybody is extremely busy and necessary experts to specify the software might not be available.

You might end up with a control system which is barely working but does not perform the way expected. You might think you optimized the workload because no work had to be done twice but working in the last minute causes a lot of overhead in terms of personal stress and coordination effort between all parties involved.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC1

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

In the end a system will be in operation which performs 'just OK' but not good. It will cause a lot of frustration and might even cause problems during commissioning and the first phase of operating the new equipment. There is an alternative approach:

One Step Ahead Development

For the last big project – the European XFEL - we have chosen a different approach. We tried to calculate our workload precise enough from the very beginning. This implies that the requirements and specifications especially on the hardware-side must be 'frozen' way before controls equipment gets installed in racks. This even differs more when the installation takes place by external companies. A well-defined milestone for the final electrical layout of the control system must be specified. This is the milestone called: 'If you meet that milestone we'll meet the commissioning milestone'.

Commissioning Controls Hardware

Regardless where the controls racks are built they will need up to three checks:

1. A factory acceptance test will make sure that the completion of the controls hardware will meet about 95% correctness in terms of mechanical and electrical installation.
2. A pre-installation check. This will be a full integration check with a fully configured control system. The controls logic will be in place and the so called database (in EPICS speak) will be configured. So called Faceplates will be available for each I/O channel. Basic graphic panels will allow checking the full functionality of each channel. Ideally also the synoptic display will be in place in order to check the signals from the hardware signal up to the operator console.
   At this level also the archive system should be configured to provide the tools for cross checking the test results.
3. Last not least all channels will be tested when the controls racks are installed in the final destination. Once this test is passed the controls are ready for machine operations.
   For a cryogenic system it means that the cryogenic operations can be started. Valves can be operated and all readout signals are available.
4. Special cases in cryogenic systems are level measurements and heater controls. The final test of these systems can be performed as soon as the first liquid is 'dropping into' the cryogenic system. Special precautions should be taken to start reading level signals and running heaters not before liquid temperatures have been reached.

The biggest advantage of this procedure is that step one and two can be carried out way before the real installation in the field takes place. It is essential to know that the controls equipment has run through a 100% signal test. As soon as the 'OK' for installation is available you know that the expected error rate on the I/O system will be less than one per mill.

Quality of Cable Works

To reach the one per mill error rate from the sensor to the operator panel you'll have to make sure that the cabling will be as reliable as the electronic racks themselves. In our case we were lucky that just one team worked on the cabling and carried out this task throughout the whole installation procedure. It should not be under estimated that getting one or more teams familiar with the cable plans for your specific layout might require working time from the controls team and might be error prone if teams are changing regularly. The one per mill error rate can only be achieved if the cabling team does _not_ change on a regular basis. Otherwise higher error rates and time consuming corrections will be necessary.

Sliding Project Plans

No project ever will stick to the initial time table and project plan. Of course the project management will always argue that milestone are carved into stone and might not slide but we are living in a real world…

How can milestones and sliding project plans be implemented side by side? This is much easier than it might look like. Implement your project plan like an 'If – Then – Else' statement in any programming language. It could look like this: If the milestone 'Installation of control racks' is kept we will need x-weeks until we are ready for machine commissioning – else – our project plan will 'slide' by (the same number of) x weeks.

This way you always keep the necessary time for the controls commissioning independent of the starting date. You might have to add constraints like 'during summer time from A to B it will take C-weeks longer due to vacation of part of the team etc.

As a result one should avoid defining the end of controls commissioning as a milestone. Since controls are always the last ones to finish – by definition – the time for preparation will get squeezed into the remaining time available and – in the end - it will not be enough to finish the task. Project management might ask why controls are late and all those things which should be avoided.

If there's a message from the experience of getting several projects implemented it is this one:
Implement a project with a minimum of hard coded mile stones and get especially controls' project plan implemented as a sliding project plan – based on previous mile stones.

Working with Industry

Working with industries can be fun – if well-defined rules have been established. It is not a secret that many companies (hopefully not in general) are happy about any change in design and delay caused by the customer especially if the customer has to provide deliverables like controls software. To avoid unnecessary discussions it is advisable that project plans are clearly defined and do not open points for interpretation or discussion.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC1

It could easily happen that deliverables by the customer are delayed because the boundary conditions and specifications are missing. Such a conflict can be avoided if the conditions are clearly described. Like: The controls software will be ready for commissioning x-weeks after the requirements and specifications have been delivered to the customer. After commissioning y-weeks will be necessary to incorporate the changes into the software. After this the controls software will be ready to commission the equipment and to run later acceptance tests.

This is a variant of the sliding project plan. The controls software will not be ready at a certain date (milestone) but after well specified conditions have been reached. In many cases these conditions are under responsibility of the industrial partner. As soon as this is understood you will reach a well-defined partnership.

CONCLUSION

Working for 36 years at DESY was interesting from the first day and except a few occasions I enjoyed working here every day. Being part of a team which built HERA in the 80^s and later the XFEL in the 2010s was a great experience which I never want to miss.

Results count: The quality of the XFEL cryogenic controls implementation is speaking for its own.

ACKNOWLEDGEMENTS

I want to thank the colleagues in the cryogenic controls group for the fruitful collaboration and joint work over 33 years at DESY. My thanks also go to the engineers and operators of the cryogenic machine group at DESY. Also the good relationship to the DESY machine controls group should be mentioned.

I want to thank everybody in the EPICS community for their excellent developments. Without the highly reliable software from the EPICS community it would not have been possible to finish the tasks at DESY and the XFEL.

Last not least I want to thank my wife Marlies for her permanent support.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC2

STATUS OF THE TPS CONTROL SYSTEM

Y.S. Cheng, J. Chen, P.C. Chiu, C.Y. Wu, C.Y. Liao, C.H. Huang, K.H. Hu, Y.T. Chang, Demi Lee, S.Y. Hsu, C.H. Kuo, C.J. Wang, K.T. Hsu
National Synchrotron Radiation Research Center, Hsinchu 30076, Taiwan

Abstract

Control system for the Taiwan Photon Source (TPS) has been delivered in mid-2014 to support commissioning and routine operation of the accelerator system. The TPS control system adopts EPICS toolkits as its frameworks. Various subsystems interface to the control system according its specific requirements. Operation experiences accumulated during last four years confirmed the system working well. Minor revisions were made to improve the system performance. Current status of the control system and ongoing developments will be summarized in the report.

INTRODUCTION

The TPS [1] is a latest generation of high brightness synchrotron light source which is located at the National Synchrotron Radiation Research Center (NSRRC) in Taiwan. TPS consists of a 150 MeV electron linac, a booster synchrotron, a 3 GeV storage ring, and experimental beam lines. The control system environment was ready in mid-2014 to support subsystem integration test and accelerator commissioning [2]. User service of the TPS was started since 2016.

Adequate and reliable functionality of control system is one of the key to the success of TPS commissioning. Control system for the TPS is based on the EPICS framework [3]. The EPICS is based on the definition of a standard IOC structure with an extensive library of driver and support a wide variety of I/O cards. The EPICS toolkits have various functionalities which are employed to monitor and to control accelerator system such as control page creation, data archiving, alarm handling and etc.

The TPS control system consists about three hundred of EPICS IOCs. The CompactPCI (cPCI) is equipped with input/output modules to control subsystems as standard IOC. The other kinds of IOCs are also supported by the TPS control system, such as BPM IOCs, PLC IOCs, various soft-IOC and etc.

To achieve high availability of the control system, emphasis has been put on software engineering and relational database for system configurations. Data channels in the order of 10^5 will be serviced by the control system. Accessibility of all machine parameters through control system in a consistent and easy manner will contribute to the fast and successful commissioning of the machine. High reliability and availability of TPS control system with reasonable cost and performance are expected.

SYSTEM COMPONENTS

The TPS control system provide an environment for operation of the accelerator system. It integrated various subsystems together [4]. Details of some parts of control system are summarized in the following paragraph.

Control Server Environment

Various servers include file server, archiver server, database server, alarm server, boot servers, ...etc. are supported by the control system. There server support operation of the whole system.

General EIPCS IOC Interface

There are many different kinds of IOCs at equipment layer to interface various subsystems and instruments. They need meet various requirements of functionality requirements, convenience and cost consideration. Type of IOCs are summary in Table 1. Most of the devices and equipments are directly connected to cPCI IOCs with EPICS. The cPCI EPICS IOC is equipped with the cPCI-6510 CPU board. The cPCI-7452 128 bits DI/DO module is used for BI, BO solution. ADC and DAC modules in IP (Industry pack) module form-factor are used for smaller channel count application, such as insertion devices control. Event system modules are in 6U cPCI form-factor. Private Ethernet will be heavily used as field-bus to connect many devices. Power supplies of all magnets except for correctors are equipped with Ethernet to the EPICS IOC. Multi-axis motion controller with Ethernet interface is the standard for the control system.

Ethernet attached devices are connected to the EPICS IOC via private Ethernet. Devices support VXI-11, LXI, Raw ASCII and Modbus/TCP protocol are connected to EPICS IOC directly by TCP/IP interface. Devices of this category include intermediate power supply, temperature acquisition (RTD or thermocouple), digital multi-meters, oscilloscopes, signal generator, and other instruments.

All corrector power supplies are controlled by the corrector power supply controller (CPSC) module [5]. The CPSC embedded EPICS IOC, it can access via channel access protocol. The CPSC equips with 20 bits DAC and 24 bits ADC. Two SFP ports supported by the on board FPGA, these SFP ports are receive correction setting from fast orbit feedback FPGAs located at BPM platforms via AURORA protocol.

All BPM platform are embedded EPICS IOCs. Some PLC based embedded IOC were adopted for machine protection and interface of pulse power supply of pulse magnet.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC2

Table 1: Type of EPICS IOCs

Type	Quantity	Applications
6U CompactPCI	~ 40	CIA IOCs, Timing, RF, ID, etc.
PLC embedded EPICS IOC	13	Pulse Power Supply, MPS
Fanless platform with PCIe	~20	Beamline gateways and Timing
Embedded w POE	~5	Image IOCs
Embedded IOP	~122	Corrector PSs
COM Express Embedded (µTCA platform)	72	Storage ring and booster synchrotron BPMs
1U platform	13 (xScale)	Transfer line BPMs
Micro ATX	3	Bunch-by-Bunch Feedbacks
Miscellaneous	~20	Software IOCs, LabVIEW IOCs, etc.

Power Supply Control

TPS power supplies control interface is divided into three categories based upon products from three different vendors. The small power supply for corrector magnets and skew quadrupoles is an ±10 Amp class power supply. This category power supply will be in module form-factor. Each power supply sub-rack can accommodate up to 8 power supply modules. A custom designed CPSC module was installed at control slot of the sub-rack. The CPSC will be embedded with EPICS IOC and provide fast setting SFP ports to support orbit feedback functionality. Power supply modules installed at the same sub-rack will interface to this CPSC module.

The intermediate power supply with current rating 250 Amp is equipped with Ethernet interface. Power supplies are expected to have internal data buffer with post-mortem capability. There are two versions of power supply in this category, sextupole power supply with 16 bits precision and quadrupole power supply with 18 bits precision.

Storage ring dipole DC power supply and power supplies for the dipoles and quadrupoles of the booster synchrotron are contracted to Eaton. Each power supply equips with RS-485 serial interface. MOXA serial to Ethernet adapters enable directly interface with the EPICS IOCs. The storage ring dipole will be control via this link. Booster dipole and quadruple power supplies are interface by precision analogue interface. The DACs and ADCs operated synchronize by the same clock and trigger to achieve better reproducibility. Waveform generate form the DAC on IOC will drive these booster power supplies. This functionality is essential for energy ramping of the booster synchrotron [6]. Control resolution of these power supplies has 18 effective bits.

Networking

Mixed of 1 and 10 Gbits/s switched Ethernet are deployed for the TPS control system [7]. The Gigabit Ethernet connection was delivered at edge switches installed at control and instruments area (CIA). The control network backbone is a 10 Gigabit link to the control system computer room. Private Ethernet is used for Ethernet based devices access which support fast Ethernet and GbE. Adequate isolation and routing topology will balance between network security and needed flexibility. The file and database servers are connected to the control and intranet network, allowing the exchange of data among them. Availability, reliability, cyber security, and network management are all considered.

Timing System

The event system consists of event generator (EVG), event receivers (EVRs) and a timing distribution fiber network [8, 9]. The EVG and EVRs can be installed with various universal I/O mezzanine modules to meet different input/output requirements. The mechanical form-factor of EVG and EVRs is in 6U cPCI module. The 125 MHz event rate will deliver 8 nsec coarse timing resolution. Fine delay is supported by the EVRTG which generates gun trigger signal. Its high resolution and low timing jitter provide accurate synchronization of hardware and software across the TPS control system.

Insertion Devices and Front-end

TPS will accommodate up to 21 sets of insertion devices (ID) in 2023 (Fig. 1). There are 10 sets in phase I (~ 2016), 7 sets in phase II (2017~2020) and 4 sets in phase III (2020~2023) for insertion devices projects. Phase I IDs include one set of EPU46, two sets of EPU48 [10] and seven sets of in-vacuum insertion devices (two sets of 2 meters long IU22-2m, three sets of 3 meters long IU22-3m, and one set of 3 meters long IUT22-3m with taper functionality). Construction for Phase II IDs are on-going consists of seven sets of 2 sets of in-vacuum undulator, 2 sets of cryogenic insertion undulator, and one wiggler, two EPUs. Motion control performed by using Galil DMC-40x0 motion controller. In-house developed EPICS device support for this motion controller was developed to support up to 200 Hz access. A cPCI EPICS IOC equips with AI/AO/BI/BO I/O modules were used for the Phase I IDs. All parameters of motion controller will be created as EPICS PVs. Update rate can be up to 200 Hz. This would be useful for feed-forward ID field error compensation which operate at 100 Hz rate at this moment. Control environment for Phase I and II insertion devices control as shown in Fig. 2 and Fig. 3 respectively. The user interface of insertion device with front-end layout is shown in Fig. 4.

To improve functionality and openness of control environment for future development and accompany with another application for upgrade of existed control system (such as the control system of Taiwan Light Source). Start from Phase-II, cPCI crate will replaced by PC platform in various form factor (include fanless embedded PC), the cPCI digital and analog I/O modules will replaced by EtherCAT modules. EtherCAT can support 200 Hz rate access which is compatible with current system.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC2

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

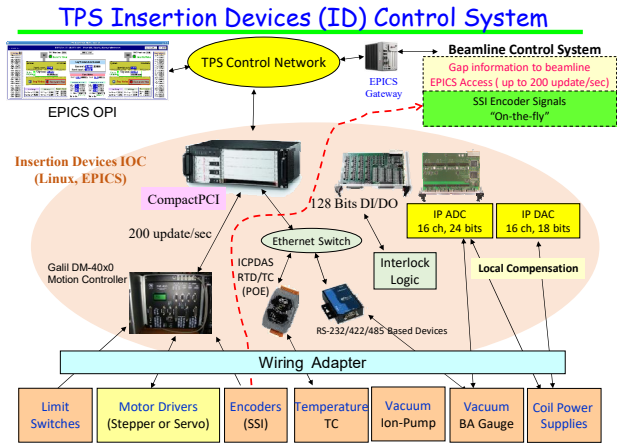


Figure 1: Standard hardware configuration for Phase I insertion devices control.

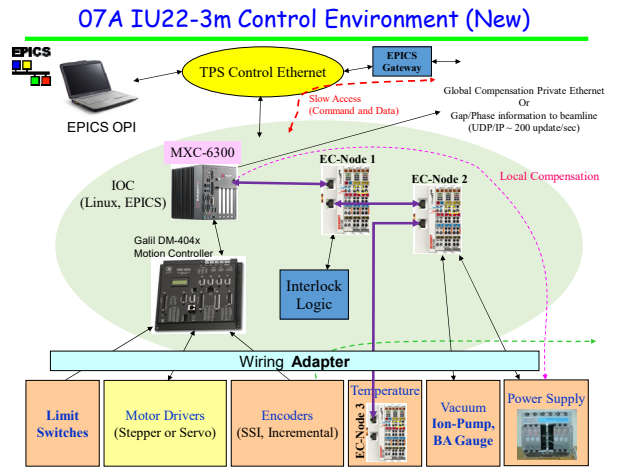


Figure 2: Standard hardware configuration for Phase II and future insertion devices control.

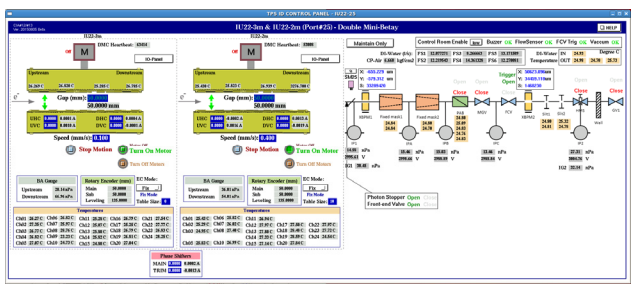


Figure 3: Graph user interface of two insertion devices (IU22-3m + IU22-2m) with front-end.

Diagnostic System

Digital beam position monitor (BPM) electronics is equipped with Ethernet interface for configuration and served as EPICS CA server with 10 Hz data rate. Another multi-gigabit interface delivery beam position for fast orbit feedback purpose at 10 kHz rate. Post-mortem buffer for orbit analysis during specific event happened like beam loss are also supported by the BPM platform. Many BPM related applications were developed to help various applications [11, 12].

Fast orbit feedbacks [13], bunch-by-bunch feedback supports, post-mortem data acquisition for other signals other than BPM and data analyzer [14], …etc. are useful for the TPS delivery stable beam.

High precision beam current reading and lifetime calculation was done at a dedicated IOC. This IOC will install EVR to received booster cycle timing signals and high resolution IP ADC modules to digitize the DCCT signal and perform beam lifetime calculation.

The GigE Vision digital cameras support for screen monitor [15], synchrotron radiation monitor, X-ray pinhole camera [16] and other applications. Counting type and integrating type beam loss monitors was connected to the control system by counter or ADC modules installed at IOCs.

PLC and Interlock

The PLC was used for most of control system related interlock system [17]. FM3R with embedded EPICS IOC is also used for some applications, such as pulse magnet power supply control and machine protection system (MPS). The MPS collects various interlock signals from local interlock subsystem of orbit, vacuums, front-ends, and etc. The beam disable commands to trip beam or inhibit injection can be distributed to the specific devices or subsystem by the global machine interlock system or uplink functionality of the event system. The summarize control page is shown in Fig. 4.

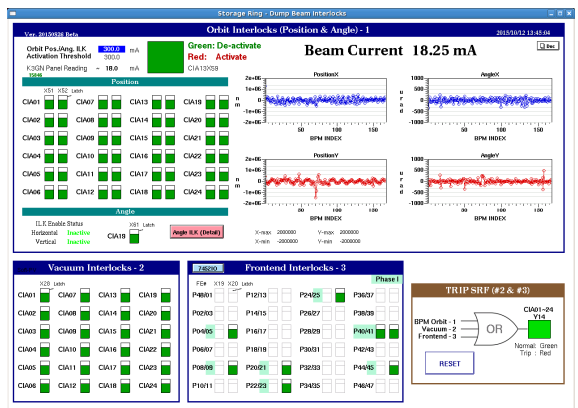


Figure 4: The machine protection system control page.

Beamline and Experimental Station Interface

To allow beamline and end-station users access to the accelerator control and timing systems, an EPICS gateway/EPICS IOC was set up. The gateway provides an interface to share data between accelerator and beamlines and to allow end-station users to control insertion devices. The EPICS IOC support provides beamlines with a flexible and configurable timing in support of experiments including injection gating, revolution clock, clock with specific frequency for streak camera triggers, beam current broadcasting via an event based timing system with a frequency signal which is proportional to the beam current, a trigger signal for data acquisition, etc.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC2

A low jitter RF clock (< 1 psec) is also distributed to the beamlines which need a high quality RF clock for laser system synchronization and a clock for streak camera synchro-scan operations via phase stabilized optical fibers (PSOF). This POSF fiber distribution is insensitive to ambient temperature variations which could otherwise cause timing drifts of tens of picoseconds with only a few degrees of temperature change. A frequency divider and phase locked loop (PLL) also support beamlines to generate a high quality clock at specific frequencies to support diverse applications. Figure 5 shows the configuration of this accelerator to beamline system interface. The fiber links include computer network for beamline and end-station, computer network for accelerator control systems, event timing system networks and a POSF fiber link.

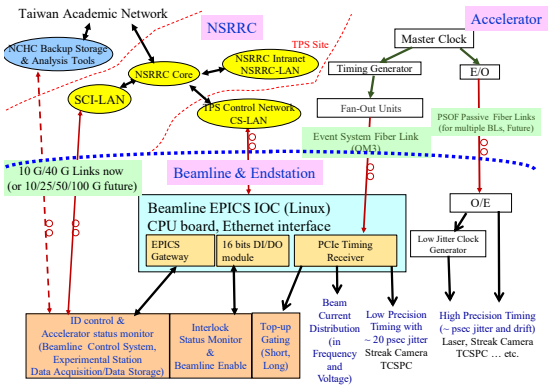


Figure 5: Accelerator control system interface with beamlines.

CONTROL APPLICATIONS

Generic applications provided by the EPICS toolkit will be used for all kinds of applications [18]. Standard tools such as the archive system, alarm handler and save/restore tools are supported. Channel Access (CA) is used as an interface for process variables (PVs) access. Simple tasks such as monitoring, alarm handling, display and setting of PVs are performed using EDM panels and strip tools. Cold start, warm up and shutdown process are done by MATLAB scripts.

Operator Interface

The operator interface level consists of Linux PCs for consoles and servers for various purposes. Operation user interface (OPI) is implemented by EDM, CSS (Control System Studio), MATLAB, and Python. The TPS launcher by the control page is built by the EDM toolkit shown as the Fig. 6. All control pages can be launched from this page. All control components are located at the foreground of the TPS accelerator illustration.

Save and Restore

To readily restore a set of the machine parameters for subsystems during operation as well as to optimize and record working points for different machine conditions, the mechanism of save and restore is developed. The save and restore function is established by using the MATLAB with the labCA. The various files of grouped PVs (Process Variables) list are created for saving the respective parameter values of each subsystem. The file with PVs and saved parameters is also selectable for resume the settings.

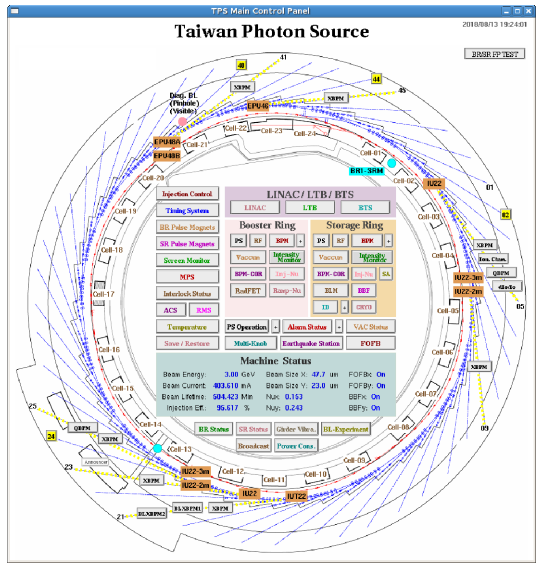


Figure 6: The TPS main control page made by EDM tool.

Injection Control

Injection control for TPS have been served since the machine commissioning [19]. Top-up injection functionality is available from day one of machine commissioning stage to assist vacuum conditioning. During last two years, several updates have been done to enhance flexibility for the injection control to help efficiency operate of the TPS. The injection control includes foreground and background processes to coordinate the operation of e-gun, linear accelerator, booster synchrotron, storage ring by the help of event based timing system. Lifetime calculation of the storage ring is also synchronized with the injection process.

Alarm Handler

The "BEAST" (Best Ever Alarm System Toolkit) of CS-Studio with the MySQL RDB is adopted as the alarm handler for the TPS, as shown in Fig. 7. A distributed alarm system monitors the alarms in a control system and helps operators to make right decisions and actions in the shortest time. In the CS-Studio alarm system, each alarm is supposed to be meaningful, requiring an operator action. An alarm is no status display that operators may ignore. Each alarm requires an operator to react because the control system cannot automatically resolve an issue.

Archive and Logbook

The archive system of CSS (Control System Studio) named BEAUTY (Best Ever Archive Toolset, yet) was built to be used as the TPS data archive system [20]. An archive engine takes PV data from EPICS IOCs via

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC2

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

channel access, and stores them in the data storage. The PostgreSQL RDB (Relational Database) was adopted as the data storage for the BEAUTY. Both the historic PVs data and the archive engine configuration are saved into the same RDB. The archived data can be retrieved in a form of graphical representation using the CSS-based data browser. Taking the performance and redundancy into considerations, the storage servers and RDB table structures are tuned relative.

The "Olog" [21] solution is selected for the TPS electronic logbook. The TPS logbook is recorded the progress about commissioning information by the commissioning team and operators, and supports print function to copy data into the logbook for logging.

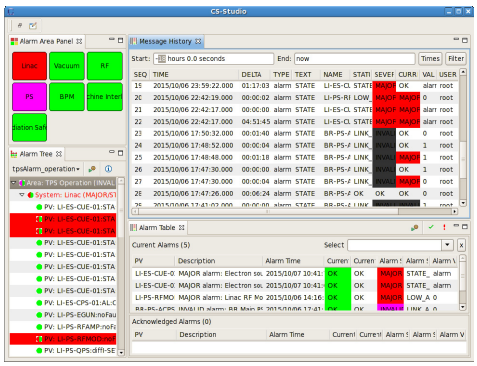


Figure 7: The CS-Studio alarm handler interface.

Miscellaneous

There are many special type applications supported by the control system such as earthquake data acquisition, Raspberry Pi/Banana Pi based EPICS IOC for special embedded applications. All of these application enrich the contents of the control system.

FUTURE PROSPECTIVES

To cope fast evolution of hardware and software technologies for future development of the TPS, small scale development with limited resources is under way. Theses development include MTCA.4 platforms for diagnostic and timing applications, aTCA platform for multiple EPICS gateway applications, open source event receiver for future beamline and end-station timing support, Python application development for next step application programs, etc.

SUMMARY

The TPS take advantages of the latest hardware and software developments to deliver high performance, rich functionality, and economically control system. TPS control system working well during last five years. During these period, several minor revised were performed to enrich of contents of the control system. Performance index includes scalability, reliability, and performance, maintainability, etc. were improved significantly since its deploy.

REFERENCES

[1] TPS Design Book, v16, September 30, 2009.
[2] C.C. Kuo *et al.*, "Commissioning of the Taiwan Photon Source", TUXC3, in *Proc. IPAC'15,* Richmond, VA, USA, May 2015, paper TUXC3.
[3] EPICS, `http://www.aps.anl.gov/epics/index.php`
[4] Y.S. Cheng *et al.,* "Status of Control System for the TPS Commissioning", in *Proc. PCaPAC'14,* Karlsruhe, Germany, Oct. 2014, paper WPO033.
[5] D-TACQ, `http://www.d-tacq.com`
[6] P.C. Chiu *et al.,* "Control Supports for TPS Booster Synchrotron", in *Proc. IPAC'14*, Dresden, Germany, Jun. 2014, paper THPRO121.
[7] C.S. Huang *et al.,* "Network Architecture at Taiwan Photon Source on NSRRC", in *Proc. PCaPAC'14*, Karlsruhe, Germany, Oct. 2014, paper WPO034.
[8] Micro Research Finland, `http://www.mrf.fi`
[9] C.Y. Wu *et al.,* "Integration of the Timing System for TPS", in *Proc. IPAC'14,* Dresden, Germany, Jun. 2014, paper THPRO121.
[10] C.Y. Wu *et al.,* "Control System of EPU48 in TPS", in *Proc. IPAC'14*, Dresden, Germany, Jun. 2014, paper THPRO123.
[11] P.C. Chiu *et al.,* "The Role of Beam Diagnostics in the Rapid Commissioning of the TPS Booster and Storage Ring", in *Proc. IBIC'15*, Melbourne, Australia, Sep. 2015, paper MOBLA01.
[12] Pei-Chen Chiu, *et al.,* "Commissioning of BPM System for the TPS Project", in *Proc. IBIC'15*, Melbourne, Australia, Sep. 2015 paper TUPB068.
[13] P.C. Chiu, *et al.,* "Preliminary Beam Test for TPS Fast Orbit Feedback System", in *Proc. IPAC'16*, Busan, Korea, May 2016, paper WEPOW040.
[14] C.Y. Liao, *et al.,* "Post-mortem system for the Taiwan Photon Source", in *Proc. IPAC'17*, Copenhagen, Denmark May 2017, paper MOPAB125.
[15] C.Y. Liao *et al.,* "TPS Screen Monitor User Control Interface", in *Proc. PCaPAC'14*, Karlsruhe, Germany, Oct. 2014, paper FPO032.
[16] C.Y. Liao *et al.,* "Synchrotron Radiation Measurement at Taiwan Photon Source", in *Proc. IBIC'15*, Melbourne, Australia, Sep. 2015, paper TUPB067.
[17] C.Y. Liao *et al.,* "Implementation of Machine Protection System for the Taiwan Photon Source", in *Proc. IPAC'14*, Dresden, Germany, Jun. 2014, paper THPRO126.
[18] Y.S. Cheng *et al.,* "Control System Software Environment and Integration for the TPS", in *Proc. PCaPAC'14*, Karlsruhe, Germany, Oct. 2014, paper FPO030.
[19] Jenny Chen *et al.*, "Injection Control of the TPS", presented at PCaPAC'18, Hsinchu, Taiwan, Oct. 2018, paper WEP21, this conference.
[20] Y.S. Cheng *et al.,* "Implementation of the EPICS Data Archive System for the TPS Project", in *Proc. IPAC'13*, Shanghai, China, May 2013, paper THPEA049.
[21] Olog, `http://olog.sourceforge.net/olog`

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC3

OVERVIEW AND STATUS OF THE SHINE CONTROL SYSTEM

Y. B. Yan†, J. G. Ding, J. F. Chen, G. H. Chen, Y. B. Leng
H. F. Miao, Y. J. Liu, Q. R. Mi, H. Zhao, C. L. Yu, H. H. Lv, H. Y. Wang, P. X.Yu
Shanghai Advanced Research Institute, Chinese Academy of Sciences
201204 Shanghai, P.R. China

Abstract

The high-gain free electron lasers have given scientists hopes for new scientific discoveries in many frontier research areas. The Shanghai HIgh repetition rate XFEL aNd Extreme light facility (SHINE) was proposed by the central government of P.R. China on April 2017, which is a quasi-continuous wave hard X-ray free electron laser facility. The control system is responsible for the facility-wide device control, data acquisition, machine protection, high level database or application, as well as network and computing platform. It will be mainly based on EPICS to reach the balance between the high performance and costs of maintenance. The latest technology will be adopted for the high repetition rate data acquisition and feedback system. The details of the control system design will be reported in this paper.

OVERVIEW

Owing to the wide range of applications of X-rays in the research fields of physics, chemistry and biology, facilities with the ability to generate X-rays were developed continuously in the last century. The free electron laser (FEL) is a novel light source, producing high-brightness X-ray pulses. To achieve high-intensity and ultra-fast short wavelength radiation, several X-ray FEL facilities have been completed or under construction around the world [1].

The first hard X-ray FEL light source in China, the so-called Shanghai HIgh repetition rate XFEL aNd Extreme light facility (SHINE), is under construction. It will utilize a photocathode electron gun combined with the superconducting Linac to produce 8 GeV FEL quality electron beams with 1 MHz repetition rate.

The control system of SHINE is responsible for the facility-wide device control, data acquisition, machine protection, high level database or application, as well as network and computing platform. It will provide operators, engineers and physicists with a comprehensive and easy-to-use tool to control the machine components to produce high quality electron beam and free electron laser.

According to the experience of SSRF and SXFEL, the control system of SHINE will be mainly based on EPICS (Experimental Physics and Industrial Control System) to reach the balance between the high performance and costs of maintenance. EPICS is a set of open source software tools, libraries and applications developed collaboratively and used worldwide to create distributed soft real-time control systems for scientific instruments such as particle accelerators, telescopes and other large-scale scientific experiments.

ARCHITECTURE

As shown in Fig. 1, the control system can be divided into four layers to ensure the performance and scalability, which are operator interface layer, middle layer, device control layer and data acquisition layer.

The operator interface layer offers graphical user interface (GUI), command line interface (CLI) and high-level application programming interface (API) to operators, engineers and physicists. It allows them to interact with the machine components.

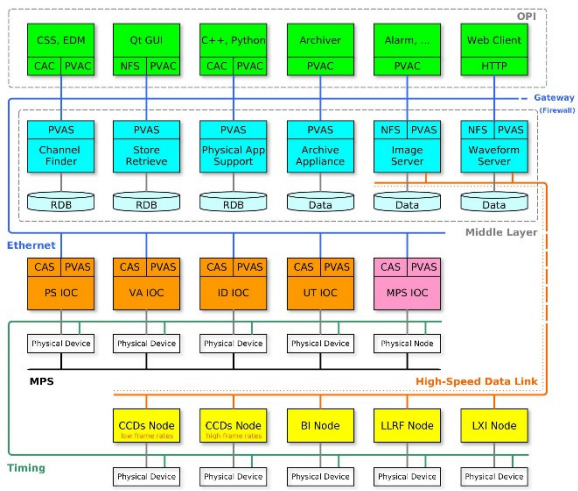


Figure 1: Architecture of the control system.

The middle layer consists of compute, storage and network devices. It provides the runtime environment for the whole control system. It also undertakes the centralized processing tasks for image and stream data acquisition system.

The device control layer is responsible for the facility-wide input and output device control, such as magnet power supply control, vacuum gauge control, stepper motor control and so on. The machine protection system will be also implemented in this layer. They are the basis components of the control system.

The data acquisition layer is designed for the high speed image and stream data acquisition, processing and storage. It involves the beam and laser diagnostics, microwave related system. Some custom software will be used at this layer.

† yanyingbing@sinap.ac.cn

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC3

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ENVIRONMENT

The central servers consist of several high performance computers, which provide the services as follows.

- EPICS Services: include the base, support modules, extensions, cross-compile support for different hardware platforms. All console computers register on the server and use the shared resources, such as display manager.
- Network File Service: provide file sharing solution for the control system. The authorized clients can view and optionally store and update files on the servers.
- Directory Services: used for the authorization of accessing various resources. The OpenLDAP will be adopted.
- Archive Services: store various runtime data for the parameters tracking and fault diagnosis. The scalar data will be achieved in channel archiver. The stream data will be achieved using dedicated system. The relational databases, non-relational databases or distributed file systems will be selected.
- Time Synchronization Service: use the network time protocol to synchronize the controllers, servers and console computers. It keep their internal clocks synchronized to a uniform reference time.

In addition to the above, other services are web service, backup service, software version control services, syslog service and so on.

The most services run on the virtual servers, which are highly scalable and available servers built on the cluster of physics servers. The architecture of the server cluster is fully transparent to the users. The virtual servers have been used for several years at SSRF, which improved the stability and reliability significantly.

NETWORK

The network is the important infrastructure for modern distributed control system. The performance and reliability is critical. In this project, the Ethernet will be used as the backbone of control network. Most of hardware devices connect to 10 GbE access layer switches. The access layer switches have two connections to separate 40/100 GbE core switches, which is designed to ensure the reliability. The VLAN (Virtual Local Area Network) will be adopt, which divides the network into multiple subnets.

Several dedicated network will be built for the device control and data acquisition sub-system. For some high-speed cameras, point-to-point optical fiber connections will be used.

DEVICE CONTROL

The device control is mainly responsible for the magnet power supply, vacuum gauge, ion pump, stepper motor and so on. The IOCs (Input / Output Controllers) are the key components of the control system. All the interfaces to hardware devices will be through the IOCs. The IOCs connect the local controllers via backplane bus, filed buses or Ethernet. By using the protocol converters (including software and hardware), the devices with field buses such as RS-232/RS-485 can be reached via Ethernet.

The IOCs hardware must be COTS (Commercial Off-The-Shelf) products with the features of high performance and cost effective. The fanless industrial embedded controllers will be the best option, such as DA-68x series from MOXA or ARK-3000 series from AdvanTech.

DATA ACQUISITION

The data acquisition is mainly responsible for the image acquisition for beam profile measurement and laser diagnostics system, waveform data acquisition for beam instrumentation system (position, charge, length and arrival-time measurement), microwave related system and general LXI devices (oscilloscope and spectrum analyzer). Due to high repetition rate, the large amount and high rate of data acquisition will be challenges.

The industrial cameras support multiple buses, such as FireWire, USB, GigE Vision, Camera Link, CoaXPress and so on. The GigE Vision is a globally accepted camera interface standard developed using the Gigabit Ethernet communication protocol. It offers the greatest technical flexibility in terms of bandwidth, cable length and multi-camera functionality. The Camera Link is a serial communication protocol standard designed for computer vision applications. It is used for the higher speed connectivity.

For the majority of low fps image acquisition, the GigE Vision cameras will be adopt. A dedicated network will be employed to transmit the images to the servers. The software will based on the areaDetector and aravisGigE modules. The image processing contains image rotations, flips, ROI (Region of Interest), profiles, etc.

For the minority of high fps image acquisition, the Camera Link cameras will be adopt. The Camera Link optical extender can be used to extend the data transmission distance over optical fiber links. The software mainly focus on the image acquisition and storage. The image processing and analysis will be completed offline.

For the LXI device, The LXI (LAN eXtensions for Instrumentation) consortium defines standard ways for Ethernet-based instruments to communicate, operate and function. The data is transmitted to the server through the dedicated network. The software will be developed via VISA (Virtual Instrument Software Architecture).

SOFTWARE

The whole control system will be based on open-source software. The CentOS (Community enterprise Operating System) will be selected, which is a free operating system distribution based upon the Linux kernel. The major software will adopt the latest stable release series of EPICS Version 3. The components of Version 7 will be evaluated and used partially.

The most IOCs will be built with the existing records and device support modules. The StreamDevice will be used for the devices with serial communication interfaces, including RS-232, RS-485, GPIB, TCP/IP, etc. The netDev

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC3

will be used for communicating with Omron and Yokogawa PLCs, s7nodave for Siemens S7 PLC. The custom IOCs will be based on asynDriver, which is a general purpose facility for interfacing device specific code to low level communication drivers.

The high level application plays an important role in the whole control system. It provides a platform to hold the information base, as well as a set of cooperating services for data access [2]. It is composed of three layers: database, service and application, as shown in Fig. 2.

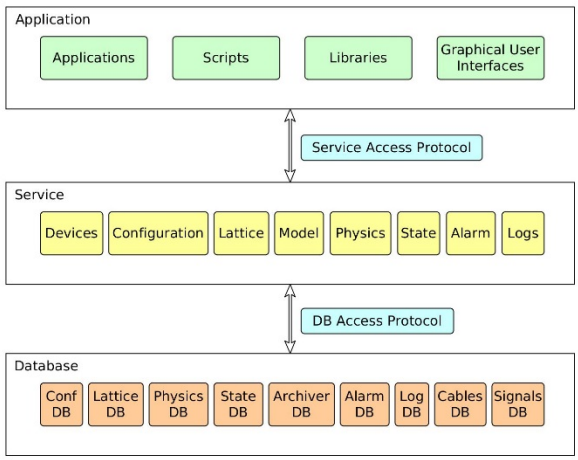


Figure 2: Architecture of the high level application.

The database is a general data storage container for the machine configuration, lattice, state, alarm and so on. The service implements a series of control and physical related logic operations, and provides standard interfaces for the application layer, including local interfaces and remote interfaces. The application layer consists of kinds of tools and components used to display the information to users. Both web-based and desktop-based interfaces will be implemented. The application layer accesses the database through the service layer.

The free electron laser facility is composed of many sub-systems. Combining data from multiple sources makes it possible to utilize data mining and analysis. The high level application provides operators, engineers and physicists easy-to-use programming interfaces to access the databases. It can simplify the application development, offload the heavy database query and shield from the database structure changes.

TIMING

The timing system provides trigger and timestamps signals to the accelerator equipment, including injector, laser, modulators, beam and laser diagnostic system. The White Rabbit (WR) technology will be evaluated.

The White Rabbit is a collaborative project including CERN, GSI, and other partners from universities and industry. It can provide sub-nanosecond accuracy and pico-seconds precision of synchronization for large distributed systems. It also allows for deterministic and reliable data delivery.

The timing system is composed of master, WR switch and node devices. The master receives reference signal from the synchronization system. The switches distribute the clock to all the nodes in the network using a hierarchical architecture. The node basic functionality comes in the form of an IP Core called WR PTP Core. They can be standalone trigger fanout modules or FMC boards, which can be embedded in the DBPM and LLRF processor.

MACHINE PROTECTION

The Machine Protection System (MPS) is designed to protect the important machine components from damage when serious abnormal situations occur. The system must be robust and reliable. It receives the interlock signals from the hardware devices and the timestamps signal from the timing system, performs the protection operations and reports the status to the operator interface layer. Take the vacuum system for example, the pressure in the vacuum chamber is continuously monitored via vacuum gauges and ion pump current. The vacuum valves are controlled by the machine protection system. An interlock action will close the proper isolation valves and stop the beam in case of anomalous pressure rises.

The system will be based on the FPGA (Field Programmable Gate Array) and adopts the hierarchical structure. It consist of one master nodes and several slave nodes. The master node receives and summarizes the signals from slave nodes of the injector, superconducting Linac, undulators, beamlines and experimental stations. The slave nodes control the executive devices, such as the vacuum valve and so on. The nodes are connected via optical fiber instead of cables.

CONCLUSION

The control system of SHINE will be a large scale real-time and interactive system with high reliability and stability. The high repetition rate and the potentially high data throughput present a great challenge in the control system that will require powerful fast feedback capabilities, large data processing capacities, and so on. The latest technologies and standards need be adopted.

REFERENCES

[1] K. Li and H.X. Deng, "Systematic design and three-dimensional simulation of X-ray FEL oscillator for Shanghai Coherent Light Facility", *Nuclear Instruments & Methods in Physics Research, A*, vol. 895, pp. 40-47, 2018, doi:10.1016/j.nima.2018.03.072

[2] H.H. Lv, Y.B. Leng, Y.B. Yan, *et al.*, "The high level application architecture of the control system for SHINE", *Nuclear Instruments & Methods in Physics Research, A*, vol. 908, pp. 167-171, 2018, doi:10.1016/j.nima.2018.08.047

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC4

HEPS CONTROLS STATUS UPDATE*

P. Chu†, D. P. Jin, G. Lei, C. H. Wang, L. X. Zhu
Institute of High Energy Physics, 100049 Beijing, China

Abstract

The High Energy Photon Source (HEPS) is a planned extremely low emittance synchrotron radiation based light source located in suburban Beijing which requires high precession control systems for both accelerator and beamline controls. This paper outlines the overall design for control systems, including equipment control, fast orbit feedback, machine protection, controls network, database, high-level application architecture, and physics applications. Early plans for beamline controls are also reported.

INTRODUCTION

An ultra-low emittance and high brightness 4th generation synchrotron light source, the High Energy Photon Source (HEPS) designed by the Institute of High Energy Physics (IHEP), will start its construction by the end of 2018. To reach the high goals listed in Table 1, especially a scale of the order of about 2500 magnets and similar number of diagnostic devices, it is necessary to have accurate installation, state-of-art equipment and high precision controls with intelligence. Therefore, the control systems are vital for the HEPS which includes not only traditional control architecture design but also quality control for the project. Also, the HEPS control system covers not only the accelerator but also the first 14 beamlines which will be constructed at the same time. To build such a complex accelerator based user-facility, it is necessary to have an overall complete design for the control systems.

The HEPS control system design and test environment setup progress which includes database work, and accelerator and beamline controls, as well as the quality control tools being developed at this early stage of the project will be described in this paper.

Table 1: HEPS Main Parameters

Main parameters	Value	Unit
Beam energy	6	GeV
Circumference	1360	m
Emittance	58 (<40 anti-bend)	pm·rad
Beam current	200	mA
Brightness	$>10^{22}$	Phs/s/mm^2/mrad^2/0.1%BW
Injection	Top-up	
Bunch structure	680 bunch， high brightness mode 63 bunch， timing mode	

DATABASE WORK

In the modern data era, it is essential to record every useful data and store the data systematically in persistent stores. Furthermore, applications to utilize the data should be developed. Due to the large database work, it is necessary to divide the entire database into many nearly independent database modules and connect them via API (Application Programming Interface) or services. This way the database modules can be developed independently by many institutes and avoid the complexity of a single database. A few database modules can be merged via links of either device ID which is based on the role the device is played in an accelerator, or equipment ID which is merely a unique QR code or RFID.

Based on IRMIS [1], as listed in Table 2, there are 17 database modules identified. At this stage of the project, *i.e.* the design and early implementation phase, databases such as Parameter List, Naming Convention, and Magnet have been developed to suit the project's current needs. In addition, colleagues from another IHEP facility, the China Spallation Neutron Source (CSNS) is collaborating with the HEPS team to develop a Logbook and Issue Tracking database and application for CSNS's early operation need. Besides the 4 database modules currently under development, a few others like Accelerator Model/Lattice, Physics Data and Machine State, and Work Flow Control/Traveler have been developed by colleagues for other projects which can be migrated here easily. The rest of the database modules listed in Table 2 will be developed at later times while they are needed. HEPS select MySQL as the database primary tool. Details for the three under-developed databases are described below.

Table 2: Planned Database Work

Parameter List	Logbook and Issue Tracking	Cable
Naming Convention	Maintenance/Operation	Security
Magnet	Inventory	Alarm
Accelerator Model/Lattice	Survey and Alignment	Machine Protection/Interlock
Equipment and Configuration	Work Flow Control/Traveler	MPS Postmortem
Physics Data and Machine State	Document DB	

Design Parameter Database

This is a database which stores the essential HEPS parameters for keeping track of different design versions. The database schema was based on ESS design [2] with necessary modifications to fit HEPS own needs.

Naming Convention Database

For a large accelerator project like HEPS, everything has to be named according to strict rules or it is hard to manage the names. The HEPS Naming Convention database provides such a systematic tool for generating names automatically according to rules which will be applied to both accelerator and beamline experiment instruments.

* Work supported by the Chinese Academy of Science, and the Chinese National Development Council.
† chuzm@ihep.ac.cn

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC4

Magnet Database

There are about 2500 magnets in HEPS with many types. The Magnet database provides a whole magnet lifetime data storage throughout from design, manufacture, test, operation and maintenance phases. At present, the design for data tables of design and several test methods has completed.

ACCELERATOR CONTROL

HEPS accelerator controls is EPICS-based, distributed systems. The system design principles are applying industrial standards, global timing system for both accelerator and experiments, and modularized subsystems for easy upgrade and maintenance. The overall accelerator control system architecture is shown in Fig. 1 with Device, Middle, and Presentation layers. The Device layer provides the control interface, such as μTCA, PXI or PLC, to devices. The Middle layer performs data assembly and persistence, and online analysis computation. Details for some control system components are described below.

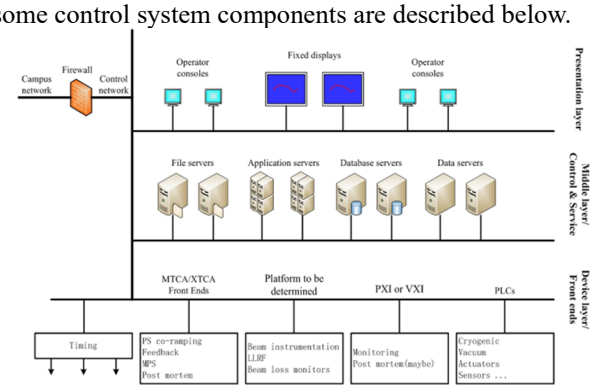


Figure 1: HEPS accelerator control system architecture.

Device Control

Presently HEPS selects the latest EPICS version 3 as the control system platform. The EPICS-based device control will choose mostly industrial standard with EPICS driver support. Besides fast communication networks for global timing system and fast machine protection system, the standard communication is through EPICS Channel Access (CA) protocol.

Fast Orbit Feedback

For the extremely low emittance requirement, a fast orbit feedback (FOFB) system is required to suppress orbit oscillations caused by ground vibration, magnet power supply ripple and other possible causes. The goal of FOFB is to reach 300 Hz – 1 kHz of bandwidth with 576 Beam Position Monitors (BPM) and 192 correctors in each transverse plane. As shown in Fig. 2, The BPMs and correctors are arranged to 16 nodes with each node as star topology, while the 16 nodes are connected in ring topology. The communication among the 16 nodes is bi-directional and through 10 Gigabit Global Links. For each node, the FOFB computation is done via Field-Programmable Gate Array (FPGA). Studies for better power supply response, computation algorithms are underway.

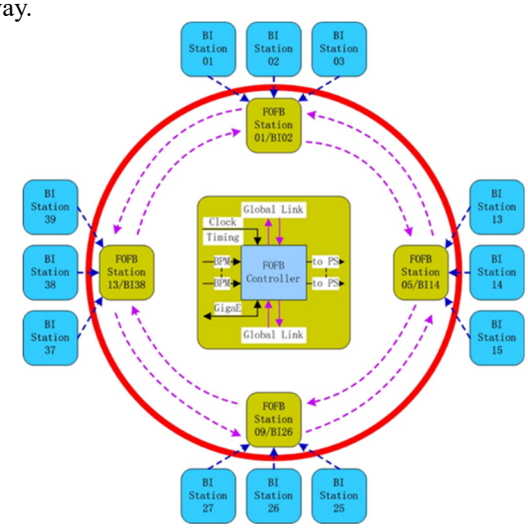


Figure 2: HEPS fast orbit feedback system architecture.

High-level Applications

As shown in Fig. 3, application flow can be divided into three layers: data, software API or services, and application GUI. For better software architecture and reusability, same functions appear in multiple applications should be converted to either regular callable APIs or service APIs in the middle layer. Furthermore, due to the nature of the functions, it is better to separate them in three groups so they don't mixed together and lose the flexibility: control system API, physics and general-purpose API, and machine learning API. The three API groups are also released independently as separate software packages. Details for these APIs will be described below.

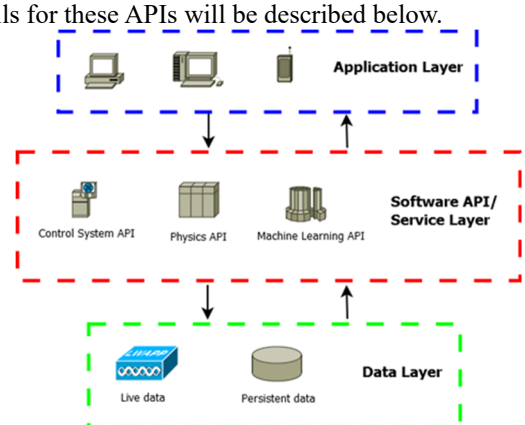


Figure 3: HEPS control system architecture.

Control System API

HEPS accelerator as well as optical beamline is based on EPICS control systems. APIs such as CA calls are packaged along with connection exceptions and operation loggings, so application developers do not have to go through tedious details. If there is any need for swapping with another control system, one can simply replace the EPICS wrapper with a different one. HEPS is considering CS-Studio API as the control system API platform.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Physics and General-purpose API

Physics API provides physics specific functions such as online model calls, and general-purpose API supports certain operations like parameter scan or correlation plot. HEPS is considering a Java-based Open XAL [3] toolkit includes not only definitive accelerator data structure but also quite comprehensive functions for most accelerators. On the other hand, beamlines can have a similar platform(s) for their physics needs.

Machine Learning API

As the artificial intelligence (AI) and machine learning (ML) is gaining momentum in many fields, it is also attracting attention for accelerators and user-facility related Big Data research. A Machine Learning API platform has been developed for accelerators to ease the use of some popular ML APIs such as Scikit-Learn and TensorFlow. For instance, APIs for pre-processing raw data prior to applying the actual computation algorithms, and the computation result showing in visual form for easy read should be provided. All these can be done with simple APIs to cut development effort. With such a platform, physicists can use Python, a popular scripting language, to develop quick data analysis applications. One can switch among ML platforms as well as algorithms for easy tests. The ML API will also be responsible for converting data format to suit many popular Big Data platforms.

BEAMLINE CONTROLS

As shown in Fig. 4, HEPS beamline control system architecture is similar to the accelerator controls'. For better manpower resources sharing, the HEPS optical beamline control is also handled by many accelerator controls experts. Also, it is not practical to have each beamline possessing its own database experts and handle all computing needs, for example. Many tools and platforms built for the accelerator can also be shared by the beamlines. Standards like naming conventions and EPICS supported devices are also shared.

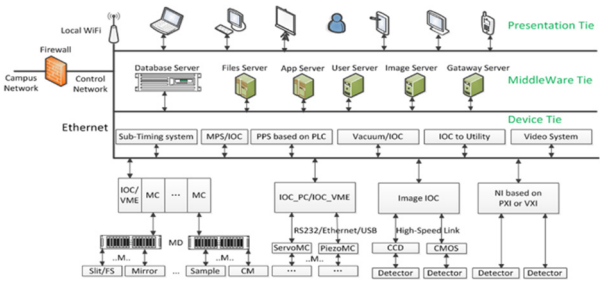


Figure 4: HEPS beamline control system architecture.

The beamline data may be much more structured than the accelerator data. Therefore, EPICS 7 which support complicated data structure is considered as the data protocol for packaging the beamline and experiment data. Still, the data structure has to be compatible for future mobile applications.

The data acquisition (DAQ) is considered along with the beamline controls. For the nature of fast DAQ and large amount experiment data storage requirements, online data reduction and analysis is essential for the beamlines. Standard data format should be chosen for compatibility with various data analysis tools and shareable among institutes.

PROJECT SUPPORT

To better integrate project data with technical data, HEPS business sector's procurement database and the Inventory database will be linked for better cost control and future operation maintenance utilization. The link between these two databases will be a unique equipment ID in the form of bar code, QR code, or RFID. Applications will be developed to take advantage of these joint databases. Other project databases include document server and personal folders may be on SharePoint platform, but still can be connected with MySQL databases with simple programming.

CONCLUSION

The HEPS control systems have been initially designed and a few tasks have been started. Modern technologies will be applied to the actual implementation. For the software shareable between accelerators and optical beamlines as well as experiment stations, collaborations are formed. Starting from project supports like databases and project management tools, the controls team is also design and development work in many areas. All the databases are saved in GitHub repository for easy collaboration access [4]. This overall modularized architecture design gives us the most flexibility and efficiency for development yet ensure high-quality and reliable products. Also, researches in new fields such as Big Data analysis have been started for such a future synchrotron light source.

ACKNOWLEDGEMENT

The authors would like to thank the Controls Group of CSNS, the HEPS Beamline Experiment Groups, and many other members of the IHEP Accelerator Centre colleagues for many fruitful discussions.

REFERENCES

[1] IRMIS, http://irmis.sourceforge.net/

[2] K. Rathsman *et al.*, "ESS Parameter List Database and Web Interface Tools", in *Proc. IPAC'11*, San Sebastian, Spain, Sep. 2011, pp. 1762-1764.

[3] J. Galambos *et al.*, "XAL Application Programming Structure", in *Proc. PAC'05*, Knoxville, Tennessee, USA, May 2005, pp. 79-81.

[4] Accelerator Database, https://github.com/AcceleratorDatabase/

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC5

!CHAOS GENERAL STATUS REPORT

Alessandro Stecchi, Claudio Bisegni, Paolo Ciuffetti, Antonio De Santis, Giampiero Di Pirro, Alessandro D'Uffizi, Francesco Galletti, Riccardo Gargana, Andrea Michelotti, Massimo Pistoni, Dario Spigone, INFN/LNF, Frascati, Rome, Italy
Luciano Catani, INFN - Roma Tor Vergata, Rome, Italy

Abstract

!CHAOS[1] (Control system based on Highly Abstracted and Open Structure) is now mature and is being employed in real operational contexts. A dedicated infrastructure, recently installed at the LNF Computer Centre, houses the framework and provides control services to different LNF installations. The !CHAOS native capability of fast storage, based on the use of a nonrelational database, has been finalized and tested with applications demanding high bandwidth. Thanks to its scalable design, the fast storage allows to accommodate multiple sources with sub millisecond timing. The EU (Execution Unit) node has also been delivered and turned out to be a "Swiss Army knife" for processing both live and stored data, inserting feedbacks and in general for correlating data acquired by the CU (Control Units) nodes. A key feature of the EU is a plugin mechanism that allows to easily integrate different programming and scripting languages such as LUA, C++, Python, also exploiting the ROOT framework, the well known scientific tool from CERN. A comprehensive description of the !CHAOS evolution, of its performances and of its use, both in scientific and industrial contexts, is presented.

INTRODUCTION

The !CHAOS project started with the ambition to create an innovative control framework exploiting software technologies developed for high performance web services and therefore capable to handle millions of users accessing the services and interacting with one another.

The framework was designed to be scalable and cloud aware and, as a result, suitable for application in many different contexts, beyond those required by the scientific community.

The right way to look at !CHAOS is as a SaaS that specializes in controls. In fact, we often refer to it with the neologism of *Control as a Service* (CaaS).

Moreover, !CHAOS ranks in that niche — still partially unexplored — between Control Systems and DAQ Systems. Indeed, the horizontal scalability of the framework allows fast storage for multiple data sources with sub-milliseconds timing, granting a centralized time synchronization among them of the order of milliseconds. This feature — embedded by design in the !CHAOS architecture — is fundamental to make cross correlations among many heterogeneous data and also opens to new applications such as diagnostic, maintenance and predictive maintenance of large scientific facilities and industrial plants.

Data handling is also greatly eased by the adoption of non-relational databases and BSON/JSON data representation throughout the system.

The !CHAOS framework is currently used both in industrial and scientific collaborations.

SYSTEM OUTLINE

Back-end Layer

The framework relies on back-end services which provide core services such as data caching and permanent storage. The back-end services are:

- the Distributed Object Caching (DOC) that continuously store the latest datasets representing all the physical and abstract entities acquired or computed by the system;
- the Distributed Object Storage (DOS) that enqueues the above datasets in a persistent database;
- the metadata storage that holds configurations and preferences data — both of the system itself and the elements under control — in a persistent database.

The key-features of all the services are that they (i) build upon non-relational logic and (ii) handle datasets consisting of BSON objects (which ultimately are blobs of bytes). As a result, different data structures can be directly stored as key-value datasets, thus without any extra data parsing or manipulation and the need of setting up many different table structures. Moreover, the inherent schemaless nature of non-relational databases, greatly speed-up, as in the case of DOS service, the writing of permanent data, for the benefit of the !CHAOS DAQ performance.

Each of the framework's core services is not binded to any particular software technology or vendor. Different commercial, or open source, software can be adopted — as explained below — to implement the back-end functionality, the choice depending on the specific needs of the context. On the contrary, it is essential that DOC, DOS and metadata storage services can run as multiple instances on multiple machines, which is a prerequisite for the horizontal scalability of !CHAOS.

The current release of !CHAOS employs Couchbase® for the DOC and MongoDB© for the DOS and metadata storage. Similar products have been successfully employed and other are currently under evaluation (such as redis and Cassandra).

Both Couchbase and MongoDB deliver distributed architecture and provide compute, storage, and processing workload partitioning to meet ever-changing requirements.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC5

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Abstraction Layer

On top of the back-end layer lies the abstraction layer which provides !CHAOS services and a common application interface to the framework core services (Fig. 1). !CHAOS services interface with the back-end through dedicated drivers so that, if you want to use different products, it is sufficient to replace the related interface drivers. The !CHAOS services are:

- the Data Service (DS) that manages the data flow to and from the DOC and DOS services;
- the Metadata Service (MDS) that manages storage and retrieval of metadata to and from the DOS service.

Practically, both DS and MDS act as *routers*: they dynamically dispatch and gather data to and from the nodes of the back-end, including, bandwidth optimization, workload balance and handling of back-end nodes failure. This design guarantees the scalability of the system and even enhances it, as the balancing logics exerted by the DS and MDS drivers can be optimized to meet particular needs.

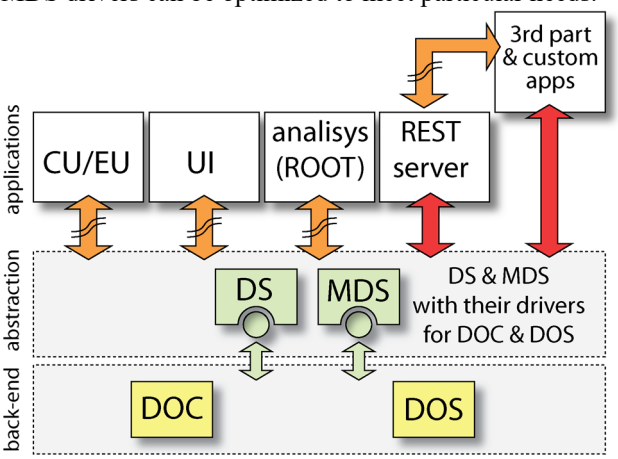


Figure 1: Layout of the !CHAOS framework.

The DOC & DOS services are implemented by Couchbase & MongoDB. The DS & MDS adapt to different back-end services by changing the corresponding drivers. In the applications layer, the segmented arrows indicate that the node can operate from outside the !CHAOS cloud, whilst the countinuous ones indicate that — for optimal performance — the node has to be within the cloud.

Applications Layer

Above the abstraction layer there is the applications layer, where different kinds of nodes perform the functions of raw data production, calculation and correlation, access to data, system metric and access from external applications. A short description of the !CHAOS main nodes is presented here below.

Control Unit When it comes to control systems, the tasks of acquisition and actuation of devices are among the most important because they are at the heart of the whole management of a plant. The CUs is the node that continuously acquire data from a device and operate it upon commands or given conditions. In !CHAOS, a device is set out by a virtualization process that — starting from a physical object — concludes with the definition of a set of meaningful variables (dataset) that fully describe it from the vantage point of the control and a set of actions you want to be able to perform on it (commands).

Each device is managed by a dedicated CU that continuously acquires it, refreshes its dataset, associates it with a timestamp and pushes it — through the DS — into the DOC and DOS. The frequencies of those pushes can be set independently. The CU can also perform advanced data processings on the spot and fill the dataset with derived quantities or complex functions result (FFT, pedestal calculation, mobile average, etc...).

Each remote command coming from a UI/EU/Control application/MDS is enqueued in a priority queue and executed accordingly. For every command the programmer can define four different handlers that encode four different states of a command: start, acquire, feedback and end. Typically, a command is in start at the beginning, then it loops at a given frequency between acquire (device acquisition) and feedback (device actuation) and exit on end.

Execution Unit The Execution Unit node (EU) is a strong point of !CHAOS being a very convenient way to inject processes in the system and have them execute in an standardized and managed manner. An EU is structurally similar to a CU with the difference that it doesn't access hardware but data already acquired, instead. An EU can receive commands, issue multiple commands to multiple EUs and CUs and access both live and stored data. These capabilities make it a very versatile tool that can execute automatic feedbacks, make correlation among live and stored data, execute background analysis on big quantities of stored data, build and issue macro-commands, implement complex procedures and so on.

In the current version of !CHAOS the EUs can access data through the DS from DOC and DOS so that they are very similar to batch processes. Developers are working to implement a stream processing mode where EU will be connected among other nodes (CU, EU) through input and output ports realizing this way processing pipelines where EUs are able to check-in to one or more CUs or EUs and directly receive continuous streamings of live datasets from them, which will dramatically increase the bandwidth and — as a result — their responsiveness. Currently an EU algorithm can be expressed in C++ native language, CERN ROOT or lua scripting. Next developments envision to include python and nodejs scripting. To support streaming processing and total abstraction of EU processing algorithms we'll support a graphical tool where the user can build its processing pipeline simply connecting "boxes" (CU-EU, EU-EU) connecting input terminals of a EU to output terminals of a CU, EU.

User Interface The User Interface node (UI) is where data are presented and user commands are issued. UI nodes can query the MDS for information about dataset structure and command set of any element, which allows for a dynamic adjustment of the control windows.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC5

REST Server The REST server maps !CHAOS C++ APIs to HTTP/REST JSON APIs allowing external applications to access I/O, control and administration services of the framework. The !CHAOS BSON internal data representation is well suited for the implementation of WEB services based on JSON notation (since BSON is just a standardized binary serialization of JSON). Any external application can act as a CHAOS node (CU, EU, UI) according to needs. Most popular development environments have HTTP support for binding and integration of !CHAOS REST API. Currently !CHAOS has bindings for MathLab, LabVIEW, JavaScript, Python (basic), which allows users to write their own control, analysis or presentation application using the preferred environment.

Analysis Framework The CERN ROOT analysis framework with native C++ !CHAOS APIs is available both as a standalone application and as a plugin able to execute scripts from within an EU. The combination of a HEP well-known analysis tool with a Control System with DAQ capability, allows users to develop new complex and efficient control and data analysis algorithms.

HARDWARE INFRASTRUCTURE

As mentioned above, the ideal type of a !CHAOS installation is that of an infrastructure fit for use as a *cloud* and therefore offer a *CaaS* service to both local and remote users. Our goal was therefore to set up an installation able to give a boost to the framework, leading it beyond the prototyping and testing phases.

We decided to utilize all our financial resources in laying solid foundations for the infrastructure and use — on top of these — hardware that can be updated later.

We obtained a dedicated room from our Computer Centre with buffered power supply and air conditioning and acquired a 32 ports core switch CISCO Catalist 4500 X and six 48 ports CISCO C2960X for the network backbone.

Then we populated 8 racks with disused hardware obtained by courtesy of CERN, totalling 472 logical cores, 824 GB RAM and 500 TB HDs.

This hardware came out of production at the end of 2013, which means that, despite being discontinued, it is still able to fulfill the function of a pilot installation.

The infrastructure houses:

- a production cluster made of 8 servers (each with 24 logical cores, 72 GB RAM);
- a pre-production cluster made of 16 servers (each with 8 logical cores, 16 GB RAM);
- a development cluster made of 16 servers (each with 8 logical cores, 16 GB RAM);
- a services cluster made of 3 servers (each with 8 logical cores, 8 GB RAM);
- 6 spare machines (each with 16 logical cores, 16 GB RAM) not configured;
- 6 NetApp storage servers with multi 10Gb/s interfaces and ~500 TB of available disk space (NFS exported).

OVirt was chosen as a hypervisor because it is an open-source product and proved — in other projects — to be reliable and scalable, features essential for the !CHAOS framework. Two hypervisors have been installed: one for production and services and one for pre-production and development so as to keep separate the management of the two groups of physical and virtual machines.

The guest machines are installed with different Linux distributions: Ubuntu Server for the MongoDB nodes and CentOS for all the other services.

The infrastructure uptime is 100% since its start-up in October 2017.

As described below, the infrastructure already provides control services to different LNF scientific installations. We are also going to open it to technological installations, small laboratories and external users working on the various lines of our accelerators.

SOFTWARE MANAGEMENT

!CHAOS is a project released as opensource software under the EUGPL 2.0 license. The framework and all its processes are entirely developed in C++ (CXX11 & C98) using CMAKE (> 3.0) as build system. The supported compilers are gcc (≥4.8) and llvm. !CHAOS compiles on most Linux distributions and MacOS.

Continuous Integration

The adopted agile workflow envisages that new developments add features and/or fix bugs, namely the last stable release must always be an improvement respect to the previous one.

To guarantee this simple — but fundamental — assumption, we have created a set of top-down non-regression tests, spanning from UIs, CUs and EUs software up to lower level services.

The source code is stored on the INFN gitlab [2] server and the whole continuous integration workflow is handled by many *runner* machines that allow you to build and execute all the tests several time a day.

Each time a git merge request is issued, a set of scripts — pipelined by gitlab — executes all those tests for the different supported OSs. Should any test fail, the merge request is rejected, otherwise the code is merged.

For the static analysis of the code, we use suitable wrappers that are called at compilation time and generate reports. Those reports are then automatically uploaded on the cloud services *Coverity Scan* and *SonarCloud* which provide an excellent static analysis of the source code (Coverity Scan) together with a more in-depth analysis (SonarCloud) and permit to trust the quality of the written code. By adopting a continuos integration method, that automatically supervises code changes, dramatically enhances productivity also making possible a truly concurrent development.

SYSTEM PROFILE

Regarding the performance of mongoDB on the pre-production infrastructure, the system showed no evident bottlenecks. Since the available hardware is not updated and has limited resources, as the number of MongoDB shards

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC5

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

increases, the tests tend to a limit that prevents verifying the real potential of both architecture and back-end.

Nevertheless, before reaching these limits, it has been verified that the throughput increases proportionally to the number of shards.

It seems evident that a consistent measure of the system performance will be possible only by utilizing physical machines of a suitable rank.

Measurements

The frequency of I/O accesses it has been measured in a minimal !CHAOS installation on a server Intel-i7 - 16GB - 256 SSD - 3.6 GHz, composed by one Couchbase community-3.1.3 (as DOC) and one MongoDB 3.4.15 (as DOS), provided as docker nodes, plus a MDS node.

The measure consists in accessing the DOC & DOS services through the DS, from a minimal client written on purpose. The client mimes a producer node (CU, EU, Analisys) on pushes and a consumer node (EU, UI, Analisys) on pulls. All the measures are calculated as an average of 1000 accesses and repeated for different number of clients (in different thereads) running in parallel (Figs. 2 and 3).

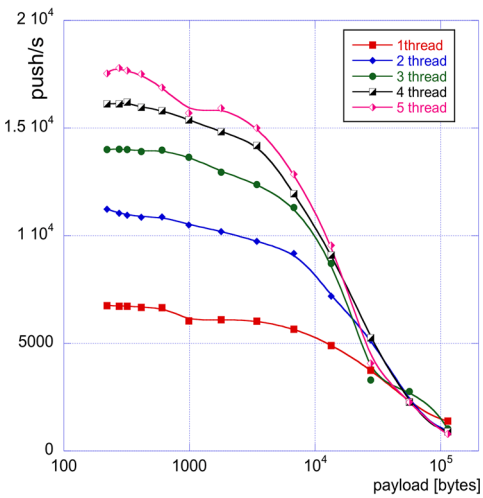


Figure 2: Push frequency vs. size of the payload at different numbers of threads.

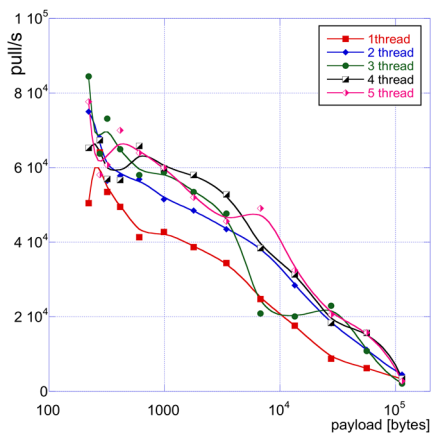


Figure 3: Pull frequency vs. size of the payload at different numbers of threads.

It can be noticed that the system is able to sustain the throughput as the number of threads increases. For higher numbers of threads, the load of the server gets too high and invalidate the measure.

By the end of November 2018, with the support of MongoDB Inc. Team a pre-production setup of MongoDB on physical machines will be arranged and will allow a quantitative measure for a multi-node configuration.

USE CASES

Beside the use of !CHAOS as control system in pilot installations at INFN-LNF: DAFNE transfer line, BTF, accumulator orbit [3]. In the following two paragraphs we present two further use cases with completely different objectives which use !CHAOS DAQ and analysis features.

Scientific

An independent data acquisition setup has been designed and realized in order to implement the fast luminosity monitor in view of the DAFNE future physics runs. Besides the total instantaneous luminosity the new diagnostic measures also the Bunch-by-Bunch luminosity. The following description focuses the attention on the systems engineering (for the scientific part refer DAFNE Luminosity Monitor [4].

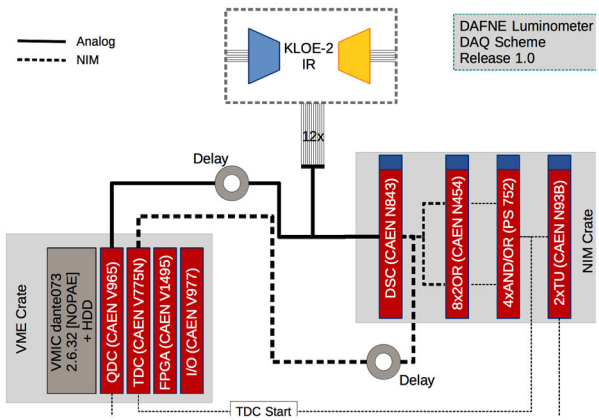


Figure 4: DAQ schema.

The new DAQ subsystem (Fig. 4) is composed by a 16 channels TDC (CAEN775N), 16 channels QDC (CAEN977), and a FPGA (CAENV1495) with few registers programmed as counters. A !CHAOS CU, running on an old VME Controller (VMIC linux 2.6.30), controls and acquires those devices at variable rates from 0 to 3KHz (depending on the collisions rate). The online computation of the Bunch-by-Bunch luminosity is performed by a ROOT application analysis connected with !CHAOS that correlates the data coming from the luminometer DAQ system with the data coming from the bunch charge monitor (acquisition 4096 points of a Tektronix scope).

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEC5

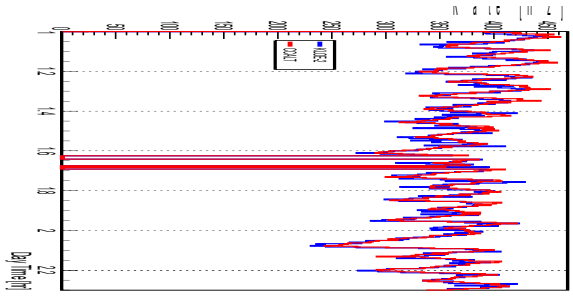


Figure 5: New (red) vs legacy luminometer (blue).

To monitor and check the results produced by the new DAFNE luminosity monitor !CHAOS also acquires the legacy luminosity monitor (Fig. 5), which updates every 15 s, and on-demand can also store beam camera frames at rate of 70 fps (Basler 640x480 8 bit) as overall cross check. All the correlations and checks of unrelated devices, without a common trigger or time, are made possible because the centralized !CHAOS infrastructure guarantees a time synchronization of the storicized datasets in the range of few milliseconds.

The possibility to have available many data coming from multiple sources that can be plotted and correlated each in real time, and the possibility to also control most of them greatly improved the productivity of the scientist team.

Industrial

INFN together with a world leader in the design and manufacture of automatic machines for the processing and packaging of pharmaceuticals, cosmetics, food, tea and coffee, is involved in a project named mAxima. The aim of this project is to setup a *cloud aware* infrastructure to acquire data from production line machines and perform predictive diagnostic to optimize their working point and prevent unwanted stops.

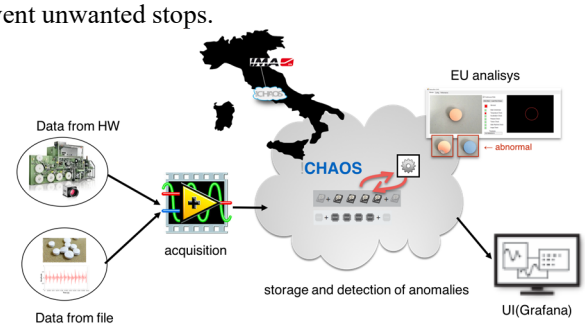


Figure 6: mAxima project: schema of DAQ.

The company provided to sensorize one of their package machines to measure: acceleration, temperature and pressure of some parts, and to acquire the images of pills that are going to be packaged. INFN provided a dedicated !CHAOS infrastructure suited to host processing predictive algorithms and their data flow (Fig. 6). Moreover created a !CHAOS support for Grafana [5]: an environment to create impressive WEB dashboards (Fig. 7). Since the real packaging machine was often not available, a synthetic flow of data were produced by a simulator able to inject respectively trends of errors in acceleration, pressure, temperature and different shapes of pills with zero or more defects, thorough this simulator we also tried algorithms to forecast failures and categorize images of bad and good pills.

Figure 7: Grafana WEB dashboard.

Currently we have hosted at INFN-LNF the infrastructure that is able to store the flow of data and produce real time forecast of failures of the given subset of parts, all these data are presented in a customizable WEB dashboard.

CONCLUSION

!CHAOS is a reality and has evolved from a mere Control System to a flexible framework offering also:

- a native DAQ system for processes with timing > 100 µs;
- a computing framework suitable to process data — tagged evenly— and produce feedbacks;
- a virtually unlimited horizontal scalability in terms of storage, memory and processing power;
- a cloud aware Control as a Service platform, suitable for different contexts (scientific, industrial, social, educational, consumer's).

REFERENCES

[1] L. Catani *et al.*, "Introducing a New Paradigm for Accelerators and Large Experimental Apparatus Control Systems", *Phys. Rev. ST Accel. Beams*, vol. 15, pp. 112804, 2012.

[2] https://about.gitlab.com/

[3] C.Bisegni *et al.*, "!Chaos Status and Evolution", in *Proc. IPAC'15*, Richmond, VA, USA, May 2015. doi: 10.18429/JACoW-IPAC2015-MOPHA046

[4] A. De Santis *et al.*, "DAΦNE Luminosity Monitor", in *Proc. IPAC'18,* Vancouver, BC, Canada, Apr.-May 2018, pp. 338-340. doi:10.18429/JACoW-IPAC2018-MOPMF089

[5] https://grafana.com

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP01

EtherCAT DRIVER AND TOOLS FOR EPICS AND LINUX AT PSI

D. Maier-Manojlovic, Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

Abstract

A combined EPICS/Linux driver package has been developed at PSI, to allow for simple and mostly automatic setup of various EtherCAT configurations. The driver is capable of automatic scan of the existing devices and modules, followed by self-configuration and finally autonomous operation of the EtherCAT bus real-time loop. Additionally, the driver package supports the user PLC to manipulate EtherCAT data in real time, implements fast real-time (single cycle) slave-to-slave communication (skipping EPICS layer or PLC completely), features guaranteed one-shot trigger signals otherwise not supported by EPICS and much more.

INTRODUCTION

For the modules and devices equipped with the EtherCAT bus interface [1], a general, real-time software interface was needed for the integration in the existing accelerator control system, both for existing facilities like SLS (Swiss Light Source) and HIPA (High Intensity Proton Accelerator), and for facilities and systems being built at the time this document was created, such as SwissFEL [2] (Swiss X-Ray Free Electron Laser).

First, we have tested the existing solutions, both related and unrelated to our controls system of choice, EPICS. Unfortunately, none of the existing commercial and non-commercial solutions we have reviewed and tested was able to cover and satisfy all of the requirements for the EtherCAT support at PSI.

CONCEPTS

Providing full support for such a wide range of systems and applications in a single package presented a problem since not every requirement or possible usage scenario could have been satisfied with a single piece of software.

EPICS control system support requires its own type of dedicated device support driver. Unlike its kernel counterparts, EPICS driver has to run in Linux userspace, since EPICS system itself is a userspace application. Aside from EPICS, the system has to support other types of applications.

To make things more complicated, the applications that are supposed to use the system are running in both userspace and kernelspace. This, of course, requires distinctively different structure of the supporting interfaces and practically double the work needed to create the system and maintain it later. Real-time applications can be created to run in either userspace or kernelspace, which in turns mean at least two separate local APIs had to be created.

EtherCAT Data Addressing

To describe an address of a given EtherCAT data entry, the following IDs have to be included:

- *master number* (since there can be multiple masters running on the same host),
- *domain number* (domain is an arbitrary, user-defined collection of PDO (Process Data Object) entries sharing the same buffer memory, EtherCAT packages and network exchange frame rate),
- *slave number* (slave is simply another name for an EtherCAT Module),
- *synchronization manager number* (synchronization managers, also known as SyncManagers or SMs, group EtherCAT PDO objects by their exchange direction (input/output) and other, manufacturer or end-user defined criteria),
- *process data object number* (process data objects, or PDOs, group entries by some arbitrary purpose defined by the manufacturer of the EtherCAT module, or if a module supports it, by end-users) or process data object entry number (process data object entries, or PDO Entries, hold the actual data exchanged by the module).

To solve this problem, we have devised a new addressing schema for EtherCAT data, as depicted in Fig. 1.

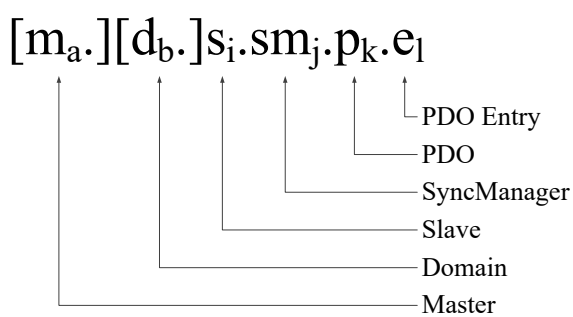


Figure 1: New schema for EtherCAT data addressing.

Master ($\mathbf{m_a}$) and domain ($\mathbf{d_b}$) can be omitted when a=0 or b=0, i.e., when the first, default master and domain are used. Similarly, the PDO entry ($\mathbf{e_l}$), PDO ($\mathbf{p_k}$) or SyncManager ($\mathbf{sm_j}$) can be omitted as well (in that order), when the user wants to address multiple entries included in a larger parent container, instead of a single entry.

The possible modifiers or addressing modes (as of v2.0.6) are:

- [.o<offset>] - **forced offset** (in bytes). This allows shifting of the starting address of the PDO entry in the buffer, effectively allowing for partial reading of the entries or using certain "tricks" to reach otherwise unreachable or hard to reach content
- [.b<bitnr>] – **forced bit extraction**, allows extraction of single bits from any larger PDO entry data, regardless of its original type or length
- [.r<domregnr>] – **domain register addressing**, replaces address modifiers *s, sm, p* and *e*, using rela-

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP01

tive entry addressing inside a domain instead. This allows end-user to simply address registers inside a module, SyncManager or PDO without having to remember or know all parts of the EtherCAT address, or to know the internal structure of a module

- [.lr<entryrelnr>] – **local register addressing**, replaces any (group) of the address modifiers s, sm and p, allowing for local relative addressing of all entries inside a slave, inside a SyncManager, or inside a PDO regardless of their actual parent container or containers
- [t<type>] or [t=<type>] provides means for **forced typecasting or type override**, changing the default type of the data entry when applied. Many typecasts are provided for this purpose, such as *int/uint* (8-, 16-, 32-, 64-bits), and also *float*, *double*, *BCD*, etc.
- [.l<length>] – **length modifier**, in bytes. Used primarily to define the length of *stringin/stringout* EPICS records, but can be used for any other buffer extraction, also

EPICS SUPPORT

Since at PSI the EPICS control system is almost exclusively used for the accelerators and device control, integrating EPICS support was a top priority.

As an example of a typical system controlling EtherCAT components, the EPICS Core is running on the Ioxos IFC 1210 Board [3], equipped with two separate Ethernet interfaces, a PowerPC P2020 CPU and the VME Bus backplane. The operating system used on these systems is a custom built Linux with the appropriate PREEMPT-RT patch.

Since EPICS has its own interface for device drivers, a special EPICS userspace driver had to be developed, using high priority real-time threads for the control loop. Without the real-time capabilities provided by the PREEMPT-RT Linux, timing and execution of the control loop would be less reliable and hence not real-time capable. However, if somewhat increased level of jitter is acceptable, the EtherCAT driver can run on a non-real-time Linux, i.e., without the PREEMPT-RT Patch installed.

Another problem that we have to solve was the fact that EtherCAT modules (slaves) are not always accepting the write values –for example, a write request in a certain cycle may fail for a number of reasons, and that means that the Ethernet packet on a return trip may contain register values which differ from the values stored in the write buffer of the driver.

This means that not only the refreshed read values, but also the write values has to be transferred back to the write buffer in order to overwrite the obsolete and potentially not matching values at the end of every cycle.

Yet, the newly received write values, unlike new read values, cannot be simply copied over the old values in the buffer, since that would effectively overwrite the new write request values which were already accepted from EPICS or other clients since the beginning of the last cycle. To solve this, a multithreaded double-buffering with the bit granularity write-mask for write requests was implemented (Fig. 2).

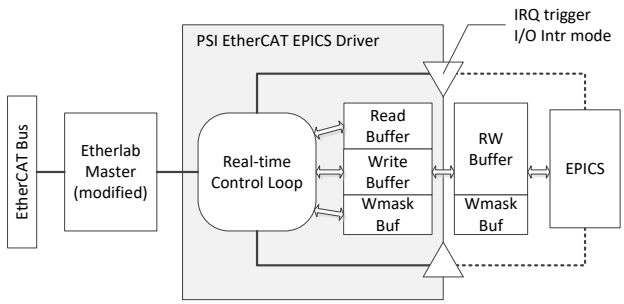


Figure 2: PSI EPICS EtherCAT driver structure.

EPICS Records

All EPICS records use a single driver type (DTYP) called **ecat1** (PSI multi-client and EPICS EtherCAT driver) **ecat2** (PSI pure EPICS EtherCAT driver). Scan rates for records can be set to any valid EPICS scan rate (from parts of a second to multiple seconds), including *Passive* and *I/O Intr*.

Register Values Typecasting

All of the extended addressing modes and modifiers, including typecasting (t=type in Input or Output field of a record), can be used in EPICS as well. Almost all combinations of provided typecasts and modificators are allowed. Using this feature it is possible to achieve high level of flexibility even in EPICS, with its rigidly defined record types.

For example, it is possible to extract a single bit from an *mbbi* record without having to create additional *calc* records, or having to recalculate the complete bit field extracted from a record of any length. Type override and other modifiers can be used for all record types, including array-type records (*aai*/*aao*). Length modifier (.l<length>) is used to define the length of the string for string-type records (*stringin/stringout*).

GENERAL SUPPORT

To support the local and remote applications wishing to connect to and use the EtherCAT hardware, the standalone, multi-client version of the driver package was created (ecat1). In this package three different subsystems have been developed in order to allow application developers a highly flexible way to access the EtherCAT slaves and their data:

- Kernelspace API
- Userspace API
- Modules, SMs, PDOs and PDO entries as *procfs* directory/file structure

Additionally, the driver automatically constructs and maintains *procfs* trees throughout its operation, and takes care of double/triple triple-buffering and write-masking

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP01

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

process needed for data exchange with the client applications. Description of the each of the access modes is presented below.

Kernelspace API

Kernelspace API (Fig. 3) is a set of functions providing the easy access to driver control loop parameters and EtherCAT data entries.

The driver provides an internal real-time control loop for buffering and exchange of TCP packets over Ethernet, Timing of the control loop is based on the host high resolution timers, but can be driven by an external source as well, such as a timing system input.

Multiple kernelspace applications and/or drivers can use the API.

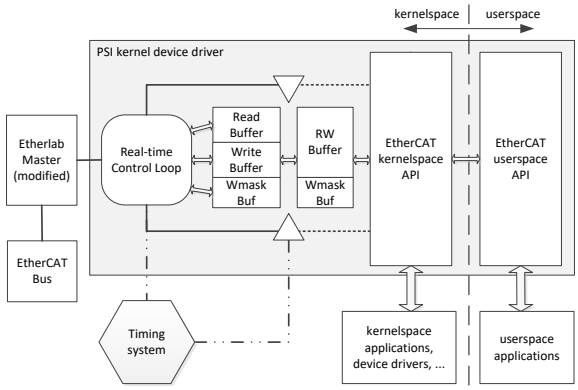


Figure 3: PSI EtherCAT driver structure.

Userspace API

Userspace API (Fig. 3) is an API, i.e., a set of functions providing (almost) the same functionality found in the Kernel API. The functions library can be used statically or dynamically with userspace applications as needed.

One of the Userspace applications that had to be covered is EPICS itself, so a native EPICS driver using this API has been developed as a part of the package. When this EPICS driver is used, both EPICS and other client applications (running in both kernel- and userspace) can be used concurrently.

EtherCAT Data in "procfs tree"

To allow even more applications to access the EtherCAT data, but without the need for an API or a dedicated remote server and client, we have developed the concept of "*procfs trees*" to represent the tree structure of EtherCAT modules and their components. *procfs trees* are a series of virtual "directories" and "files" constructed on-the-fly by the drivers in the Linux host *procfs* file system.

Each directory represents some kind of a parent container, such as a slave, a SyncManager, a PDO, a domain or a master (Fig. 4). Each virtual file in these directories represents either a direct representation of an EtherCAT PDO entry, or a utility file representing the data about the system or about the containers present.

There is also a special *cmd* entry provided in the *procfs tree*, allowing a CLI or application to interactively talk to the driver and sending commands (for example, add entry, add PDO, add slave entries, list data, etc.). Also, entry data can be read or changed using the CLI as well.

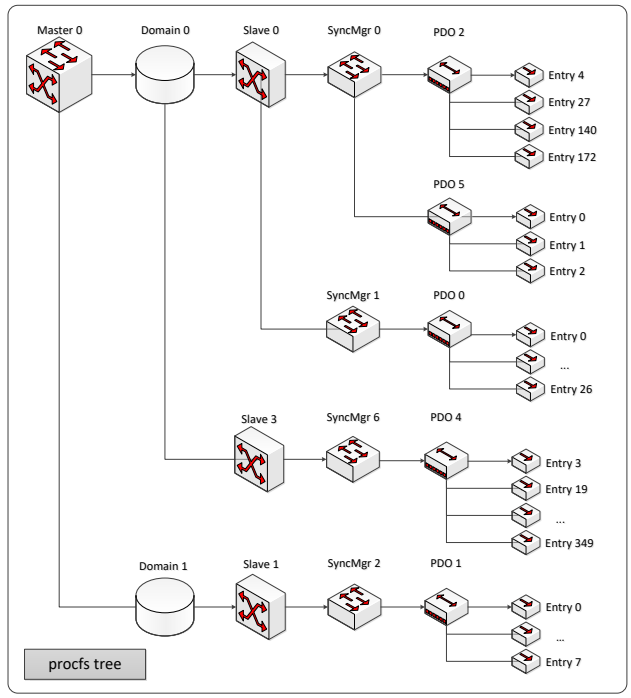


Figure 4: *procfs tree* general structure.

EXTENSIONS AND UTILITIES

The PSI EtherCAT support package offers several extensions and tools. The most important ones are described below.

Slave-to-slave Communication

It is often the case that the data on the EtherCAT bus has to be transferred from one EtherCAT device or module to another, preferably in real-time. For this kind of communication, two different directions of transfer can be observed, *upstream* and *downstream*.

Upstream slave-to-slave communication is transfer of data from a module further away on the EtherCAT bus from the master to a module closer to the master. Downstream communication is the transfer from a closer module to one further "down the stream" from the master. The stream in this case represents the path an EtherCAT TCP packet is travelling, and its direction remains constant as long as there are no physical changes in the bus configuration and modules present.

From real time point of view, this communication is highly deterministic, yet not identical – downstream communication (send on one module, receive on another) can, theoretically, be done in the same bus cycle, hence costing exactly zero bus cycles to execute. Upstream communication, due to the way TCP packets are handled by the EtherCAT, will take exactly one bus cycle to complete.

We have decided to implement the slave-to-slave communication (*sts*) with constant cost of completion, in this

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP01

case, exactly one bus cycle for both upstream and downstream communication requests.

In EPICS, *sts*-communication transaction requests can be inserted as follows:

```
ecat2sts <source> <destination>
```

For example:

```
ecat2sts r8 r0
ecat2sts r2.b0 r0.b6
ecat2sts s2.sm0.p1.e0 s1.sm0.p1.e0
ecat2stss3.sm3.p0.e10.b3 s4.sm2.p1.e0.b7
```

As can be seen in examples above, any valid addressing mode and/or modifier can be used for source and destination. API access is done by calling a function to register a transaction request, but the addressing remains the same.

Support for Programmable EtherCAT Modules

The PSI EtherCAT drivers and utilities also support setting up and live programming of programmable EtherCAT modules and devices, such as, for example, EtherCAT network bridges (EL6692, EL6695), motor controllers, and so on.

From EPICS, any module can be programmed by using *ecat2cfgslave* set of commands, for example:

`ecat2cfgslave sm <arguments…>` - configures one Sync Manager for the given slave.

`ecat2cfgslave sm_clear_pdos <arguments…>` - clears (i.e., deletes) all PDOs for a given Sync Manager (SM).

`ecat2cfgslave sm_add_pdo <arguments…>` - adds a PDO with index pdoindex to a Sync Manager.

`ecat2cfgslave pdo_clear_entries <arguments…>` - clears (i.e., deletes) all PDO entries associated with the given PDO.

`ecat2cfgslave pdo_add_entry <arguments…>` - creates a new PDO entry and associates it with the given PDO.

Programmable network bridge EtherCAT modules, such as EL6692 or EL6695, have their own, simplified commands for programming entries:

`ecat2cfgEL6692 <netbridge_nr> in/out <numberofbits>`

Support for User PLC in Dynamic Libraries

During the development and testing of EtherCAT drivers, the need for user-defined PLCs has been identified. In addition to slave-to-slave communication mentioned earlier, users sometimes wanted to be able to directly manipulate the EtherCAT data themselves.

A possibility to create a C/C++ based PLC, placed in a user-created dynamic library that is automatically detected and loaded, if needed was added.

This way, there is no need to change and recompile the driver directly, user only has to create a dynamic library with certain functions present, and it will be called (with a callback-function to prevent real-time jitter), inside each EtherCAT cycle.

Now the users have the possibility to directly manipulate EtherCAT module values in each cycle, in real-time, without the need of a slow, interpreted script based solution, or doing it using EPICS record, which is also an order of magnitude slower than our solution.

CONCLUSION

In this paper, we have presented the PSI EtherCAT driver packages and described the package components.

The system using ecat2 driver is already successfully used at PSI for several years, for all systems controlling EtherCAT modules from EPICS and from real-time applications. The other version of the driver, ecat1, is currently undergoing final testing at PSI under full-load and real-life conditions.

As is usual with such systems, it is to be expected that changes will be made to this package in the future to accommodate needs and new requirements of expanding number of users of the system, the existing features will be extended and streamlined and the new features and components will be added.

REFERENCES

[1] Beckhoff GmbH, http://www.beckhoff.de/.

[2] Paul Scherrer Institut (PSI), SwissFEL, http://www.psi.ch/media/swissfel/.

[3] Ioxos Technologies, IFC 1210 – P2020 Intelligent FPGA Controller, http://www.ioxos.ch/.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP02

BLISS – EXPERIMENTS CONTROL FOR ESRF EBS BEAMLINES

M. Guijarro*, A. Beteva, T. Coutinho, M. C. Dominguez, C. Guilloud, A. Homs,
J. Meyer, V. Michel, E. Papillon, M. Perez, S. Petitdemange
ESRF The European Synchrotron, 71 Avenue des Martyrs, 38000 Grenoble France

Abstract

BLISS is the new ESRF control system for running experiments, with full deployment aimed for the end of the EBS upgrade program in 2020. BLISS provides a global approach to run synchrotron experiments, thanks to hardware integration, Python sequences and an advanced scanning engine. As a Python package, BLISS can be easily embedded into any Python application and data management features enable online data analysis. In addition, BLISS ships with tools to enhance scientists user experience and can easily be integrated into TANGO based environments, with generic TANGO servers on top of BLISS controllers. BLISS configuration facility can be used as an alternative TANGO database. Delineating all aspects of the BLISS project from beamline device configuration up to the integrated user interface, this paper will present the technical choices that drove BLISS design and will describe the BLISS software architecture and technology stack in depth.

BLISS PROJECT SCOPE

The BLISS project brings a holistic approach to synchrotron beamline control. The scope of the BLISS project goes from hardware control up to the end-user interface. BLISS does not include data analysis, which is devoted to another software package at ESRF called *silx* [1].

CONFIGURATION

The BLISS configuration entity, a.k.a **Beacon**, aims to provide a complete and centralized description of the entire beamline. BLISS distinguishes between 2 kinds of configuration information: either configuration is **static**, as a stepper motor axis *steps per unit*, ie. the configuration information will not change over time once the object is configured ; or the configuration is subject to change, like a motor velocity for example. In this case, this is called a **setting** and settings are all backed up within the redis database [2].

Static Configuration

The **static** configuration consists of a centralized directory structure of text based files, which provides a simple, yet flexible mechanism to describe BLISS software initialization. The *YAML* [3] format has been chosen because of its human readability (cf. Figure 1).

BLISS is an object oriented library and its configuration follows the same model. Objects are identified in the system by a unique **name**. BLISS reserves the YAML key *name* as the entry point for an object configuration.

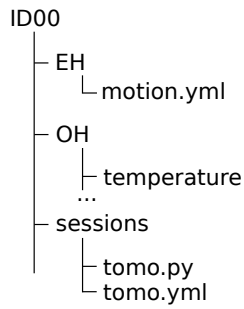


Figure 1: YAML tree example.

Each particular BLISS class may choose to profit from the BLISS configuration system. The BLISS configuration is powerful enough to describe not only control objects like motors, counter cards or detectors but also user interface objects like sessions or procedures.

The following YAML lines exemplify motor and session configurations:

```yaml
# motion.yml
class: IcePAP
host: iceid311
plugin: emotion
axes:
- name: rotY
  address: 3
  steps_per_unit: 100
  acceleration: 16.0
  velocity: 2.0

# tomo.yml
class: Session
name: tomo
config-objects: [rotY, pilatus, I, I0]
setup-file: ./tomo.py
measurement-groups:
- name: sensors
  counters: [I, I0]
```

Settings

Beacon relies on *Redis* to store **settings**, ie. configuration values that change over time, and that needs to be applied to hardware equipments at initialization time. This allows to be persistent across executions. Taking again the motor example, if a motor velocity is set to a certain amount from a BLISS session, when it is restarted the last known velocity is applied to the axis. Settings values use *Redis* structures: settings can be hashes (mapped to a Python dictionary), lists, and scalar values. BLISS offers a

* guijarro@esrf.fr

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP02

`bliss.config.settings` helper submodule to deal with Beacon settings directly from the host Python program.

Beacon Server

A client can access the remote configuration through a service provided by the *beacon-server* which, on request, provides a complete or partial YAML configuration. The BLISS library provides a simple API for clients to retrieve the configuration from the server as a singleton **Config** object:

```
>>> from bliss.config.static import get_config
>>> config = get_config()
>>> rotY = config.get('rotY')
>>> rotY.position()
23.45
```

The *beacon-server* is also responsible of managing a *Redis* server instance and optionally a **configuration web application** (cf. Figure 2).

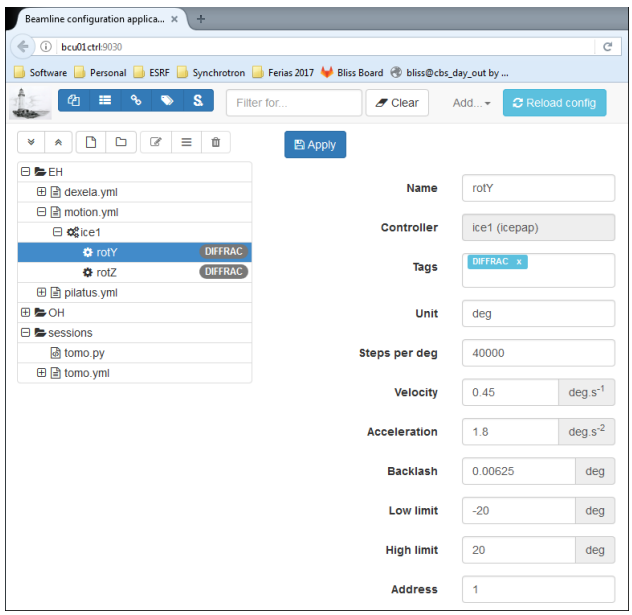


Figure 2: Beacon configuration tool.

TANGO Database

Additionally, Beacon can also provide an alternative implementation of the TANGO Database [4] service based on the same YAML configuration structure.

HARDWARE CONTROL

To the hardware control point of view, challenging points are to support an increasing number of devices to fit the experimental needs of scientists and to be able to deal with increasing complexity of devices (synchronization or communication protocols for example)

Generic Controllers

To achieve these goals, BLISS provides **generic controllers** which implement the complex, logical part of the control for each main class of devices encountered and leave to the developers the task to implement only the specific part of the control.

This approach is very efficient for instruments with a great variety of models. The price to pay is an increase of the complexity of the generic controllers. But this strategy is not exclusive: some controllers are, at least for now, not generic; either because we have no common behavior between different models or because it is much simpler to have a dedicated control. We can mention: Keithley electrometers or some ESRF cards like *OPIOM* or *MUSST*.

The first generic controllers we provide are dealing with:

- Motors Controllers
- Multichannel Analyzers (for fluorescence detectors)
- Temperature Controllers
- 2D detectors (via *Lima*)

Motors

Motor controllers are based on five fundamental classes (`Controller`, `Axis`, `Group`, `Encoder` and `Shutter`). The generic motor controller objects, and derivative devices, provide management of:

- typical basic parameters: velocity, acceleration, limits, steps per unit
- state, motion hooks, encoders reading, backlash, limits, offsets
- typical actions: homing, jog, synchronized movements of groups of motors

The minimal coding part to support a new controller consist, for the developer, in providing implementation of elementary functions like: `read_position()`, `read_veolcity()`, `set_velocity()`, `state()`, `start_one()` and `stop()`.

A **Calculation Controller** is also proposed to build virtual axes on top of real ones.

The list of motor controllers already implemented in BLISS, in use at ESRF, includes (but is not limited to) controllers like Aerotech, FlexDC, Galil, IcePap, Newfocus, PI piezo or Piezomotor PMD206.

Multichannel Analyzer Controllers

The principle is very similar for MCA electronics. An interesting detail of the implementation is the usage or zeroRPC [5], to deport control from a windows computer to the workstation where BLISS is running. This behavior allows to cohere with the **direct hardware control** principle.

First targeted MCA are XIA devices: Xmap, Mercury and FalconX. They will be followed by Maya2000 from OceanOptics and Hamamatsu.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP02

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Simulators

For each type of generic controller, we have built "simulation devices" to test our own code and to provide test devices to help users with the creation of their control sequences.

A simulator like the **mockup motor controller** is used to test the logical part of the motor controller within the frame of a collection of unit tests executed in a continuous-integration process.

SCANNING

BLISS implements a general scanning engine to run all kinds of scans, that emancipates from the dichotomy of step-by-step or continuous scans. Indeed, BLISS introduces the concepts of **acquisition chain**, **acquisition master** and **acquisition device** to be able to perform any kind of scan.

Acquisition Chain

The representation of the acquisition chain is a **tree**. The hierarchical nature of the acquisition chain allows to formalize the dependencies between nodes. There are 2 kinds of nodes:

- master nodes, that trigger data acquisition
- device nodes (leaves), that acquire data

Acquisition chain objects expose 3 methods corresponding to the 3 phases of a scan:

- `prepare()`
- `start()`
- `stop()`

During the preparation phase, the acquisition chain tree is traversed in **reversed level order** (reversed Breadth-first search [6]) in order to prepare the device nodes first, then masters and so on until the tree root ; on each element, `.prepare()` is called. Preparation is decoupled from start in order to make sure minimum latency will happen when starting the scan. Indeed, during preparation each equipment is programmed or configured for the scan. By default, preparation of all equipments is done in parallel.

At start, the same tree traversal procedure is applied as within the preparation phase ; on each chain element, `.start()` is called ; device nodes will begin to wait for a trigger whereas master nodes will start to produce trigger events. It is important to note that devices are **always** started before masters, and that trigger events can be hardware or software. A scan can be seen as an iterative sequence ; after `.start()` method is executed, the acquisition chain enters the first iteration.

It continues until the first master signals acquisition is finished, or in case of error, or if the scan is interrupted. Then, `.stop()` methods are called on each tree element.

Monitoring scan example The following chain describes a scan with one timer master, triggering 3 diode counters (cf. Figure 3).

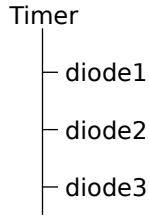


Figure 3: Monitoring scan chain.

Basic scan example This chain describes a scan with one `motor` master, triggering a `timer` master, that triggers in turn 3 diode counters (cf. Figure 4). This is typically the kind of scan *Spec* does with the `ascan` or `dscan` macros, except that in the case of BLISS the step-by-step or continuous nature of the scan does not depend on the acquisition chain, but on the type of master and device nodes.

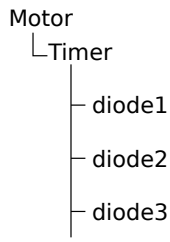


Figure 4: Basic scan example.

Associating basic scan and monitoring This chain describes a scan with 2 top masters: one `motor` and one `timer` (cf. Figure 5). The branch with the motor is like the basic scan above, except that a 2D detector is taking images at predefined motor positions, and for each image it acquires X and Y beam position. On the second branch, there is a simple temperature monitoring. The two top masters run in parallel.

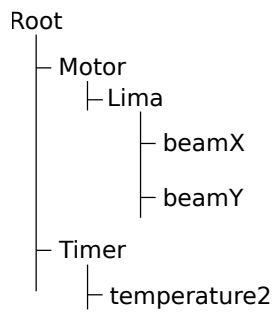


Figure 5: Associating basic scan and monitoring.

DATA MANAGEMENT

BLISS has built-in data management facilities as a first-class citizen. Each node object in the chain has a name, which clearly identifies data sources. Associated with the tree view of the acquisition chain, BLISS creates a data model from the bottom-up that closely follows experiments.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP02

Acquisition Channels

Acquisition chain objects, being masters or devices, define zero or more `AcquisitionChannel` objects which have:

- a name
- a type
- a shape

Acquisition channels describe the kind of data produced by the underlying BLISS control objects.

Data Writing and Publishing

During scans, data is placed in the appropriate acquisition channels; then, the scanning engine temporarily publishes the channels to the *Redis* database, either as plain values for scalars or as references to data files for bigger data. Concurrently, channel data is written by the active data writer object. By default, BLISS saves data in HDF5 format.

Any external process can monitor *Redis* to get notified of on-going acquisitions.and to explore acquired data. This facilitates online data analysis. Scan data is kept in *redis* for a configurable amount of time (set to one day by default).

BLISS provides a Python helper module to iterate over produced data.

BLISS USER INTERFACES

On top of the BLISS library, two user interfaces have been developed in order to provide an entry point for users on beamlines to get access to BLISS functionalities.

Bliss Command Line Interface (CLI)

The `bliss` command line interface is based on *ptpython* [7]. It provides a Python interpreter enhanced with BLISS-specific features. Most notably, the interpreter input loop is replaced to include `gevent` [8] events processing.

`bliss` can load BLISS sessions, via a `-s` command line switch. The command line interface automatically loads session objects, and executes an optional setup script. All globals are exported to the `bliss.setup_globals` namespace, in order to allow users to import session objects in their own scripts.

BLISS Shell Web Application

BLISS ships with an 'experimental' version of a web-based command line interface similar to `bliss` (see above), offering more graphical display possibilities thanks to the web platform (cf. Figure 6).

CONCLUSION

This document presented the context for the launch of the BLISS project, and went through a technical review of all aspects of the development currently conducted at ESRF to renew the beamline experiments control system in the perspective of the EBS [9].

At the moment, BLISS is in an active development phase. Middle term goals include the development of new hardware

Figure 6: BLISS web shell.

controllers, the port of *Spec*-based experiment protocols to BLISS with the collaboration of ESRF scientists, and the improvement of BLISS user interfaces to provide data visualization capabilities using the ESRF *silx* toolkit.

BLISS has already been deployed on Macromolecular Crystallography beamlines, and more ESRF beamlines will benefit from BLISS before the end of the year: Materials Chemistry and Engineering (ID15A), High-Energy Materials Processing (ID31), Materials Science (ID11). BLISS takes up the challenge of deploying a complete new system while the former one is still in production and while beamlines stay in user operation. The BLISS project main objective is to have all ESRF beamlines equiped with BLISS in 2020.

BLISS opens new perspectives in term of beamline experiments control, to bring advanced scanning techniques and enhanced data management to all ESRF beamlines.

REFERENCES

[1] silx, `http://silx.org`

[2] redis, `http://redis.io`

[3] YAML, `http://yaml.org`

[4] TANGO, `http://tango-controls.org`

[5] ZeroRPC, `http://zerorpc.io`

[6] Edward F. Moore,"The shortest path through a maze", in *Proceedings of the International Symposium on the Theory of Switching*. Harvard University Press. pp. 285–292. As cited by Cormen, Leiserson, Rivest, and Stein, (1959).

[7] ptpython, `https://github.com/jonathanslenders/ptpython`

[8] gevent, `http://gevent.org`

[9] J. M. Chaize *et al.*, "The ESRF Extremely Brilliant Source - A 4th Generation light source", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017. `doi:10.18429/JACoW-ICALEPCS2017-FRAPL07`

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP03

A GENERAL SOLUTION FOR COMPLEX VACUUM SYSTEM CONTROLS

G. Bischof, A. Barbour, A. Walter, B. Sobhani
Brookhaven National Laboratory, Upton, New York, USA

Abstract

At the National Synchrotron Light Source II (NSLS-II) there are many different ultra-high vacuum system configurations on the unique beamline end-stations. The proposed controls solution attempts to capture the requirements of all of these configurations with a single standard logic and graphical user interface. Additional design considerations include: resource management for multiple users, providing a high level of abstraction to simplify operation for users, providing a high level of flexibility to do nonstandard operations, minimizing shock from pressure differentials when opening valves, supporting a variety of pumps, and maximizing pump lifetime. At NSLS-II it was determined that all vacuum configurations can be captured by the composition of three standard objects: a "rough vacuum group", "high vacuum group", and a "smart vacuum manifold" which implements a blocking queue. These objects can be flexibly linked together to meet the needs of the beamline experiments. This solution is platform independent, but implemented and tested here using Pfeiffer vacuum Pumps, Allen Bradley PLC, EPICS, and Control System Studio (CSS).

INTRODUCTION

At the National Synchrotron Light Source II (NSLS-II) there are many different ultra-high vacuum system configurations on the unique beamline end-stations. The end-station vacuum systems often differ from the vacuum systems of the rest of the accelerator and beamline in that they frequently need to have sections vented. They also often have unique vacuum requirements for their experiments. End-station vacuum systems often incorporate roughing pumps and turbo pumps to meet their pumping speed and throughput requirements. End-station vacuum systems also incorporate ion pumps and NeG pumps to achieve the best possible vacuum. Some configurations are quite simple, for example: a single vacuum chamber with a vacuum system consisting of a roughing pump, isolation valve, turbo pump, and a gate valve. Other vacuum system configurations are more complex. Figure 1 shows a vacuum system configuration that is typical for a more complex end-station. Vacuum system configurations can vary greatly between end-stations.

There are challenges that come with working with these vacuum systems. Multiple users can be executing independent operations on the vacuum system that require access to the same resource. Currently this is handled by staff coordination. Due to the complexity of these vacuum systems, users are often uncomfortable with executing vacuum system operations. Even expert staff can make mistakes. This project attempts to address these challenges, and simply the operation of end station vacuum systems.

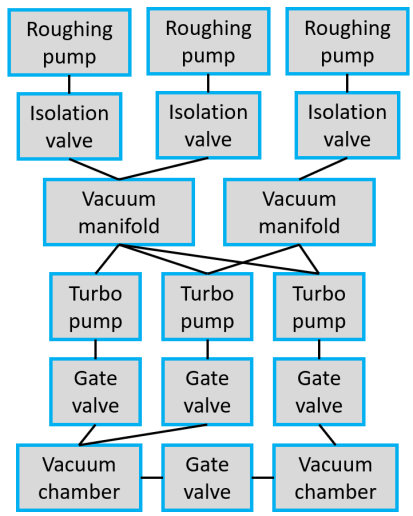


Figure 1: Example vacuum system configuration.

REQUIREMENTS

The goal of this project is to create a modular standard logic and Graphical User Interface (GUI) that supports all of the different end-station vacuum configurations. The design should simplify user operation, by automating common operations, and by providing an abstraction to hide the complexity of the system. The automation of the common operations should reduce the chance for human error and improve response to faults. Users of the system may also need to do a non-standard operation not supported by automated logic, so the solution must also allow for manual control of all output signals. The GUI should clearly and simply communicate vacuum status to its users. The modular standard logic and GUI should make it more efficient to automate new vacuum systems by allowing a copy and paste of standardized modules and linking them together to meet the vacuum requirements.

SYSTEM DESCRIPTION

The end-station vacuum systems at the NSLS-II are composed of a small set of hardware devices: Roughing pumps, vacuum gauges, flow meters, angle valves, gate valves, turbo pumps, ion pumps, and NeG pumps. The signals of the vacuum devices have been wired to distributed PLC I/O modules. The distributed I/O communicates to the PLC via ethernet. At NSLS-II we use EPICS, which is a distributed real-time control system and SCADA, which allows us to integrate many independent control

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP03

systems and devices over ethernet. A distributed control system like EPICS is not required for this solution.

As shown in Figure 2, a unique part of this solution is the logical grouping of the vacuum devices into three standard objects. These objects can be linked together in different arrangements to meet the various vacuum requirements. It was determined that the composition of these standard objects can meet the requirement for all end-station vacuum systems at the NSLS-II. The three standard objects are the "rough vacuum group", "high vacuum group", and "smart vacuum manifold". These objects each have their own function in the PLC logic and their own templated GUI. The functions in the PLC logic, as well as the GUI for each object, can be linked together to represent the entire vacuum system configuration. To provide a complete end-station vacuum solution the vacuum groups can be integrated with additional vacuum components such as valves, ion pump, and NeG pumps.

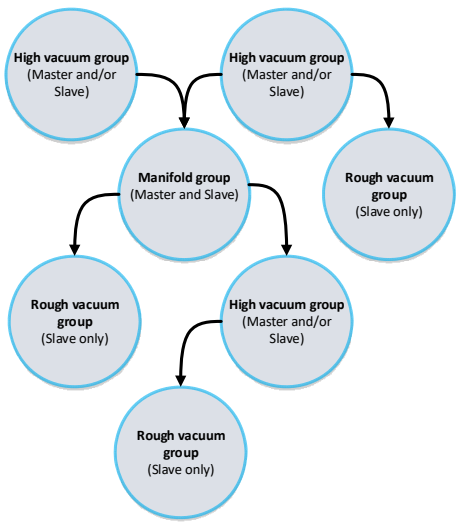


Figure 2: Example of object hierarchy.

VACUUM GROUP OBJECTS

Each vacuum group object consists of PLC logic, an EPICS template file (used when implemented with EPICS), a detailed GUI, and a minimalistic abstracted GUI. This section will describe these features that are common to all vacuum group objects.

PLC Logic

In order to simplify the logic, it does not use a finite state machine. Instead, each output signal is given an independent logic. The functionality of the vacuum group is best explained by its output logic. All vacuum group objects have “Force mode”. If the force mode option is selected, the user can select the value for each output signal manually. This gives the user the ability to do any operation that they want. After disabling force mode all outputs will return to their automated outputs determined the requested vacuum state. Each vacuum group has a set of parameters which can be used to specialize each instance of the object. Critical parts of the PLC logic are presented in this paper.

Detailed GUI

The detailed GUI displays the value of all signals, allows the user to enable and use force mode, and set the value of the vacuum group instance’s parameters (see Figure 3).

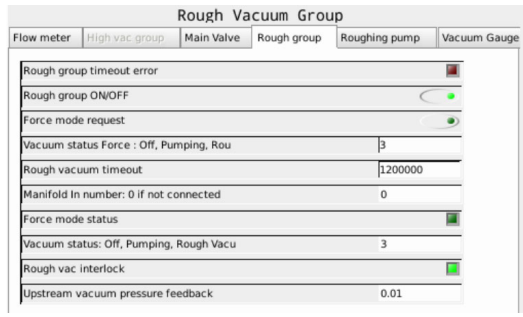


Figure 3: Rough group detailed GUI.

Abstract GUI

The abstract GUI is designed to be minimalistic, intuitive, and hide a reasonable amount of detail about the system. Boolean statuses are displayed as red/green indicators. For example, if the vacuum is better than the vacuum threshold parameter, the vacuum pipe will turn green. The abstract GUIs also display alarm messages if there are any alarms present. The abstract GUI is made with all SVG graphics. The abstract GUIs can be linked together to depict the entire system.

ROUGH VACUUM GROUP

A rough vacuum group (Figure 4) is responsible for supplying rough vacuum to a chamber, the high vacuum group or manifold. The rough group has only two states, “On” and “Off”. It can be turned “On” by pressing the power symbol on the GUI, or by a request from a master (manifold or high vacuum group). Clicking the menu icon will open the detailed GUI.

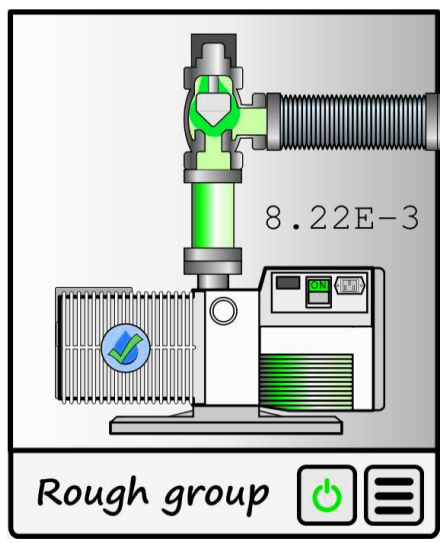


Figure 4: Rough vacuum group abstract GUI.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP03

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Components: Roughing pump, water flow meter/switch, vacuum gauge, isolation valve (main valve).

Input signals: Roughing pump water flow, roughing pump on/off status, roughing pump warning, roughing pump alarm, rough vacuum value, upstream vacuum gauge value, main valve opened feedback, main valve closed feedback.

Output signals: Roughing pump on/off command, main valve command.

Command inputs: Rough group on/off, force mode request.

Status outputs: Force mode status, vacuum status.

Parameters: Flow meter low threshold, flow meter lolo threshold, rough vacuum timeout, manifold input number, rough flow meter option, vacuum gauge high threshold, vacuum gauge hihi threshold.

Hard-stop alarms: Roughing pump water flow fault, main valve feedback error, alarm signal from roughing pump, roughing pump status feedback error.

```
If RoughVac < (0.95 * UpstreamVac) then
    InputLessThanUS := 1;
end_if;
If RoughVac > (1.3 * UpstreamVac) or not Output_RPOnOff then
    InputLessThanUS := 0;
end_if;
```

Code block 1: Rough group intermediate logic

```
If CmdRoughOn and Interlock_RoughOK and not HardStopAlarm then
    Output_RPOnOff := 1;
end_if;

if (Input_MainVlvClosed or Alarm_MainVlvFB)
and (not CmdRoughOn or not Interlock_RoughOK or HardStopAlarm) then
    Output_RPOnOff := 0;
end_if;

if CmdRoughOn and Interlock_RoughOK and not HardStopAlarm
and Input_RPOnSts and (InputLessThanUS or RoughVacGood) then
    Output_MainVlvCmd := 1;
end_if;

if not CmdRoughOn or not Interlock_RoughOK or HardStopAlarm
or not Input_RPOnSts then
    Output_MainVlvCmd := 0;
end_if;
```

Code block 2: Rough group output logic

HIGH VACUUM GROUP

The high vacuum group consists of a turbo pump, roughing line vacuum gauge, high vacuum gauge, and an optional gate valve (GV) on the inlet side of the turbo (Figure 5). The high vacuum group can be used to generate high vacuum in a chamber, or in a manifold. The logic for the high vacuum was developed to not require a vacuum gauge between the GV and turbo inlet. The high vacuum group can function as a master, by making requests from a smart manifold, or a rough group. It can also function a slave and respond to requests from a manifold. It can also be setup to do both at the same time.

The high vacuum group has several states that can be requested, “Play”, “Pause”, “Stop”, and “Vent”. These automated states should cover the most of the operations required by the users. In the “Play” state, the turbo is running, and GV is open. In the “Pause” state, the turbo is running and the GV is closed. In the “Stop” state, the GV is closed, the turbo is stopped, and the turbo section is vented if the vent option is enabled. In the “Vent” state, the GV is open, the turbo is stopped, the chamber and turbo section are vented. The “Vent” state can be requested through the detailed GUI.

The most critical section of logic for all of the vacuum groups is the logic of the GV command. Opening this valve at the wrong time can damage the turbo, or components in the chamber. The parameters, “MaxDifferential” and “TurboMaxPressure” should be selected carefully. “MaxDifferential” is the maximum pressure differential across the GV where it is still safe to open the GV. A smaller value for this parameter will reduce shock on components in the chamber. “TurboMaxPressure” is the maximum chamber pressure where it is safe to open the GV on a full speed turbo. It is recommended to contact your turbo manufacturer to get this information.

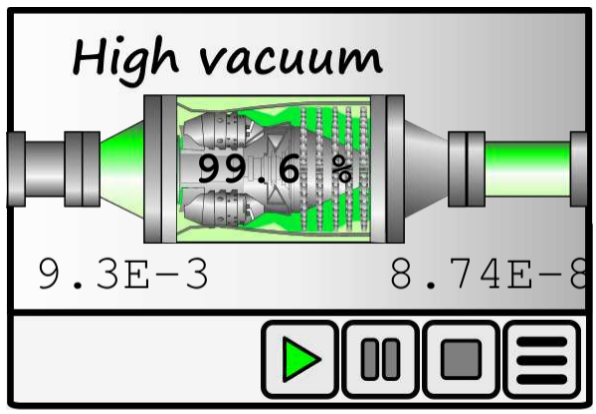


Figure 5: High vacuum group abstract GUI.

Components: Turbo pump, roughing line vacuum gauge, high vacuum gauge, gate valve (optional).

Input signals: Turbo error, turbo speed, gate valve position feedback, fore-line vacuum pressure, chamber vacuum pressure.

Output signals: Turbo auto vent, turbo on/off, gate valve command.

Command inputs: “Play”, “Pause”, “Stop”, “Vent”, force mode request.

Status outputs: Force mode status, rough vacuum status, high vacuum status, current vacuum state.

Parameters: Gate valve option, turbo vent option, roughing-line vacuum pressure status thresholds, chamber vacuum pressure status thresholds, high vacuum timeout, rough vacuum timeout, smart manifold output port number, MaxDifferential, TurboMaxPressure.

Hard-stop alarms: Turbo pump alarm, GV feedback error.

```
If RoughVacuum < (0.9 * HighVacuum) then
    InputLessThanUS := 1;
end_if;
If RoughVacuum > (1.1 * HighVacuum) then
    InputLessThanUS := 0;
end_if;
```

Code block 3: High vacuum group intermediate logic

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP03

```
// Cmd_VacStateRequest possible states
// 1 = Stop
// 5 = Play
// 6 = Pause
// 9 = Vent
if (Cmd_VacStateRequest = 5) or (Cmd_VacStateRequest = 9) and Interlock_HighVacOK
and (InputLessThanUS or (ChamberVacLessThanThreshold and TurboSpeedGreaterThan80)
and not GreaterThanMaxDifferential and not HardStopAlarm then
    Output_MainVlvCmd := 1;
end_if;

if (((Cmd_VacStateRequest = 1) or (Cmd_VacStateRequest = 6))
or not Interlock_RoughOK or not Interlock_HighVacOK or HardStopAlarm)
and Interlock_MainVlvClose then
    Output_MainVlvCmd := 0;
end_if;

Output_TurboOnOff := Interlock_RoughOK and Interlock_HighVacOK and Sts_RoughTCG
and ((Cmd_VacStateRequest = 5) or (Cmd_VacStateRequest = 6));

Output_TurboVentEnable := (Cmd_VacStateRequest = 9) or (not Output_MainVlvCmd
and Input_MainVlvClosed and (Cmd_VacStateRequest = 1) and Param_VentOption);
```

Code block 4: High vacuum group output logic

SMART VACUUM MANIFOLD

The smart vacuum manifold is used to share rough vacuum with multiple high vacuum groups or manifolds. The manifold can also be configured to also to be backed by a high vacuum group or another smart manifold group. The output side of the manifold is defined as the side that goes to the lowest pressure vacuum. The other side is called the input side, this side in most cases is connected to the rough groups. The smart manifold consists of an array of input valves, and an array of output valves as shown in Figure 6.

The smart manifold acts as both a master and slave. High vacuum groups can make request to manifold for rough vacuum. The manifold can make requests to rough vacuum groups or other high vacuum groups.

Requests to the manifold can be "blocking" requests, or "non-blocking requests". A blocking request means that all other output valves other than the output that is receiving the request must be closed. If multiple blocking requests are issued at the same time the requests will be added to a FIFO queue in the order that they were received. If there are no blocking requests, then all non-blocking requests can be fulfilled at the same time. This is useful for allowing multiple users to make requests at the same time.

Output valves of the manifold can be configured as "soft-pump" valves. Soft pump valves have two different size openings. The smaller opening size can be adjusted. In situations where is desirable to reduce stresses on chamber components, a soft-pump valve is useful. First the small opening of the valve will open, once the output pressure the below the set threshold the valve will fully open.

Due to the array nature of the smart manifold object it will be simpler to describe it in words and not code.

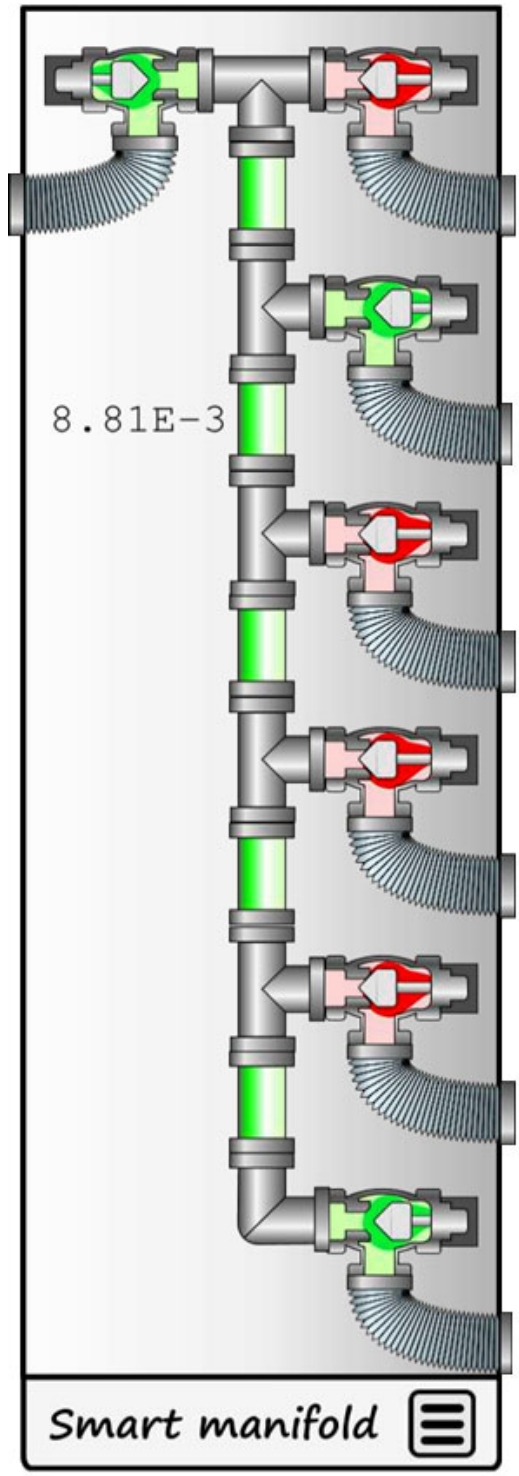


Figure 6: High vacuum group abstract GUI.

Components: Array of output angle valves, array of input angle valves, manifold vacuum gauge, array of output vacuum gauges.

Input signals: Array of output angle valves position feedbacks, array of input angle valves position feedbacks, manifold vacuum pressure, array of output vacuum pressures.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP03

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Output signals: Array of input angle valve commands, array of output valve commands.
Command inputs: Force mode request, vacuum state request (array of integers), master block request (32 bit integer).
Status outputs: Force mode status, manifold vacuum status, input array vacuum status, output array vacuum status, blocking status array.
Parameters: Input valve count, output valve count, output valve soft-pump array mask, input array vacuum pressure status thresholds, output array vacuum pressure status thresholds, blocking request timeout.
Hard-stop alarms: None.
Blocking queue: The smart manifold object has an input called "master block request", which is a Boolean array. Each object that will be making requests for vacuum from the smart manifold can set the bit of the index which matches the output port that it is connected to. This will initiate a blocking request by pushing the index to a FIFO queue. If the item at the front of the queue stops blocking, by clearing its block request, it will be popped. If an item blocks for a time greater than the parameter "blocking request timeout" it will be popped and pushed to the end of the queue. If there are any blocking requests present, only one output valve can be open at a time.
Output logic if there is an item in the queue: Only the output valve for the master at the front of the queue can be opened. To open this valve the closed feedback of all output valves must be observed, and the interlock satisfied. The slow pump output valve will open if the master requesting the "Play" or "Pause" state and the slow pump option is enabled for the port. The corresponding angle valve will fully open when the pressure on the master side becomes better than the output array vacuum pressure status threshold parameter. The output valve will close if the master is no longer requesting the "Play" or "Pause" state, or if the queue has moved to the next item.
Output logic if queue is empty: If the interlock for the output valve is satisfied and the slow pump option is enabled for the port, then the slow pump output valve will open**,** if the master on the same port is requesting the "Play" or "Pause" state. The corresponding angle valve will fully open when the pressure on the master side becomes better than the output array vacuum pressure status threshold. The output valve will close if the master is not requesting the "Play" or "Pause" state.

CONCLUSION

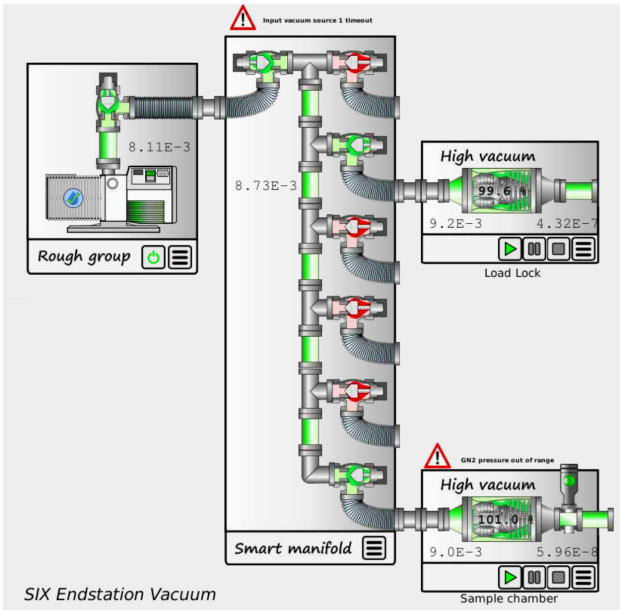


Figure 7: SIX end-station vacuum GUI.

This solution was deployed and tested at the SIX beamline and is currently being used for their end-station. It currently meets the goals defined in the requirements section. Figure 7 shows their current GUI, depicting their current vacuum system configuration. Some polishing is still required, and further development and testing to improve the robustness is planned. The CSX, IOS, and ESM beamlines were also considered and have the existing hardware to roll out the new vacuum control system. This new system will provide unprecedented protection and automation of these beamline vacuum end-stations.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP04

DEVELOPING AND VALIDATING OPC-UA BASED INDUSTRIAL CONTROLS FOR POWER SUPPLIES AT CERN

Michael Ludwig, Marc Bengulescu, Ben Farnham, Jonas Arroyo Garcia, Pablo Gonzalez Jimenez, Fernando Varela, CERN, Geneva, Switzerland

Abstract

The industrial control systems of CERN's experiments are undergoing major renovation since 2017 and well into CERN's second Long Shutdown (LS2) until the end of 2019. Each detector power-supply control system runs several hundred software instances consisting of many different components in parallel on a large scale, broadly distinguishable as servers and clients. Our accumulated experience during LHC runs proves that some complex control issues are impossible to detect using stand-alone components on a small scale only. Furthermore, new components must be developed well before the electronics becomes available, without impact on operations. Moreover, during LS2, the improved and now widely established Open Protocol Communication Unified Architecture (OPC-UA) replaces OPC-DA as middleware protocol. For these reasons, we developed a simulation environment to emulate the real, and valuable, CAEN power-supply electronics underneath the OPC-UA servers. This distributed simulation is configurable to mimic and exceed the nominal conditions during production and provides a repeatable setup for validation. This paper discusses the functionality and use of this simulation service.

WHY A SIMULATION?

The control system for the CERN experiments can be segmented into three layers. Firstly, the electronics layer for power supplies and many other components, which includes commercially available modules connected to the detector sub-parts. Some CERN experiments need many thousands of power channels laid out in “electronic trees” spanning a wide variety of characteristics and behaviour, generally using networked modules from different commercial vendors running black box firmware.

The second layer provides the OPC-UA [1,2] interfaces to the different kinds of power supply modules. OPC-UA provides standardized abstraction and communication, and replaces its now obsolete predecessor OPC-DA. Several tens of server instances connect to the power channels for a typical experiment, each server with its specific electronic tree.

Lastly, the client layer provides classical supervisory control and data acquisition (SCADA) services including logging, error handling and analysis, history data storage and functional abstractions like finite state machines and interfaces for specific detector functionality. The two uppermost layers rely entirely on the accurate and robust translation of the physical electronic trees into the software representation with the intended functionality. A commercial object-orientated SCADA system, WinCC-OA [3], is used to realize much of the upmost layer. This now mature control system has to undergo updates and functional fixes, which all have impact on mission criticality in some way, but where their deployment should not require any dedicated production time.

The investigation of issues and the actual deployment of updates requires very careful coordination and is, generally, a delicate process for the two following reasons. Firstly, still affordable laboratory tests can never approach the same complexity, scale and diversity as a real running system with inputs and errors. Any new versions of control components, concerning also new versions of the electronics, need testing and validation through all control layers. Problems related to scaling, which were reproducible in the experiments in the past but not in laboratory tests, and could not be confirmed by the vendor either, need to be investigated on appropriate scale and complexity, to then be followed up adequately.

Secondly, a system with delicate high voltage detector-hardware must not be used for testing critical and safety related software. Even if one would take the risk, it would be very hard to identify and re-produce any complex failure scenario for further debugging. A satisfying validation for a working control system of that scale could be obtained by performing global centralized “dry runs” using real electronics, but disconnected high voltage output stages, yet such runs are deemed impractical and too expensive. For these reasons a scalable low-level platform which provides **ven**dor **u**nified **s**imulation (VENUS) for power supplies together with a realistic behaviour is needed as an efficient and flexible solution which can also support training and development prototypes.

REQUIREMENTS

Scale and Validation

Both issues, scale and validation, can be efficiently addressed with a simulation environment which emulates the electronics of the first layer. Large scale “developer tests” are possible, being much more independent on whether the electronics is available, in production, or even only a planned purchase. Replaying complex and demanding scenarios should push various controls software components in the upper layer to their respective limits, thus providing useful stress tests. If scale and complexity of the production systems can be significantly exceeded in the simulation, we assume with good confidence that our systems are adequately validated.

Performance and Complexity

Changing parameters in the electronic tree create events with associated data payload, which are pushed upwards to the subscribing server. Event rates of several kHz are

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP04

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

common for real electronic trees of some hundred channels. Saturation typically occurs from about 10kHz onwards, then some buffering and filtering strategies are applied from the server on and upwards. The simulation should be able to deliver event rates significantly exceeding the real electronics. It should also potentially scale better than the most complex real electronics tree and it should be configurable and manageable in compatible ways. Evidently, it must integrate seamlessly to emulate the behaviour of the first layer, and it should be capable of producing and repeating scenarios beyond nominal conditions to prompt associated error behaviour.

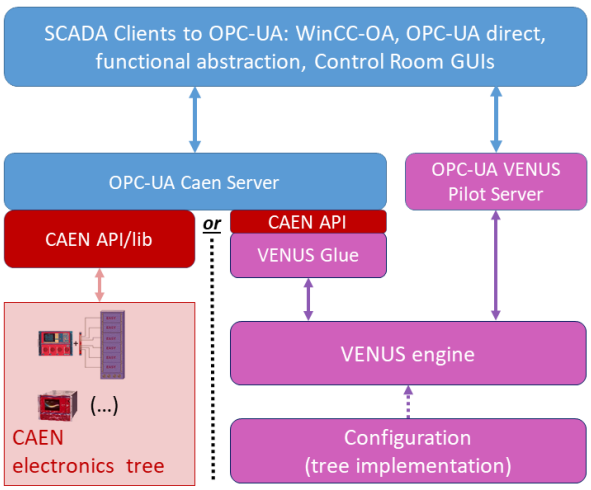


Figure 1: VENUS architecture in the case of CAEN. Colors represent types of communication: vendor specific (red), Ømq-protocol (violet) or OPC-UA (blue).

Furthermore, the simulation should replace power supplies of all major commercial systems used at CERN [4], in terms of software. All of the following discussion refers to the CAEN power supply systems, since they are a widely used system at CERN, provide deep functional complexity including radiation hard sub-trees and feature specific powering options. Simulation of the remaining vendors follows the same principles adapted for their systems.

ARCHITECTURE

The VENUS architecture is shown in Fig. 1. The SCADA layer interfaces to OPC-UA CAEN servers, which communicate with either the real electronics through the CAEN API, or through the replacement API for simulation. A configured VENUS Engine provides the same underlying behaviour by software and is managed by a separate OPC-UA Pilot Server. The simulation replaces the first control layer entirely, where the API for each type of real power supply defines the behaviour of the simulation in each case. The VENUS glue establishes isolated communication to the VENUS engine for the specific CAEN API. Each instance of a VENUS engine is configured to reflect the specific electronics tree to be played.

Configuration

The configuration defines the instances of each type of module including any electronic subtrees for the radiation hard versions. The firmware of each module determines its real API, which evolves over many years and must be reproduced in all cases. Configurations can be discovered from existing electronic trees, designed to produce large scale stress test scenarios with subtrees and mixed module types, define consistency tests and also model yet-to-be-purchased electronics.

Pilot Interface

Further functionality beyond the scope of the vendor API is needed to provide accurate power supply behaviour. A dedicated OPC-UA pilot server communicates directly to the VENUS engine in order to perform:

- channel load management; a load model needed to draw currents,
- ramp switching; an external signal switching between two parameter sets for pre-ramping and ramping,
- current-limit tripping; an excess current switches the channel off in a well-defined manner,
- artificial noise; in order to mimic randomly occurring behaviour like current spikes and degrading loads,
- bulk-channel operations; like tripping selected channels by decreasing their loads to draw currents exceeding their nominal limits,
- event clock tuning to produce higher or lower event rates for use in stress test,
- status and health monitoring of the engine.

The specific functionality in the SCADA layer relating to the pilot is kept carefully isolated from all other clients.

DESIGN AND INTEGRATION

Building Standards

The building, integration and execution of VENUS as a scaling service is shown in Fig. 2. A framework for rapid OPC-UA server development, quasar [5], is used to generate server skeleton code for a configured quasar project. This quasar project configuration defines the layout and characteristics of the OPC-UA address space; interface to the SCADA clients. Quasar supports two toolkits, Unified Automation (commercial) [6] and Open62541 (open source) [7], and guarantees compliance to OPC-UA. The artefacts, VENUS engine and OPC-UA servers, are built from specific source code together with the generated protocol buffer message formats [8] and the quasar generated codes. All artefacts are continuously built using Jenkins-CI to guarantee integrity.

API Serialisation

Each function call of the vendor API, which is implemented in plain C for CAEN, is independent of every other call to a large extent and no state information is kept in the API. Therefore communication between the CAEN API and the object-orientated engine can be realized in transactional pairs of request-reply messages in the VE-

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP04

NUS glue. Each message pair maps to a specific API function call and its results. Message pairs need to be executed as transactions to protect against re-entrant calls and parallel requests stemming from multi-threaded servers. The initial calls for hardware discovery are issued serially from the server and the simulation engine replies back with the corresponding part of its electronic tree.

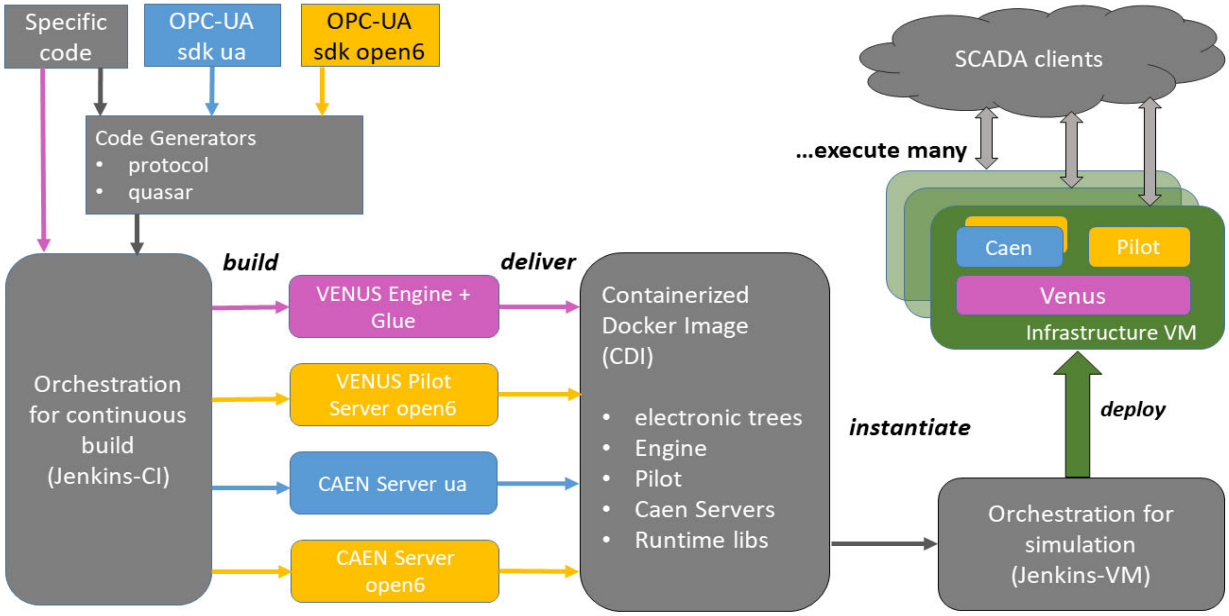


Figure 2: Building, integration and deployment as a service in a cluster of virtual machines. The workflow stages source, build, deliver, instantiate, deploy and execute are shown. Colour and shape coding: standards (pink, yellow, blue), tools (grey, round corners), infrastructure service (green), source codes (sharp corners), complex clients (cloud).

Deployment and Scalability

A set of all servers and the engine, together with their runtime dependencies and previously validated configurations for the electronic trees is delivered into a containerized docker image (CDI) [9] as shown in Figure 2. Both OPC-UA sdk-flavours for the servers may be used and compared, but the open6-sdk is preferred for the pilot due to licensing restrictions. The CDI can also be dissociated into separate units which communicate over network for further distributed stress testing. The orchestration for simulation, which is optimized for continuously running services, selects one of the valid configurations for deployment and runs it in a virtual machine (VM). Many VMs with identical or different configurations are used to realize simple or complex scenarios for the SCADA client layer. Up to one hundred VMs can be managed using the available resources, where each VM offers some ten to several thousand channels.

CONCLUSION AND OUTLOOK

The need for a fast, flexible and highly scalable simulation service arises from the ongoing control system renovation at CERN for LS2. As long as the vendor API in question is easily serializable, a clean solution based on distributed networking with protected message pairs can be naturally found. A seamless integration, without any impact on the very complex SCADA layer, is provided by replacing the vendor's implementation of the API and adding a pilot interface for further management. Setups can be run all-in-one on single computers or widely distributed. Since the VENUS implementation uses C and C++ it can cover event rates well above and below the characteristics of the real electronics. The construction of challenging stress tests becomes thus possible, leading to a systematic and repeated validation of a large, complex and mature system, as used in the experiments. Further work is ongoing to extend VENUS for the other vendors.

REFERENCES

[1] W. Mahnke, S.-H. Leitner and M. Damm, *OPC Unified Architecture*. Springer, Berlin Heidelberg, Germany, 2009.

[2] The OPC Foundation, "OPC Unified Architecture", http://opcfoundation.org/opc-ua/

[3] WinCC-OA, http://www.etm.at

[4] CAEN, http://www.caen.it;
ISEG, http://www.iseg-hv.com;
WIENER, http://www.wiener-d.com

[5] P. P. Nikiel, B. Farnham, S. Schlenker, C.-V. Soare, V. Filimonov and D. Abalo Miron, "Quasar – A generic framework for rapid development of OPC UA servers", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, paper WEB3O02.

[6] Unified Automation GmbH, "C++ based UA Server SDK".

[7] Open62541, http://open62541.org

[8] Google Protocol Buffers, https://developers.google.com/protocol-buffers/

[9] Docker container technology, http://www.docker.com

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP05

WIRELESS INTERNET OF THINGS (IoT) APPLICATION IN THE TLS

C. H. Kuo, C. J. Wang, T. Y. Lee, W. U. Lin, Y. K. Lin, S. Fann, H. H. Chen, B. Y. Chen,
H. C. Chen. S. J. Huang, C. S. Huang, J. A. Li
NSRRC, Hsinchu 30076, Taiwan

Abstract

The internet of thing is applied in the accelerator more frequently than before. There are many advantages in data acquisition and control-oriented applications, for example, easy to distribute remotely and less cables needed, low noise generated, many commercial interfaces for choosing. The stable wireless communication is also applied in the measurement system. The high reliability and security of wireless communication with server and client structure is introduced. The structure design and implementation of IoT are summarized in this report.

INTRODUCTION

The applications of IOT are based on Raspberry Pi 3 and Intel Celeron N3050 system in the operation group.

The Raspberry Pi 3 is applied in the temperature measurement with angalog/digital interface. The Raspberry Pi3 system is show in Figure 1.

Hardware specifications are listed:

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display

Figure 1: Raspberry Pi 3 system.

The Intel N3050 system is apllied in the various displays these include of alarm, security monitor system. The systme is shown in Figure 2. Intel Celeron N3050 Kit is an entry-level, small-footprint. Equipped with a dual-core Celeron® processor, this Intel NUC includes new features. These features include items such as a VGA port, SDXC card slot and TOSLINK audio output. It also continues to offer 4x USB 3.0 ports, 4K video support, and 802.11ac Wi-Fi. The hardware specifiaction are lists:

(a) Model A (b) Model B

Figure 2: Intel Celeron N3050 system. There are two difference form factors are applied. Model A is suppoted in 4 K display and IR remote control, model B is applied in the weak wireless signals area with high gain antenna.

- Soldered-down dual-core Intel® Celeron® processor N3050 with up to 6 W TDP
- 4K HDMI display
- USB 3.0/SATA6.0
- IR Receiver Sensor
- Intel® Wireless-AC 3165 + Bluetooth 4.2
- 10/100/1000 ethernet

This system are applied in the 4K display. Alarm and security monitor are adapted in this application.

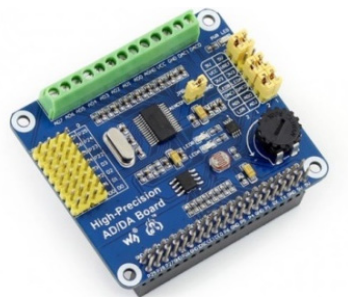


Figure 3: High-precision AD/DA convert.

Wireless lan and bluethooth interface are built in this low cost production that can provide amout of arrangement with simple DC power and manpower. In most time, the doors of storage ring tunnel are close. This is useful in the temporary measurement and system diagnostic for this short period time.

INTERFACE OF IoT

Analog Interface

Since there are no AD/DA functions on Raspberry Pi GPIO interface, this may restrict the application development on Raspberry Pi sometimes. Hence, we have chosen high-precision AD/DA board from Waveshare

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP05

https [1] to compensate for this drawback. The features of this AD/DA board includes on-board ADS1256 (from Texas Instrument) of 8 channel 24 bit high-precision ADC (4 channel for differential inputs) with up to 30K samples per second data rate, on-board DAC8532(also from Texas Instrument) of 2 channel 16bit high-precision DAC, and 4 pins dedicated for digital I/O which are individually configurable as either inputs or outputs by the GPIO control register. The usage of this expansion board enables one to implement those data acquisition and external device control tasks on Raspberry Pi for applications of Internet of Things (IoT) field. This hardware board is shown in Figure 3.

Figure 4: Graphic user interface of AD/DA board in the control console.

The communication between the expansion board and Raspberry Pi is handled over an SPI-compatible serial interface that can be operated with a 2-wire connection. For the Raspberry Pi (SPI master), we have installed the Raspbian Jessie 4.9.24-v7+ operating system and all applications of software are built on it. EPICS (Experimental Physics and Industrial Control System) 3.15.5 release environment has been downloaded and established on the Raspberry Pi [2-3]. Consequently, the channel access technique is implemented on the expansion board and Raspberry Pi system. The C++ based asynPortDriver [4-5] device support is utilized to obtain duplex data interchanged between SPI master and slave. In EPICS database, waveform records are used for octet commands and responding data between both sides. For the purpose of initializing required commands sent to ADS1256, array subroutine records of EPICS are utilized. The sequence records of EPICS have been constructed for ADC (single ended and differential inputs), DAC and GPIO control logic sequences. The graphic user control interface of AD/DA board is shown in the Figure 4.

Digital Interface

Another application focuses on one-wire temperature and humidity sensors which communicate with the raspberry Pi using a single GPIO pin. The AM2302 and DHT22 sensors from Aosong [6] is utilized to implement the application. The DHT22 is shown in the Figure 5. It is digital connection with SPI interface. Because of better control over size and precise timing of sensors, C/C++ programs are compiled into *.so dynamic library. A software input output controller of EPICS is used for storing sensor readings whose values are updated by pyEpics (channel access for Python) [7] in the database. The method to link Python with C/C++ is to use a wrapper called Boost.Python [8] that binds C/C++ and Python. It simplifies the coding and provides methodology for calling to C++ objects in Python. The C++ namespace and Boost.Python specially defined macro are used for calling C/C++ functions within the previously mentioned *.so dynamic library [9].

Since EPICS channel access mechanism is implemented on the Raspberry Pi system, we have selected and utilized the CSS (control system studio) [10] channel access client software to build the graphic user interface for our applications. Please refer to Figures 6, which are obtained from acquired screenshots.

Figure 5: Temperature sensor DHT22 with digital interface.

In this context, we would like to investigate the possibility of constructing a corporate WiFi interconnected IoT network by CSS graphic user interface, database archiving tool, EPICS system, and Java EE server. This is only the initial stage, and a lot of efforts must be dedicated to in the future. We have expected more obstacles lying ahead of us, and we need to overcome those persistently.

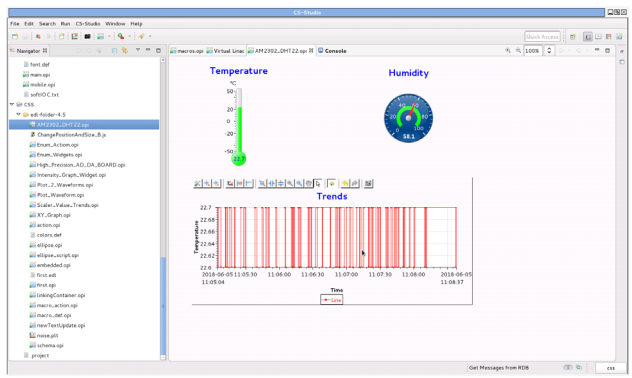


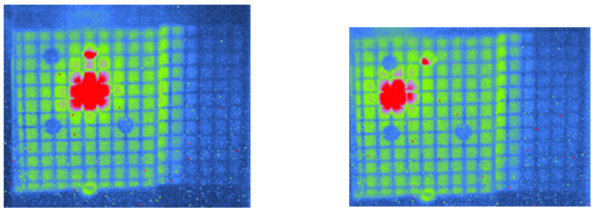
Figure 6: Graphic user interface of DHT22 temperature sensor.

The temperature isn't sense in the tunnel of storage ring. In the most case, the temperature sensor is far away from instrument. In Figure 7, one of the transport line quadrupole magnet cooling pipes is fail, the beam size is divergence. It is hard to injection from transport line to storage ring, but air temperature of tunnel is just 0.1°C difference. DHT22 supports high resolution and precision measurement. Low cost and less calibration with digital interface keep from cabling problem.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP05

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.



(a) Cooling water is fail (b) Cooling is recovery

Figure 7: Beam profile before injection in end of the transport line. (a) Cooling effect is gradually failed from water pipe congestion. (b) External fan to replace cooling water.

OTHER APPLICATIONS

The following applications are used in the operation and system diagnostics:

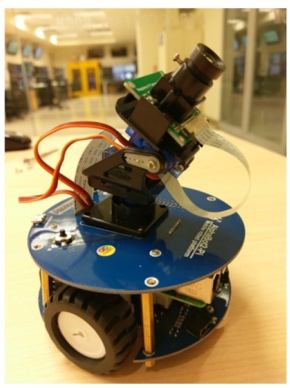


Figure 8: AlphaBot2 is tested in the TPS control room.

AlphaBot2

AlphaBot2 includes of multi interface in Raspberry PI3. It features rich common robot functions including line tracking, Bluetooth/infrared/WiFi remote control.

- 5-ch infrared sensor, analog output, combined with PID algorithm, stable line tracking
- Onboard modules like line tracking, obstacle avoiding, needs no messy wiring
- TB6612FNG dual H-bridge motor driver, compared with L298P, it's more efficient, more compact, and less heating
- N20 micro gear motor, with metal gears, low noise, high accuracy
- LM2596 voltage regulator, provides the Pi with stable 5V power
- TLC1543 AD acquisition chip, allows the Pi to use analog sensors
- PCA9685 servo controller, make it more smoothly to rotate the pan head
- CP2102 UART converter, easy for controlling the Pi via UART

The software programming is still going for auto pilot application. This system is shown in Figure 8.

Alarm and Security System Display

Alarm system of operation is based on the BEAST [11]. The css software with Intel N3050 platform is applied in the TV display. The alarm GUI is shown in Figure 9.

CONCLUSION

Wireless IoT is tested and applied in the various applications with low cost platform and easy distribution. The system reliability is still struggling with wireless interface. In the control room, the application is very stable. In the tunnel of storage ring, this wireless system isn't reliable. It still takes some efforts to improve in the future, such as radiation shielding, schedule system reboot, and so on.

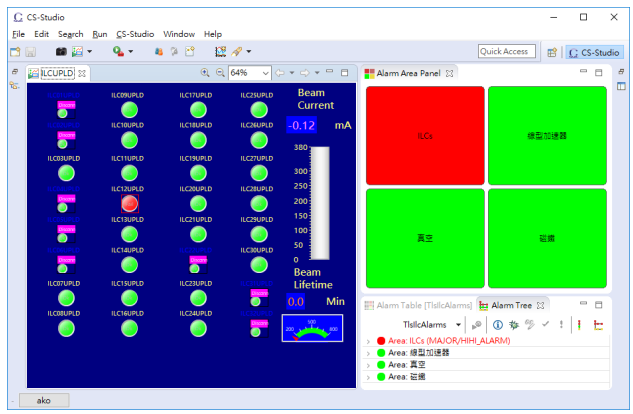


Figure 9: CSS alarm display with IoT.

ACKNOWLEDGEMENTS

Thank for Y. S. Cheng and Y. T. Chang of instrument and control group. They provide for network and database supporting. From these the resources, we can establish wireless control interface and data achieving for system analysis.

REFERENCES

[1] https://www.waveshare.com/wiki/High-Precision_AD/DA_Board

[2] http://prjemian.github.io/epicspi/

[3] https://epics.anl.gov/

[4] https://epics.anl.gov/modules/soft/asyn/R4-12/asynPortDriver.html

[5] https://github.com/ffeldbauer

[6] http://www.aosong.com/

[7] http://cars9.uchicago.edu/software/python/verview.html

[8] https://www.boost.org/doc/libs/1_66_0/libs/python/doc/html/index.html

[9] Molloy Derek, *Exploring Raspberry Pi Interfacing to the Real World with Embeded Linux*, John Wiley & Sons Inc., pp. 214-215.

[10] https://github.com/ControlSystemStudio

[11] https://github.com/ControlSystemStudio/cs-studio/wiki/BEAST

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP06

DATA ARCHIVING AND VISUALIZATION OF IRFEL*

Y. Song, X. Chen, C. Li, K. Xuan, J. Wang, G. Liu†, NSRL, USTC, Hefei, Anhui 230029, China

Abstract

An Infrared Free Electron Laser Light (IRFEL) is being constructed at National Synchrotron Radiation Laboratory. The EPICS Archiver Appliance provides the functions of historical data acquisition, archiving, migration, retrieval and management in the IRFEL facility. A Single-Page Web Application is developed for the data visualization based on Vue.js framework and Highcharts JavaScirpt library. A unified interface is developed for the visualization to integrate multiple data sources and provide the same retrieval entry of the historical data from EPICS Archiver Appliance, the real-time data from EPICS IOC, the statistical data from database and the alarm information from the Phoebus. This paper will describe the implementation details of data archiving and visualization of IRFEL.

INTRODUCTION

Tunable Infrared Laser for Fundamental of Energy Chemistry (FELiChEM) is the significant scientific instrument, which is supported by the National Natural Science Foundation of China in 2013. Infrared Free Electron Lasers (IRFEL) is the core part of FELiChEM, which can accelerate beam to 60MeV and generate middle-infrared and far-infrared laser [1]. The control system of IRFEL is developed based on Experimental Physics and Industrial Control System (EPICS) [2].

The operation of a particle accelerator complex is a long-term experiment. It is essential to record Process Variables (PVs) for further data analysis. With the continuous development of EPICS, a series of data archiving systems has been released in the EPICS community, such as, Channel Archiver, RDB Channel Archiver and the EPICS Archiver Appliance [3]. The EPICS Archiver Appliance (AA) is used as the data archiving tool for the IRFEL. A Web-based GUI is developed for the data visualization. It not only provides the function of historical data query, but also integrates real-time data, statistical data and alarm information into a web application. This paper will describe the implementation details of data archiving and visualization of IRFEL.

SYSTEM ARCHITECTURE

The EPICS Archiver Appliance (AA) is a data archiving tool released in the EPICS community and provides the functions of data acquisition, storage, migration, retrieval and management in the IRFEL facility [4]. AA divides the data into three types: short term store (in RAM disk), medium term store (in local SSD disk) and long term store (in NFS).This mechanism makes AA have high data retrieval performance, which is helpful to machine status analysis and fault diagnosis.

Figure 1 shows the overall structure of the whole system. It is a typical web application and implements the complete separation of the front and back end. The data displayed on the front end can be divided into four types: real-time status, historical data, statistical data and alarm information. As a reverse proxy server, Nginx provides a unified interface for querying these four types of data. The initial sources of the data are all the IOCs in the control system, but they are processed through different ways. The implementation will be described in detail about how to archive, store, process and visualize the data in the following sections.

IMPLEMENTATION OF SERVER SIDE

Real-time Data

It is necessary to display the real-time state of accelerator (such as beam current, running state, etc.) on the web page. However, data archiving tools generally have some time delay to write data to the database and real-time data can not be obtained from archiving tools. In order to solve this problem, the real-time state push is implemented by WebSocket communication from IOC to web page directly.

The WebSocket protocol provides a way of creating web applications that support real-time bidirectional communication between clients and servers. Figure 2 shows the principle of the status push program. Its core is the *Websocketd*, which is a small command-line tool that will wrap an existing command-line interface program, and allow it to be accessed via a WebSocket. As long as you can write an executable program that reads STDIN and writes to STDOUT, you can build a WebSocket server in any programming language [5].

A python script is developed to monitor the PV change in IOC Record via Channel Access and writes them into STDOUT. Any text printed by the process to STDOUT shall be sent as a WebSocket message whenever a newline is encountered. In this way, the JavaScript script running in the browser can get the latest change values of the PVs through the WebSocket and display them on the web page.

Historical Data

The EPICS Archiver Appliance comes with a web interface that has support for various business processes. The web interface communicates with the server principally using HTTP/JSON. There is a rich catalog of business logic that lets the user add/modify/delete PVs from the archiver. This design provides convenience for its integration with other systems. Therefore, client application can get historical data directly from this interface.

* Work supported by National Natural Science Foundation of China (No.11375186, No.21327901)
† Corresponding author, gfliu@ustc.edu.cn

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP06

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

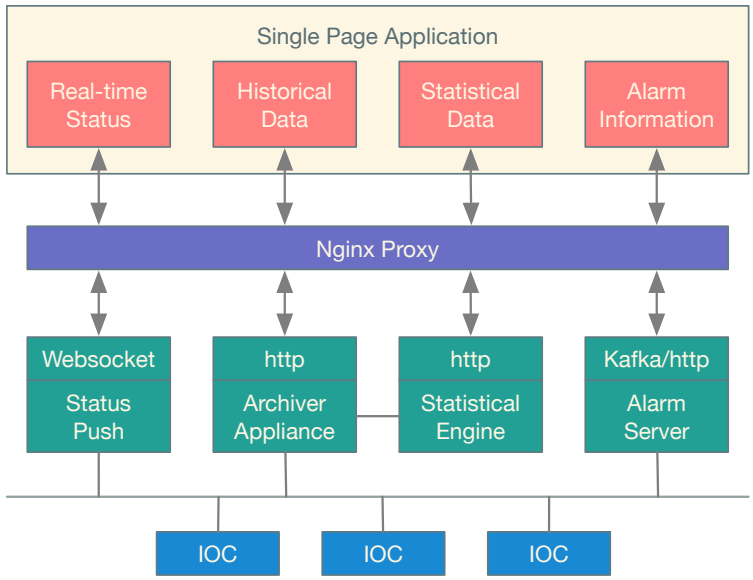


Figure 1: Architecture of the whole system.

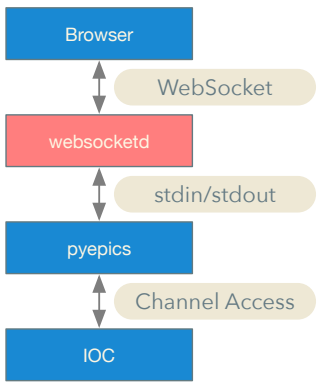


Figure 2: Principle of the status push program.

Statistical Data

The statistical data are obtained after processing historical data in AA. It mainly includes operation mode statistics and integral current query. A daemon is developed using Python based on Flask to get data from AA, perform calculations within itself, and provide HTTP interfaces for client side.

Alarm Information

The Alarm system of IRFEL is build based on Phoebus. Compared with BEAST using Java Message Server(JMS), Phoebus uses Kafka as the communication method between alarm server and alarm client [6]. Similar to the way of processing Statistical Data, a daemon is also developed based on Flask to get alarm message from Kafka using kafka-python library and provide HTTP interfaces for client side query.

IMPLEMENTATION OF CLIENT SIDE

The EPICS Archiver Appliance offers a Web UI for typical configuration tasks and data presentation. However, this Web UI doesn't meet our requirements in the PV classification and search. So we developed a new web application to provide real-time data display and historical data query as a Web-Based GUI.

The Web-based GUI is a Single-Page Application (SPA) which is more application-like and interacts with the user by dynamically rewriting the current page rather than loading entire new pages from a server, which is used as the client side to interact with users. The JavaScript framework Vue.js is used to develop the SPA. Vue.js is a progressive framework for building user interfaces and is also perfectly capable of powering sophisticated SPA when used in combination with Single File Components and supporting libraries [7].

Figure 3 shows the screenshot of the Web-Based GUI, which provides five tabs: Operation Status, Historical Data, Data Analysis, Alarm and Help. The front end runs in an Express web server and the data in the web page is obtained directly from the the Nginx via HTTP/JSON. The operator can select the PV by the check box in front of the node. The chart component on the right is the data visualization part based on Highcharts JavaScirpt library [4]. We can find the pv nodes in the tree, then select the check box in front of it, and the right side will show the curve during the last 24 hours. The buttons below provide many features, such as viewing and downloading raw data, setting coordinate axes, switching between linear and logarithmic coordinates, and so on.

SUMMARY

In order to facilitate the construction and commissioning of IRFEL facility, we developed a new system to archive, store, process and visualize the data. The back end service contains data from multiple sources. The front end web application provides a operation interface for users. Nginx acts as a reverse proxy server to connect the front and back-

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP06

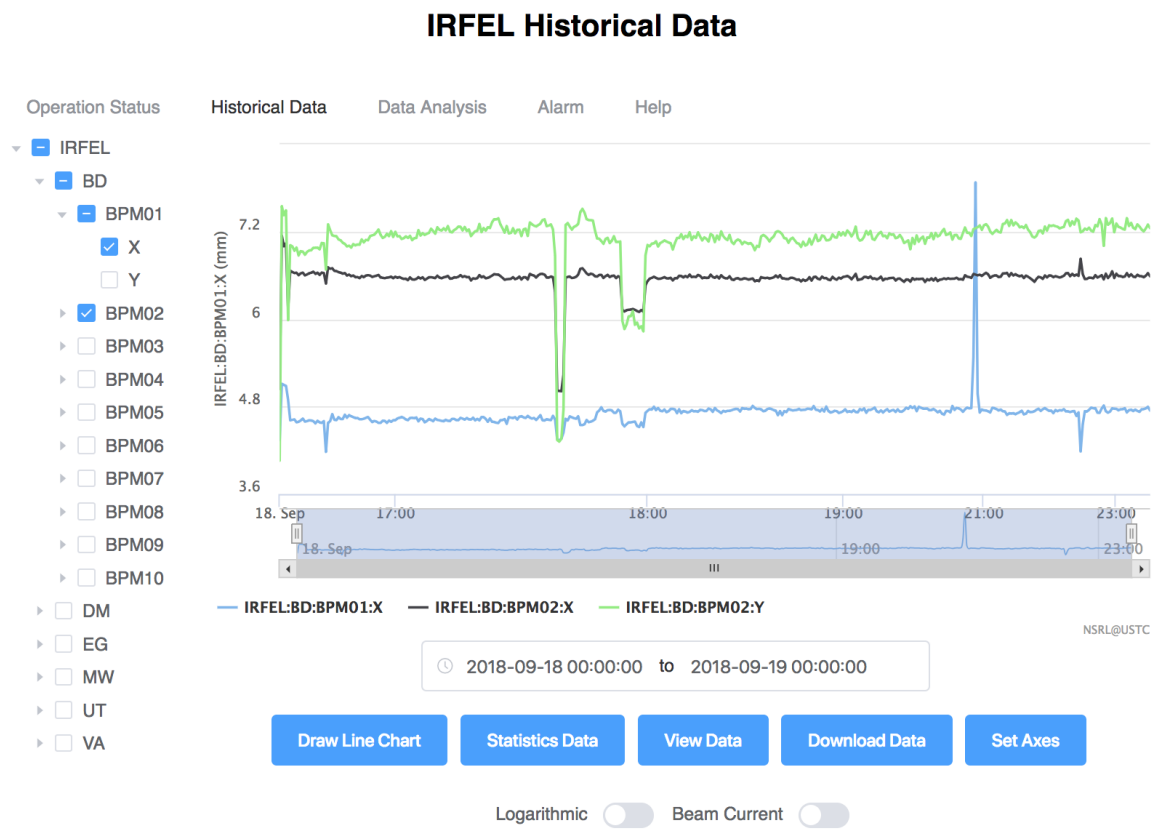


Figure 3: Screenshot of the Web-Based GUI.

end. The design of the complete separation of the front and back end is also convenient for future expansion, like adding new data types and adding new methods for interaction with users.

The system was put into operation in July 2018. The test results indicate that the new archiving and visualization system is reliable, flexible and convenient to operate.

REFERENCES

[1] H.T. Li, Z.G. He, Q.K. Jia, Q. Luo, L. Wang, and S.C. Zhang, "Status of FELiCHEM, a New IR-FEL in China", in *Proc. IPAC'16*, Busan, Korea, May 2016. doi:10.18429/JACoW-IPAC2016-MOPOW026

[2] S. Xu, G. Liu, Y. Song, and X.K. Sun, "Control System Design for Front End Devices of IRFEL", in *Proc. IPAC'18*, Vancouver, BC, Canada, Apr/May 2018, pp. 4920–4922. doi:10.18429/JACoW-IPAC2018-THPML109

[3] M. Shankar, L. Li, M. Park, *et al.*, "The EPICS Archiver Appliance", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015. doi:10.18429/JACoW-ICALEPCS2015-WEPGF030

[4] Highcharts home page, https://www.hcharts.cn/

[5] Websocketd home page, http://websocketd.com/

[6] Phoebus Documentation, https://phoebus-doc.readthedocs.io/en/latest/

[7] Vue.js home page, https://vuejs.org/

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP07

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

INNOVATIVE GRAPHICAL USER INTERFACES DEVELOPMENT: GIVE THE POWER BACK TO USERS

Gustavo Segura Millan, Adrien Ledeul,
Alexandru Savulescu, Bartlomiej Styczen, Daniel Vasques Ribeira
CERN, Geneva, Switzerland

Abstract

Graphical User Interfaces (GUIs) for Supervision, Control and Data Acquisition Systems (SCADA) are usually oriented to specialist users. In big organizations like CERN, where different teams play the roles of operators, scientists and instrumentation specialists, providing a unique or static user interface usually results in a situation of dissatisfaction for everyone. Providing distinct user interfaces for each type of user increases the development and maintenance effort and makes the software evolution heavier. The approach taken in the design and development of GUIs for radiation and environment protection at CERN addresses these issues by integrating user interface changes as an embedded software functionality. Key users were provided with a tool to build, deploy and maintain their own tailor-made user interfaces, in a graphical way and without the necessity of learning any kind of programming or scripting languages. Other benefits observed from this solution includes the reduction of the resources spent on the support and maintenance and the increase of the frequency of GUIs updates, executed without compromising the underlying control system. This paper describes the innovative design that was implemented.

INTRODUCTION

The Health Safety and Environment Unit at CERN provides a SCADA system for the radiation protection and environment monitoring of particle accelerators, experiments and the environment. The system called REMUS [1] (Radiation and Environment Monitoring Unified Supervision) runs 24/7 hours 365 days a year.

REMUS is a large system in its class (600.000 tags) interfacing 550 monitoring stations built convening 75 device types and 3000 measurement channels. Its GUI has 600 different synoptic views used by around 50 concurrent users from a total population of 200 authorised users. Those users have different roles, needs and expectations of the system.

Naturally, the GUIs require frequent updates to adapt to the ever changing CERN environment (installation of new instrumentation or regular changes of the instrumentation type or location).

The design of REMUS addresses two problems by design:

- Evolution and extension of the GUIs impact many users with different needs, sometimes conflicting.
- Frequent GUI updates are resource demanding tasks.

The REMUS approach to deal with these challenges is to provide the final users with the possibility of building their own GUIs.

BUILDING USER INTERFACES FOR SCADA SYSTEMS

The usual approach to design SCADA GUIs consists of making a schematic representation of the target process, installation or technical facility (synoptic views). Then placing widgets that represent devices, providing animated readings for supervision purposes and commands to control the underlying process. If the system is large, it may be necessary to present it in smaller parts or sub-systems in order to reach a lower level of details and to add a navigation mechanism that facilitates visualization and operation for users.

The attained level of detail is often the one of individual equipment, sensors and actuators under supervision and control from which the system is doing data acquisition. In order to complete the SCADA GUI it is necessary to add an access control mechanism to in order to grant the proper level of access rights to the authorised users.

BUILDING REMUS GUI

REMUS users have differing needs and expectations from the system. REMUS provides interfaces for operators of accelerators and experiments, physicists, instrumentation specialists, radiation protection engineers, environmental engineers, CERN Fire Brigade, instrumentation maintenance teams and the software support team. Providing a unique GUI for all these users regardless of their role is not a good approach as there is no interface that can suit everybody.

User interfaces need to adapt to the diverse user roles and require frequent modifications to handle the continuous changes in the supervised instrumentation.

The REMUS interface is organized in a tree-structure of synoptic views displaying CERN surface and underground areas. Widgets representing hardware devices and measurement channels are displayed on the synoptic views. There are different versions of widgets for each device type, addressing specific levels of detail required by different types of users.

REMUS Applications

In order to provide users with the most suitable interfaces, REMUS has a special kind of users named *Application Administrators*, who takes care of building interfaces for all other users. Administrators are neither part of the REMUS support team nor software developers.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP07

Administrators can build *REMUS Applications* composed of synoptic views, widgets and a customisable alarm screen. These views are built to supervise an infrastructure or process (e.g.: accelerator or experiment radiation, water or air release to the environment, etc.). Applications include a sub-set of the supervised instrumentation and an access control mechanism.

Users Build the User Interfaces

Application administrators are provided with a tool named *REMUS Application Editor* that allows them to create new applications or modify existing ones (Fig. 1).

The application editor is a user friendly graphical tool that does not require any knowledge of programming languages or scripting to be able to build or modify user interfaces. The editor user composes the interface using drag-and-drop, without writing a single line of source code.

The application editor functionality include:

- Building synoptic views and placing background images (e.g.: schema of process, experiment plan, etc.)
- Adding navigation among the synoptic views
- Selecting widgets (devices) from a catalogue and positioning them on the synoptic views
- Linking widgets to existing devices declared in REMUS database

Applications are stored in the form of xml files.

In order to modify a running Application, REMUS provides administrators with a *download* function that writes the application as an .xml file that can be edited and *uploaded* back to REMUS Servers. REMUS servers then update all the clients running the uploaded application in real-time, without the need to restart the clients.

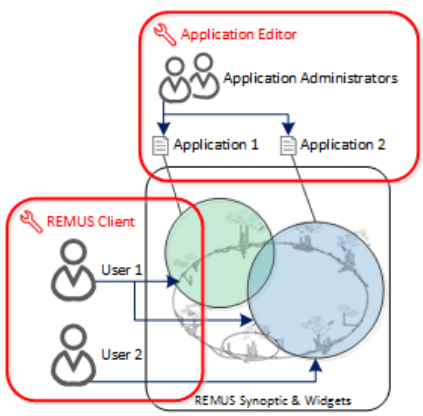


Figure 1: REMUS Application definition and scope.

REMUS automatically stores the successive versions of each application and makes all the versions available to administrators who can reuse them if necessary.

Access Control

Each application has its access control mechanism. Administrators can grant other users one of the following access levels:

- Visualize: real-time and archived data visualization.
- Acknowledge: visualize rights + the right to acknowledge alarms and system faults.
- Parametrize: acknowledge rights + the right to change device parameters.
- Configure: parametrize rights + the right of installing new devices.
- Admin: configure rights + the right to edit the application and giving access rights to other users.

Other Benefits

The outcomes of the REMUS strategy for building GUI are very positive:

- GUI support and maintenance efforts are distributed among Application Administrators that are close to the needs of the final users. Administrators are just final users with additional responsibilities.
- The GUI evolution and growth become the responsibility of the final users (administrators are users).
- Applications are tailor-made for specific kind of users and focus on the devices of their interest (only a sub-set of the total instrumentation) such as a specific accelerator or experimental area.
- The performance of the REMUS client software is improved, as the traffic between clients and server is reduced to the one generated by the devices displayed in the loaded client application.
- The REMUS support team focuses its effort on integrating new devices, developing widgets and improving core functionalities.

GUI FOR SAFETY SYSTEMS

Implementing changes in software used in safety systems differs from other type of systems in the process of test, acceptance and validation of the changes that affect safety functions. The processes for such systems are stricter, longer, more formal and tedious than for non-critical software systems.

GUI interfaces are not part of safety functions but they are very important especially when operators take safety-related decisions based on the information displayed by those interfaces.

REMUS as a Safety-related System

REMUS software runs in the CERN control rooms and provides operators with real-time information of the radiation conditions and of the release of effluents to the environment.

Based on the information provided by REMUS applications, operators can decide to stop accelerators and experiments for safety reasons or call for an intervention to stop or mitigate the pollution of the environment.

One of the particularities of REMUS is that the very frequent modifications of the user interfaces do not require extensive testing.

Modifying the GUIs in REMUS is, in most cases, just adding or modifying applications.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP07

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Changing or modifying a REMUS application is just using a function of the system that administrators can operate whenever they need to.

When a user needs to change a REMUS application, the user downloads the application from REMUS, edits and changes it with the Application Editor tool. The Application Editor software ensures that the application is valid and well-formed and writes it in the form of an .xml file. When the user uploads the modified application, REMUS parses it to construct the UI that becomes automatically available to all authorised user. The parsing process verifies that the application is coherent and well-formed and detects errors beforehand.

No further acceptance procedures are required for the validation of new applications or for implementing changes in existing ones. The benefits of this mechanism are:

- Reduction of the resources needed by the software development team in formal test and validation process.
- Increased user involvement in the process as they take the responsibility of the GUI construction.
- Increased user confidence in the system.
- Not extra software development required despite the dramatically increase of GUI modifications and updates.

CONTINUOUS OPERATION

At CERN, accelerators and experiments work 24/7 during run periods that last for months only interrupted by short technical stops taking place every few weeks and lasting for few hours. Long shutdown periods have a duration of a few months and take place between run periods.

Control systems remain unchanged during long run periods.

Generally, the deployment of new control software, upgrades and modifications takes place mainly during shutdowns in order to get the time necessary for extensive test and validation on quasi-real operational conditions.

The previous radiation and environment SCADA systems operated at CERN years ago (RAMSES [2], ARCON [3]) used to follow the same development and deployment cycle in which software upgrades used to take place during long shutdown periods. Only small modifications, typically related to GUI, were deployed in short technical stops.

REMUS allows modifying and deploying applications at run-time during accelerators operation without compromising the operation. The mechanism implemented to build and upload the applications ensures the integrity of the REMUS System at run-time (Fig. 2).

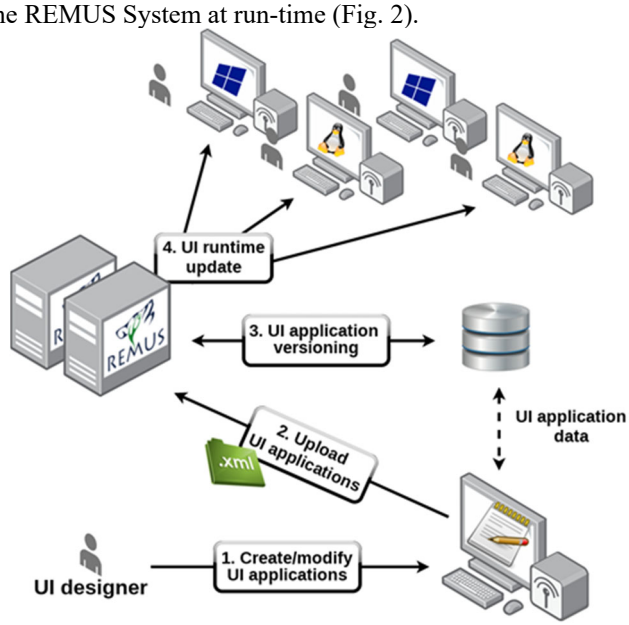


Figure 2: REMUS Applications edition.

CONCLUSION

The REMUS approach for the management of GUI has proven to be very versatile bringing many benefits to the system such as:

- Reduced software development and maintenance costs.
- Increased user satisfaction.
- Fast user adoption.
- Decreased "time to market" of new GUIs releases.

REFERENCES

[1] A. Ledeul, G. Segura, R.P.I. Silvola, B. Styczen, and D. Vasques Ribeira, "REMUS: The new CERN Radiation and Environment Monitoring Unified Supervision", in *Proc. 15th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'15)*, Melbourne, Australia, October 2015, paper TUD3O03, pp. 574-577, ISBN: 978-3-95450-148-9, doi:10.18429/JACoW-ICALEPCS2015-TUD3O03

[2] G. Segura Millan, D. Perrin, L. Scibile, "RAMSES: The LHC Radiation Monitoring System for the Environment and Safety", in *Proc. ICALEPCS'05*, Geneva, Switzerland, Oct. 2005, paper TH3B.1-3O.

[3] D. Forkel-Wirth, G. Segura Milan, D. Perrin, S. Roesler, C. Theis, H. Vincke, Hz. Vincke, P. Vojtyla, M. Widorski, "ARCON/RAMSES: Current Status and Operational Risk", Workshop on LHC performance 2010, Chamonix, France.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP08

A UNIVERSAL SYSTEM BASED ON WEBSOCKET AND JSON FOR THE EMPLOYMENT OF LabVIEW EXTERNAL DRIVERS

Alessandro Stecchi, Claudio Bisegni, Paolo Ciuffetti, Giampiero Di Pirro, Alessandro D'Uffizi, Francesco Galletti, Andrea Michelotti (INFN/LNF, Frascati (Roma))

Abstract

One of the heaviest workloads when installing a Control System on a plant is the development of a large number of device drivers. This is even more true in the case of scientific facilities for which you typically deal with many custom devices and legacy code. In these cases, it is useful to consider the Rapid Application Development (RAD) approach that consists in lessen the planning phase and give more emphasis on an adaptive process, so that software prototypes can be successfully used in addition to or in place of design specifications. LabVIEW [1] is a typical RAD oriented development tool and is widely used in technical laboratories where many standalone programs are developed to manage devices under construction or evaluation. An original system that allows software clients to use external LabVIEW drivers is presented. This system, originally created for the !CHAOS Control System [2], is entirely written in LabVIEW and is based on JSON messages transmitted on a WebSocket communication driving LabVIEW VIs through dynamic calls. This system is completely decoupled from the client and is therefore suitable for any Control System.

DESCRIPTION OF THE ELF SYSTEM

The project called ELF (External LabVIEW Functions executor) stemmed from the need to reuse as much as possible the huge amount of software — especially device drivers — already written in LabVIEW, to speed up the implementation of new controls currently being implemented with !CHAOS, a control framework developed at the National Laboratories of Frascati of the INFN.

As the project evolved, its usefulness was also evident not only for the re-use of existing LabVIEW Virtual Instruments (VIs) but also as a proficient method for the collaborative development of complex LabVIEW programs. In fact ELF provides for the VIs acting as drivers to be called by reference and not be wired into other G code, which greatly facilitates the team development.

Ultimately, this work consists in the realization of an environment able to have LabVIEW VIs execute in a well managed and standardized manner, upon calls coming from a client application written in any language.

The project requirements where:

- to adopt widely used communication protocol and data-interchange format for the communication between the client and ELF;
- to write the whole ELF code in G: the LabVIEW graphical programming language;
- to be able to deal with simultaneous calls from multiple clients;
- to adopt a unique template for the VIs to be executed and a well defined syntax for their call;
- to realize a modular architecture, allowing its usage both from non-LabVIEW and LabVIEW clients;
- to get such performance that it could be used in a wide range of applications.

It is worth to point out that the ELF employment is non restricted to the call of device drivers but extends to any VIs complying with the adopted template and syntax.

COMMUNICATION BETWEEN CLIENT APPLICATIONS AND ELF

First we must clarify what is meant — in this context — for client application (*client* from now on). We therefore define as client any program that accesses the ELF to perform a set of predefined functions implemented in LabVIEW and obtain results back, if any.

As an example, in our case the typical client is a !CHAOS Control Unit (CU): the control node that continuously acquire data from a device and operate it. To access a physical device, the CU performs a RPC by using a C++ *skeleton* that relays the function *opcode* along with its arguments to a *stub* that eventually runs the actual driver. In !CHAOS, drivers are usually implemented as C++ programs but, if they are available as LabVIEW VIs, the *skeleton* can direct the calls towards the ELF system which acts as *stub*.

The communication between a client and ELF meets the client/server model, with ELF playing the role of server, and follows the WebSocket [3] protocol.

WebSocket provides full-duplex communication between nodes over a single TCP connection with data minimally framed, to the benefit of real-time data transfer.

It provides a standardized way for the server to send content to the client without being first requested, and allowes messages to be passed back and forth while keeping the connection open. The WebSocket handshake starts with an HTTP request/response then — once the connection is established — the communication switches to a bidirectional binary protocol which doesn't longer conform to HTTP. This method is advantageous since HTTP *flows* through proxies and therefore it is possible to open connections even from remote clients laying within a NAT, which is very convenient for distributed system where remote control nodes can be spread anywhere (as in the case of !CHAOS).

As data-interchange format it has been adopted JSON (JavaScript Object Notation) because it is lightweight and also because JSON documents are ultimately strings, which facilitates the passage of different data formats to LabVIEW VIs.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP08

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

We built the ELF as a modular system made of two LabVIEW top-level VIs working jointly. The first module (named as EPROS: Elf Proxy Server) is a proxy server that connects in WebSocket to the client and interfaces it to the second module (named as SHIELD: Simple High-level Interface for Enhanced LabVIEW Drivers), where the various functions are executed. EPROS and SHIELD talk to each other by mean of JSON documents passed through LabVIEW queues.

This architecture, shown in Fig. 1, allows two different modes to access the ELF system:

- Indirect Mode: a client not written in G or even written in G but running in a LabVIEW session different from the ELF one's, will utilize WebSocket/JSON and both the EPROS and SHIELD modules;
- Direct Mode: should the client be a LabVIEW program running in the very same session of the ELF, it would be useless to go through WebSocket, being much more convenient to utilize native LabVIEW queues. In this case, the client will use just the SHIELD module and exchange the JSON messages directly through its queues.

VIs PREPARATION AND INSTALLATION

When dealing with legacy software or even software written by more than one person, it is essential to define strict guidelines that allow for its integration.

In ELF, the integration of drivers for a given device, provides for the establishment of a list of functions (namely the different actions that have to be performed on the device) and for each of these, the set of arguments to be passed and the returned results. Each function is identified by an *opcode*.

Once the opcode list has been defined, the LabVIEW developer has to prepare for each opcode a dedicated VI able to perform its function. All these opcode VIs must adhere to a template and have the same connector pane, with:

- a string input for the JSON document containing the parameters;
- a cluster containing all the variables (named as *service bus*) that the developer considers to be meaningful to describe the status of the device;
- a string output for the JSON document containing the results (or errors) to be returned to the client.

If the developer writes the opcode VIs starting from scratch, then he will edit them directly according to the template, otherwise he will have to readjust the code he already owns, for example by encapsulating it in *wrapper* VIs compliant with the template.

Once the opcodes have been prepared, they must be *installed* in the ELF system. As already mentioned, ELF is designed to completely decouple the opcode VIs both from the client and from itself. Therefore, the opcode VIs installation comes down to their copy in a predefined directory tree, so that they can be located and launched *by reference* by the SHIELD in accordance with the JSON directives coming from the client (see Fig. 1).

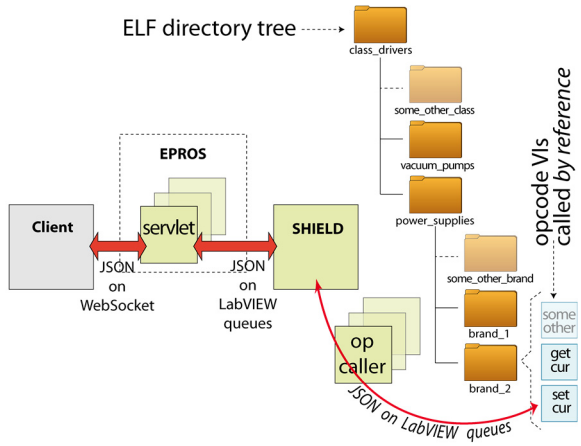


Figure 1: The ELF system. The installation of drivers (compliant with the ELF guideline) comes down to their copy in a directory of the ELF tree.

The ELF directory tree mirrors the grouping of devices into abstract classes.

A class of devices is set out by a virtualization process that — starting from a physical object — concludes with the definition of a set of meaningful variables (*dataset*) that fully describe it from the vantage point of the control and a set of actions (*commands*) you want to perform on it. In that way, we can bring together — for instance — different brands/models/types of power supplies by identifying them with the same dataset and commands. Therefore, the directory tree shall consist of a first level of directories corresponding to the abstract classes of devices, each of which contains other directories that are specific to the different brands/models/types of real power supplies.

Ultimately, the developer of opcode VIs has to prepare them in accordance with the specifications and drop them in the proper folder. It is important to point out that, although the directory structure is defined, its content is not. So, if no branch of directories is available for the class of elements to be installed, the developer is free to create it and populate it with its opcode VIs.

At the end, the new set of opcode VIs can be located by mean of a *uri* in the form:

<base_path>/class_drivers/power_supplies/brand_1/

CLIENT-ELF WORKFLOW

There is a slight difference in the sequence of actions a client must perform, according to whether he accesses the ELF system in direct or indirect mode. Direct access is in fact reserved for clients running in the same LabVIEW session of ELF (more precisely, of SHIELD). This means that both client and SHIELD are running on the very same machine, which makes the connection and authentication steps unnecessary. The workflow for indirect access is therefore described, since it also covers the direct mode.

WebSocket Connection

The client requires a WebSocket connection to the EPROS by sending a WebSocket header (this is a pure HTTP request) with the *uri* that identifies the drivers for the desired class. The EPROS checks the header format and

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP08

asks the SHIELD to confirm that the required VIs exist. If no errors occur, the EPROS accepts and establishes the connection.

After receiving the HTTP header, all the subsequent phases of the workflow are carried on by exchanging proprietary JSON messages.

Authentication

The client sends an authentication request containing a shared key; the EPROS checks the key and if no errors occur, grants access to the client. In order to serve concurrently multiple clients, the EPROS starts *by reference* an instance of servlet dedicated to that very client. From this moment on, the EPROS servlet behaves as a PROXY, since it limits itself to relay back and forth the messages between the client and the SHIELD and to manage any communication error.

Initialization

When a client transmits an initialization request with the *uri*, the SHIELD opens an instance of a VI (named as *op-caller*) that will manage all the opcode requests for the given device. It also creates a UUID associated to the instance of the op-caller and returns it back, appended to the uri. The concatenation of the uri with the UUID is essential to be able to later discriminate equal instances but related to different devices of the same class. At this point SHIELD use the op-caller to launch *by reference* the opcode VI that performs the initialization of the physical device and — where appropriate — of the channel it communicates through.

Full Duplex Data Flow

This is the operational phase, where the client has the ELF performing the desired actions by sending JSON request messages in the form:

```
{
  "req_id":[int32],
  "msg":{
          "uri":[string],
          "opc":[string],
          "par":{...json document...}
  }
}
```

where:

- "req_id" is a transaction identifier;
- "uri" identifies the specific op-caller instance;
- "opc" specifies the action the external driver has to do;
- "ele" identifies the element (e.g. its address);
- "par" is a JSON document containing the parameters associated to the opcode and the specific element identifier (e.g. its address).

For instance, the message:

{"req_id":123456,"msg":{"uri":"/LabVIEW_external_drivers/power_supplies/brand_1/FFA0C0D8","opc":"set_cur","par":{"ele":"01", "value":10.5}}}

calls the opcode "set_cur" with the parameter "value" equal to 10.5, which cause to set the power supply current to 10.5 [A].

At each request message, SHIELD handles the client call by relaying it to the op-caller that calls by reference the appropriate opcode VI.

All client requests are always answered with frames in the form:

```
{
  "req_id":[int32],
  "msg":{
          "err":[int32],
          "err_msg":[string],
          "err_dmn":[string],
          "result":{...json document...}
  }
}
```

where:

- "req_id" is a transaction identifier that has to be equal to the transmitted one;
- "err" is the error code (0 = no error);
- "err_msg" describe the error (present only if err ≠0);
- "err_dmn" is the error domain (present only if err ≠0);
- "result" is a JSON document containing the result returned by the opcode VI. This field is present only if the opcode VI provides for some values to be returned.

Deinitialization

The client asks for the deinitialization of the physical device (if applicable) and the release of associated resource. Consequently, the SHIEL destroys the op-caller instance as well as all the related opcode VIs hanging in memory and remove the UUID from its internal list.

WebSocket Disconnection

The client sends a standard WebSocket "close connection" message which is answered by the EPROS that eventually will also drop the TCP/IP connection.

EPROS AND SHIELD

The EPROS Module

As said above, a WebSocket connection starts with an HTTP request and then switches to WebSocket over the same underlying TCP/IP connection. This means that on the server side there must be an always running TCP listener, to accept requests from various potential clients, as well as various WebSocket servers running, to service those already connected clients. The EPROS is therefore made of a top level VI that keeps listening and, when a TCP connection request arrives from a client, launches by reference an instance of a WebSocket server that take charge of that client. This instance (named as *servlet*) is a replica of a VI Template (VIT) and has a dataspace completely independent from the other servlets.

The servlet is made of four parallel loops data-independent form one another that can concurrently execute the following tasks:

- check and serve WebSocket connection and authentication requests;
- receive WebSocket request messages;
- communicate with the SHIELD through LabVIEW queues;
- send WebSocket answer messages to the client.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP08

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

The four loops pass information to one another by using queues, which is fast and maintain the data-independency among them.

The low level WebSocket read and write functions have been written from scratch and cover the following WebSocket actions: read text frame (also multiframes), ping, pong, close connection.

The SHIELD Module

The SHIELD must be able to manage multiple clients at the same time. To do that,

when a client asks for the initialization of a given device, it launches by reference a dedicated instance of op-caller that will handle all the calls for that device. There is a one-to-one correspondence among EPROS servlets and SHIELD op-callers, being each pair the two endpoints of a channel dedicated to a specific device.

When a call with a given opcode arrives for the first time, the op-caller starts by reference the opcode VI which is in the folder specified by the uri. The opcode VI executes the action and then stays idle in memory, ready for the next call with the same opcode. To permit this, all the opcode VIs must be set as "pre-allocated clone reentrant execution" which provides for a mutual data-independency and no loading time at call.

The SHIELD provides also for the garbage collection of dandling instances.

SYSTEM PERFORMANCES

The ELF system has been tested to understand its potential fields of application.

We therefore made some measurements to determine the frequency of execution of an opcode VI upon continuous calls coming from a client.

This figure depends on the client performance, the rank of the machines running both client and ELF system, the network and obviously the intrinsic execution time of the opcode VI that can vary widely. An opcode VI could in fact run on a remote machine and having to drive a device connected through low baud-rate serial line or run on embedded processor and having to simply set a register of a board resident in its own bus.

The measurement was therefore made net of the opcode execution time, by using a dummy VI that always returned a constant value to the get_curr (get current) opcode.

The client application was a standard !CHAOS Control Unit running on a virtual machine and the ELF system was running on another virtual machine, with ping time between the two of ~500 us. The read process can be reduced to three main steps:

- the client sends a WebSocket message with the opcode "get_curr" to the servlet;
- the opcode VI executes and returns the value to the servlet;
- the servlet sends a WebSocket message with the result back to the client.

Therefore, the full query consists of two WebSocket transmissions that are performed synchronously by the Control Unit. In this condition the overhead for the query is ~1 ms. The measure has been done by varying an internal parameter of the CU (sleep-time) that allows the modulation of its running frequency. It can be seen in Fig. 2 that as the CU sleep-time decreases the frequency of calls increases up to 1 kHz, which is consistent with the overhead of 1 ms due to network latency.

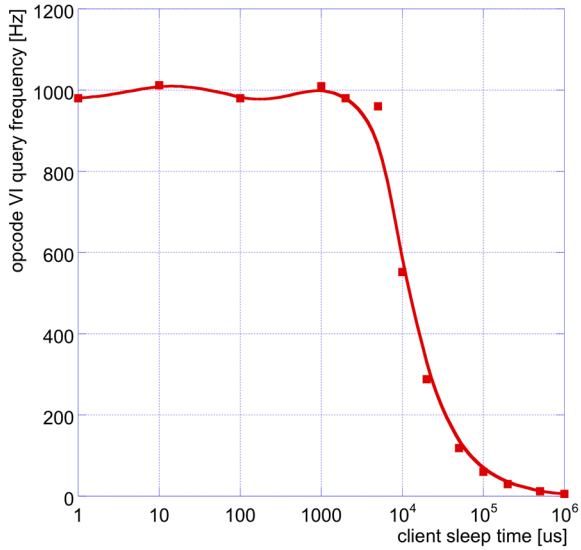


Figure 2: Frequency of calls of a LabVIEW driver.

The sharp breakthrough at 1 kHz must not be considered as a limit of the ELF system because it is due to the CU running synchronously and therefore executing the queries in a series. We are going to do more tests with asynchronous calls in order to work out the actual limits of the system.

The SHIELD module has also been tested in direct mode using a minimal LabVIEW client written on purpose. In this case the process takes 100% of the CPU and the frequency dramatically increases, according to the CPU performances.

REFERENCES

[1] LabVIEW, http://ni.com

[2] L. Catani *et al*, "Introducing a New Paradigm for Accelerators and Large Experimental Apparatus Control Systems", *Phys. Rev. ST Accel. Beams,* Nov. 2012, vol. 15, p. 112804.

[3] WebSocket, https://tools.ietf.org/html/rfc6455

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP09

SWISSFEL ELECTRON BEAM DIAGNOSTICS TOOLS AND THEIR CONTROL SYSTEM COMPONENTS

P. Chevtsov†, D. Anicic, V. Arsov, Paul Scherrer Institut, 5232 Villigen PSI, Switzerland
M. Dach, Dach GmbH, 5200 Brugg, Switzerland

Abstract

The main driving part of the X-ray free electron laser facility (SwissFEL) at Paul Scherrer Institute (PSI) is a compact electron linear accelerator (linac). The machine is highly optimized to generate a superior FEL radiation with the lowest suitable electron beam energy. In order to meet extremely stringent SwissFEL require-ments for electron beam quality and stability, a variety of advanced beam diagnostics tools were developed and implemented at PSI. All these tools are integrated into the SwissFEL control system. The paper describes basic control elements of advanced electron beam diagnostics tools and their operational performance.

INTRODUCTION

The X-ray free electron laser facility (SwissFEL) at Paul Scherrer Institute (PSI) is driven by a normal conducting linear accelerator (linac), which generates electron bunches with a repetition rate of 100 Hz and charges in the range from 10 pC to 200 pC. In order to provide user experiments with extremely stable, brilliant and ultrashort (~10 fs) photon pulses, which are tunable in the whole wavelength interval from 0.1 to 7 nm, the operations of the SwissFEL machine pose very challenging demands on a number of relevant electron beam parameters, such as the bunch energy, charge, length, arrival time, compression, transverse position, profile, etc. A series of Beam Diagnostics Monitors (BDMs) developed at PSI allows one to obtain all those parameters online. Each BDM is a beam instrumentation device dealing with a particular beam parameter (e.g., arrival time).

In the SwissFEL control system environment, BDMs are implemented as Beam Diagnostics Tools (BDTs), which are based on the PSI Generic Beam Diagnostics Electronics (GBDE) platform in the VME standard and Advanced Data Processing and Control System (ADPCS) setup concept.

GENERIC BEAM DIAGNOSTICS ELECTRONICS PLATFORM OVERVIEW

The GBDE platform is schematically shown in Fig. 1. It combines BDM specific detectors and front-end electronics with common solutions for digitization, FPGA based digital signal processing and interfacing [1]. The platform is very efficient for interacting with control systems.

The analogue signals from BDM detectors are collected by front-end electronics, which is implemented as mezzanine boards sitting on standard PSI Analogue Carrier (PAC) cards. The electronics is in charge of signal conditioning, which is detector specific and typically deals with the optimization of dynamic ranges, signal-to-noise performance, input/output levels, etc.

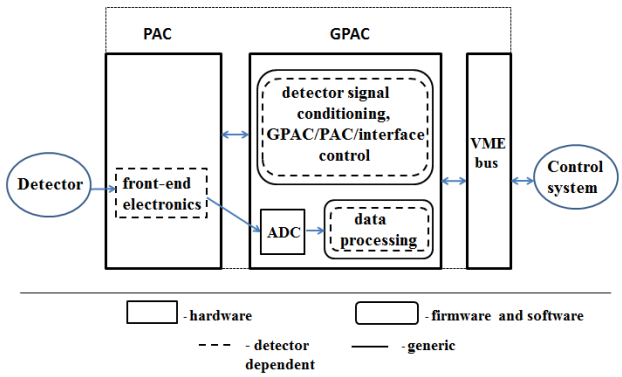


Figure 1: PSI GBDE platform schematics.

PAC output signals are read out by Generic PSI ADC Carrier (GPAC) cards, where they are digitized by fast ADC mezzanine boards and then processed digitally. The major ADC board type that is used for SwissFEL beam diagnostics applications has eight 12 bit 500 Msps ADCs. The ADC sampling clock is provided externally by the SwissFEL reference distribution system [2] or a locally generated BDM clock signal. GPAC cards also regulate GPAC working parameters and handle the communication with the external control system (via the VME bus) and electronics, such as, for instance, PAC cards. All GPAC control and data processing functions are supported by the firmware and software running on several FPGA and embedded CPU (PowerPC) modules.

Two VME bus memory blocks are allocated for the communication of GPAC cards with the control system. Memory mapped GPAC control and status registers make up the GPAC Control Block (GCB). The GPAC Data Block (GDB) comprises the results of signal processing by GPAC firmware and software.

We note that the GPAC firmware, which is in charge of the communication with the control system, is common for beam diagnostics applications. In particular, it maps GPAC control and status registers into GCB, timely reacts on commands from control applications by monitoring this block, and generates the VME bus interrupts. On the other hand, the ADC outputs are processed by firmware and software, which are detector dependent. In major applications, the data processing is triggered externally.

The GBDE platform gives beam diagnostics physicists and electronics engineers a generic way of the BDM development. As the result, their main efforts are concentrated on BDM specific electronics, firmware and software.

† pavel.chevtsov@psi.ch

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP09

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ADVANCED DATA PROCESSING AND CONTROL SYSTEM SETUP CONCEPT

The described above GBDE platform provides a natural, VME bus based, way for integrating BDMs into the control system. The platform also makes it possible for beam diagnostics applications to perform a lot of signal and data processing functions directly on the used GPAC cards. This allows notable computing resources of input-output controllers (IOCs) of the control system, which are usually engaged for such functions, to be allocated for advanced data processing and control tasks.

The integration of BDMs into the SwissFEL control system as well as the development and deployment of advanced beam diagnostics data processing and control tasks in the control system environment are significantly simplified with the use of the Advanced Data Processing and Control System (ADPCS) setup concept, which was formulated and realized at PSI.

Based on this concept, each BDM is supported by its dedicated ADPCS setup. Each setup consists of common and BDM specific software and hardware.

Common ADCPS hardware elements are an IOC and an event receiver (EVR) card, which are placed into one VME crate together with the associated GPAC and PAC cards. For the SwissFEL project, such a GPAC handling IOC is IFC 1210 intelligent FPGA controller board (from IOxOS) with a Power PC dual core processor. The EVR is a new generation EVR-VME-300 card (from Micro-Research Finland), which interfaces the SwissFEL timing and event distribution system.

The ADPCS setup software consists of three parts: GPAC Interface Control (GIC), Operational Conditions Control (OCC) and Advanced Data Processing (ADP) software. The development of ADPCS software is done in the frames of the PSI application development environment that uses Git for software revision control as well as powerful software installation and loading tools [3]. The core of this environment is the EPICS toolkit that provides control system developers with a variety of robust control solutions.

Several software packages were developed at PSI to support ADPCS setup software.

The software of VME bus Memory Handling (VMH) package is implemented as an extension of a generic device support software module (regDev) on top of the memory mapped (mmap) access and IOxOS FPGA (Tosca) drivers [4]. This software allows one to efficiently handle VME cards with the use of standard EPICS records directly referencing registers and memory of those cards. The templates of such records and numerous examples of their use are the essential part of the package.

A wide variety of scientific data manipulation functions are provided by the Mathematical Support (MS) package, which is based on the GNU Scientific Library [5]. The most important functions for beam diagnostics applications are on-line statistics (e.g., mean, standard deviation, variance, higher moments, etc.) in specified time frames and signal fitting (e.g. Gaussian). All these functions are engaged as parts of standard EPICS subroutine records.

The Instrument Interface Handling (IIH) package provides a standard way of dealing with serial (RS232) devices. Based on the StreamDevice software [6], this package reduces all related control development process to few standard steps: listing serial device commands into a standard protocol file, creating an EPICS database with the use of record templates referencing those commands, and running this database on any IOC, which has a serial port available for interfacing the device.

The Embedding MATLAB (EML) package allows one to convert MATLAB programs in C codes, compile them, embed into EPICS and run on any IOC. As the result, closed loop control applications, which are written in MATLAB and have moderate (around 100 Hz) update rates, can achieve a time deterministic performance [7]. The interaction between ADPCS software parts and support packages mentioned above are schematically shown in Fig. 2

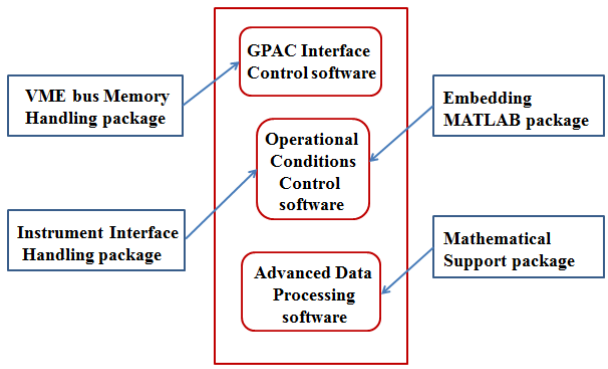


Figure 2: ADPCS software parts and their support packages.

The GIC software deals with the communication between the IOC and GPAC. It is represented by EPICS records directly associated with GPAC registers and data by pointing their VMH package based device/driver support software to the corresponding parts of GCB and GDB VME bus memory blocks. This software allows the control system to handle GPAC/PAC electronics and access BDM data processed by GPAC firmware/software. We note that EPICS records associated with GPAC registers are common for all BDMs, whereas records dealing with GPAC data are BDM specific but can easily be created based on standard templates.

In order to provide reliable beam diagnostics parameters, the operational conditions of BDM detectors and their electronics (e.g., temperature, humidity, voltages, etc.) must be continuously monitored and kept close to their optimal values. In the frames of ADPCS setup concept, this task is performed by the BDM specific OCC software supported by the IIH and EML packages. As a rule, this software runs on IOCs directly interfacing the BDM specific control hardware.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP09

Any additional data transformation functions, which are needed to be applied to BDM measurement results, are delegated to the ADP software, which is supported by the MS package and can run on any control system IOC. The common part of this software is represented by a set of EPICS records providing BDM data statistics in any specified time frames (e.g., 1 hour, 24 hours, etc.).

The ADPCS setup concept gives control system engineers a standard way of dealing with PSI advanced beam diagnostics monitors. This way significantly simplifies the integration of such monitors into the control system environment.

In order to keep the overall SwissFEL beam synchronous data consistency, beam diagnostics systems have to provide a deterministic time response at 100 Hz beam repetition rate.

If control and data processing algorithms for such systems are relatively simple, as in case of Beam Loss Monitors [8], then their realization on GPAC FPGA systems is straight forward and fully covers the required time deterministic aspects.

However, any complex control or data processing algorithm requires a lot of effort to be implemented and maintained on FPGAs. In these conditions, the ADPCS setup software allows one to achieve the required time deterministic aspects by running such algorithms directly on EPICS IOCs under RT patched Linux. As a result, all development and maintenance problems are significantly simplified.

ADPCS setup software and hardware implementations for SwissFEL beam diagnostics applications demonstrate how EPICS, which is intended for slow control tasks, can be enhanced for dealing with time deterministic aspects in conditions of 100 Hz beam synchronous operations.

For example, the Bunch Arrival-time Monitor [9] measurement result is a combination of two readouts. The first one is the GPAC ADC output, which provides the modulation of the reference laser pulse. The second one is the position of the MX80 motor (from Parker), which is used for zero crossing feedbacks. In the SwissFEL, there are movable chicanes, which influence the bunch arrival time for many tens of picoseconds. These arrival time changes are measured by the motor positions, which should be read out in a time deterministic way. The MX80 is controlled via RS232 but its controller can output raw encoder quad signals, which are counted by the ECM514 (from Kramert) VME card in a time deterministic (triggered) mode. As the result, the positions, which carry the arrival time information, are measured time deterministically.

ACKNOWLEDGEMENTS

The authors are grateful to D.Zimoch, D. Llorente Sancho, G. Marinkovic and T. Sustar for their help with GPAC settings and operational support.

CONCLUSION

The PSI Generic Beam Diagnostics Electronics platform and Advanced Data Processing and Control System setup concept allow one to develop beam diagnostics tools based on generic hardware and software solutions, which are powerful, reliable and easy to implement. Being automatically integrated into the accelerator control system and providing deterministic time response functionalities synchronized with the SwissFEL electron beam, such tools are valuable instruments for online machine setting up and operations.

REFERENCES

[1] W. Koprek *et al.*, "Overview of applications and synergies of a generic FPGA based beam diagnostics electronics platform at SwissFEL", in *Proc. IBIC'15*, Melbourne, Australia, Sep. 2015, pp. 165-169.

[2] S. Hunziker *et al.*, "Reference distribution and synchronization systems for SwissFEL: concept and first results", in *Proc. IBIC'14*, Monterey, CA, USA, Sep. 2014, pp. 29-33.

[3] R. Krempaska *et al.*, "Control system configuration management at PSI Large Research Facilities", in *Proc. ICALEPCS'13*, San Francisco, CA, USA, Oct. 2013, pp. 1125-1126.

[4] regDev, mmap, tosca, https://github.com/paulscherrerinstitute

[5] GSL – GNU Scientific Library, https://www.gnu.org/software/gsl

[6] StreamDevice 2, http://epics.web.psi.ch/software/streamdevice

[7] P. Chevtsov *et al.*, "MATLAB control applications embedded into process control controllers (IOC) and their impact on facility operations at Paul Scherrer Institute", in *Proc. ICALEPCS'17,* Barcelona, Spain, Oct. 2017, pp. 416-419.

[8] C. Ozkan Loch *et al.*, "System integration of SwissFEL beam loss monitors", in *Proc. IBIC'15*, Melbourne, Australia, Sep. 2015, pp. 170-174.

[9] V. Arsov *et al.*, "First results from the bunch arrival-time monitors at SwissFEL", in *Proc. IBIC'18,* Shanghai, China, Sep. 2018, paper WEPA20.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP10

UPDATE ON THE STATUS OF THE FLUTE CONTROL SYSTEM

W. Mexner*, E. Blomley, E. Bründermann, C. Fehlinger, A.-S. Müller, R. Ruprecht,
T. Schmelzer, M. Schuh, N.-J. Smale, Karlsruhe Institute of Technology, Karlsruhe, Germany
S. Marsching, aquenos GmbH, Baden-Baden, Germany
I. Kriznar, Cosylab, Ljubljana, Slovenia

Abstract

The first phase of FLUTE, a new linac based test facility and THz source, is currently being commissioned at the Karlsruhe Institute of Technology (KIT). It consist of an RF photo gun and a traveling wave linac accelerating electrons to beam energies of 40 to 50 MeV. The control system is based on a virtualized infrastructure running Ubuntu Linux and Linux KVM. As base for the SCADA system we use EPICS 3.15 with Control System Studio (CSS) for the GUI. The long term data storage is provided by a Cassandra NoSQL database. This contribution will present the architecture and the current status of the FLUTE control system.

INTRODUCTION

FLUTE [1] is a new R&D linac accelerator (see Fig. 1) offering beam energies of 7 to 41 MeV for the development of accelerator technology, new diagnostics and instrumentation for fs bunches. It will be used as a test facility for the study of bunch compression with all related effects and instabilities like space charge, coherent synchrotron radiation (CSR) as well as the different generation mechanisms for coherent THz radiation. Furthermore it will serve as a broad band accelerator-based source for ultra-short and intensive THz pulses, e.g. for time- & frequency-domain spectroscopy of kinetic processes.

CONTROL SYSTEM OVERVIEW

The design of the FLUTE control system [2] is based on EPICS 3.15 with Control System Studio (CSS) as the main operators interface to the control system.

The DAQ has been driven by the demand to make systematic studies of the beam compression and the generation of THz radiation. The pulse synchronous (10 Hz) data-acquisition based on a Cassandra NoSQL database has to record for each pulse diagnostic information about RF pulses, laser pulses, pulse charge and of course the overall accelerator settings.

Operator GUI Concept

The top level panels of the operator GUI have an synaptic approach. A 3D model of the Accelerator is used to for easy orientation (see Fig. 2). An error is visually connected to the device and to the position along the machine. The corresponding control panel for the device can be opened by a direct link on the panel. In addition all device panels are included in a list, which can be used to navigate to the devices, grouped by their task. As the accelerator is in its first phase the starting overview panel shows the injector section. For each of the next sections there are also synaptic overview panels planned as well as for the complete machine.

Stepper Motor Control

The FLUTE accelerator and the KARA Beamlines are using the same inhouse developed stepper motor control and driver concept. For motion control, we use the OMS (Pro-Dex) 5 axis Ethernet based MAXNET controllers and as 2 phase motor drivers we use the BCD130.x family of MIDDEX electronic allowing up to 12800 micro steps per revolution. EPICS hardware integration was done with the Pro-dex OMS asyn motor base axis support. For axes using an encoder, we experienced the problem that the motors would sometimes move randomly. We could solve these problems by applying a patch that synchronizes the motor with the encoder position before every movement.

Timing System

The timing system for FLUTE is based on the hardware components from Micro-Research Finland [3]. We use the VME form-factor for both the event generator (EVG) and the majority of event receivers (EVRs). However, we do not use the VMEbus for controlling these boards. Instead, we use their Ethernet interface to control them using a UDP/IP-based protocol.

* wolfgang.mexner@kit.edu

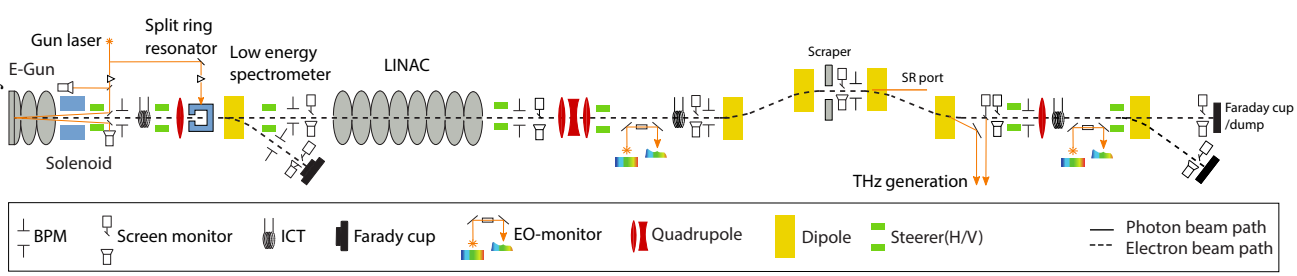


Figure 1: Scheme of the FLUTE accelerator with all installed and planned components [1].

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP10

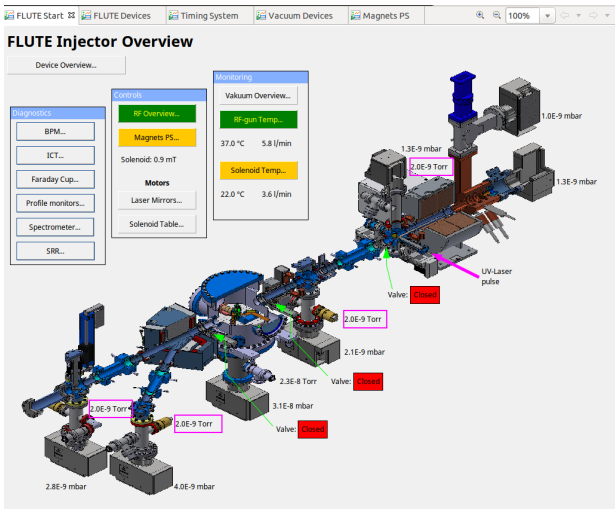


Figure 2: CSS Synaptic Operator view.

As the traditional EPICS device support for MRF devices [4] did not support this way of controlling the devices and its architecture did not include a layer for implementing different ways of transport, we developed our own EPICS device support [5] which includes such an abstraction layer.

Thanks to the flexibility of this new device support, we can essentially use the same code for controlling MRF boards over the network and controlling them locally over PCIe, which we need for some µTCA EVRs. The same device support is also used for the timing system of KIT's KARA facility, where it has proven to work reliably as well.

RF System

The RF uses a pulse type Klystron to produce a 4 µs long 3 GHz burst of 45 MW of power. This power is split between the E-Gun and the LINAC. The power supplies, for charging the Pulse Forming Network (PFN) and controlling the klystron coils are all slow control using PLCs, which are fully integrated into EPICS and CSS. However, MTCA based controllers such as the laser synchronization feedback loop, Low Level Radio Frequency feedback loop (LLRF) , and BPM electronics (readout via MTCA) which were developed for the FEL at DESY are not yet fully integrated to EPICS. As a stepping stone to full integration, these devices are presently used with DOOCS and JDDD installed on the devices. For standard configuration settings a remote desktop is used to access the JDDD panels. For the 'daily operator use', the more used configuration parameters have been integrated to EPICS and CSS using either gateways or Chimera TK adapter (the latter developed in cooperation by aquenos, DESY, HZDR and TUD). These methods are also used to transfer fast data capture to the operator display and database for storage. This is also true for the timing distribution system which, because of historical reasons, have a mix of the MRF VME and the X2 MTCA timing systems.

The software of the LLRF systems is based on the ChimeraTK framework [6]. Thanks to the use of this framework, essentially the same software can be used in control-system environments based on DOOCS, EPICS, or OPC-UA, making it possible to use the same hardware and software in a number of facilities like ELBE, the European XFEL, FLASH, FLUTE, and TARLA with only minor tweaks. This has helped to reduce the man-power needed to commission the system at FLUTE significantly.

SERVER INFRASTRUCTURE

Most services needed for FLUTE run within virtual machines on a virtualized infrastructure. We use the Linux Kernel-based Virtual Machine (KVM) [7] for hosting both Linux and Windows virtual machines (VMs). We use Ubuntu 14.04 LTS as the operating system for the VM hosts and manage the VMs using libvirt [8]. To ensure that basic services like network gateway (firewall) and DNS are always available, these essential services are provided by multiple VMs, hosted on different host systems. These host systems use a dedicated direct link for heartbeats, so that high-availability management is not affected by disruptions of the networking service.

This combination has proved very reliable: This virtualized infrastructure (based on two host machines) has been running for four years while needing virtually zero maintenance. In the near future, we plan to migrate the system to Ubuntu 18.04 LTS so that we can continue to apply critical security updates past April 2019.

We assessed that virtualization is not the best option for services with significant demands on the CPU or memory. For such services, sharing hardware resources does not make sense because it would only increase the demands on the virtualization hosts' hardware, while not profiting from the sharing of resources (which is the primary objective of virtualization in the first place).

For this reason, we also use three Supermicro Microcloud chassis [9], each containing 12 nodes. Each node is equipped with a Xeon E3 processor, 32 GBs of memory and two hard-disk drives. We use these nodes to run services like the Cassandra PV Archiver [10] that provides long-term storage of process data and the service for processing camera images.

NETWORK INFRASTRUCTURE

In order to improve the reliability and security of the network services, we use a dedicated network for all measurement and control equipment of the accelerator. The network is completely separated from the institute's office network. The VM hosts are based in both the dedicated accelerator network and the office network, so that selected VMs can provide services for transferring data between the two networks in a secure, well-controlled fashion. One example of such a service is the EPICS PV Gateway [11] that enables users to access EPICS process variables from their office computers, but limits all access to be read-only, thus ensuring no one can accidentally interfere with the operation of the accelerator.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP10

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

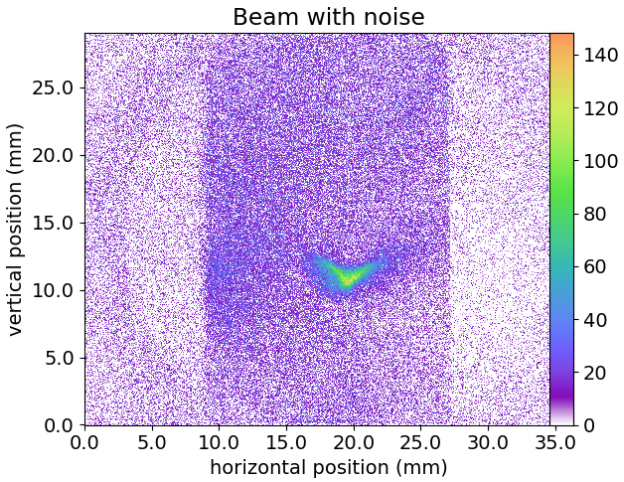


Figure 3: First FLUTE Beam.

CSS CLIENT DEPLOYMENT

The deployment of CSS consists of two separate, but closely related tasks: First, deploying CSS itself (the actual software along with its specific configuration for the environment), and second deploying the panels used inside CSS. The first task could be accomplished through means of the operating system (e.g. packaging and deploying CSS as a Debian package), but the panels change too frequently to make this a feasible approach for them. In addition to that, due to its heritage from Eclipse, CSS works around the concept of a "workspace" and no two instances using the same workspace can run at the same time.

These issues caused us to develop a couple of Python scripts that take care of installing or updating CSS, preparing a fresh workspace on each start, and cleaning up this workspace after CSS is closed. These scripts were originally developed for use at KIT's KARA facility, but they were quickly and easily adapted for use at FLUTE. These scripts are distributed alongside the panels as part of a Subversion repository. When the local working copy of the repository is updated, both the panels and the scripts are updated and thus the version of the panels and the version of CSS that is automatically installed by the scripts always match. By creating a fresh workspace on every start, the operators always start with the same, familiar view (Fig. 2), regardless of which panels have been opened or closed previously. This was a crucial factor to improve the acceptance of CSS as the user interface for operators.

SUMMARY AND OUTLOOK

In May 2018 first electron beam was produced and observed by different diagnostic systems such as the screen monitors (see Fig. 3), beam position monitors, integrated current transformer, and Faraday cup. Based on the recorded data the different systems can be calibrated which is needed for the full integration into the control system. At FLUTE the first user's experiment is foreseen in 2019 within the ARIES Transnational Access programme funded from the European Union's Horizon 2020 R&I programme under GA No 73 08 71.

ACKNOWLEDGEMENT

The authors would like to thank DESY's MSK group and KIT's Institute for Data Processing and Electronics for their support regarding the RF system. They would also like to thank PSI's diagnostics group for their help regarding beam diagnostics.

REFERENCES

[1] A. Malygin *et al.*, IPAC'18, Vancouver, 2018, TPHMF068. A. Malygin, A. Bernhard, E. Bründermann, A. Böhm, S. Funkner, I. Križnar, *et al.*, A. Malygin *et al.*, "Commissioning Status of FLUTE", in *Proc. IPAC'18*, Vancouver, BC, Canada, Apr/May 2018, pp. 4229–4231. doi:10.18429/JACoW-IPAC2018-THPMF068

[2] S. Marsching *et al.*, "Status of the FLUTE Control System", in *Proc. PCaPAC'14*, Karlsruhe, Germany, 2014, paper WPO013.

[3] Micro-Research Finland, http://www.mrf.fi/

[4] EPICS mrfioc2 device support, http://epics.sourceforge.net/mrfioc2/

[5] EPICS MRF device support provided by KIT, https://github.com/kit-ibpt/epics-mrf

[6] M. Killenberg *et al.*, "Abstracted Hardware and Middleware Access in Control Applications", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017. doi:10.18429/JACoW-ICALEPCS2017-TUPHA178

[7] Linux Kernel-based Virtual Machine, https://www.linux-kvm.org/

[8] libvirt Virtualization API, https://libvirt.org/

[9] Supermicro MicroCloud, https://www.supermicro.com/products/nfo/MicroCloud.cfm

[10] Cassandra PV Archiver, https://oss.aquenos.com/cassandra-pv-archiver/

[11] EPICS PV Gateway, https://epics.anl.gov/extensions/gateway/index.php

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP11

ETHERCAT SOLUTION FOR THE TPS CONTROL SYSTEM

C. Y. Liao, C. Y. Wu, Y. S. Cheng, K. H. Hu, K. T. Hsu
NSRRC, Hsinchu 30076, Taiwan

Abstract

The EtherCAT real-time Ethernet technology is now widely used in the field of industrial automation. This paper is to evaluate and establish the EtherCAT (Ethernet for Control Automation Technology) solution for digital and analogue I/O and motion control in accelerator applications. Thanks to developments at the Diamond Light Source, EtherCAT is integrated into EPICS with support for most devices of a general data type. Preliminary tests and plans are summarized in this report.

INTRODUCTION

The Taiwan Photon Source (TPS) is a low emittance, high brightness synchrotron light source, located in Hsinchu, Taiwan. New control technologies have begun to emerge in the past few years and we pay specific attention to the EtherCAT technology [1], which is a fieldbus system based on Ethernet and was developed by Beckhoff Automation [2] in 2003. It has become one of the mainstream communications interfaces for connecting to PLCs, sensors, servo motors, I/O and other automation equipment. The goal of EtherCAT development is to enable Ethernet applications to reduce data update time, lower communications jitter and reduce hardware costs for application to automation. The EtherCAT protocol is standardized in IEC 61158. The EtherCAT master sends a telegram through to each slaves and each EtherCAT device reads the data and inserts its data into the frame as it moves downstream. EtherCAT supports almost any topology, such as line, tree or star.

Several support programs were developed for communication between EPICS IOC and the EtherCAT master. At the Diamond Light Source (DLS) support is being developed based on a scanner process through the IgH EtherCAT which can transfer data between EPICS IOC and EtherCAT master [3] through the UNIX socket. Based on IgH EtherCAT, the PSI [4] also develops an EtherCAT driver, ecat2, for the same purposes. The HE Youngcheng [5] uses the OPC (Object Linking and Embedding for Process Control) gateway driver, which can operate in a Windows OS system. In this study, the DLS-EtherCAT driver is used. It can support most EtherCAT devices of a general data type. A system architecture, plans and implementation will be presented in this report.

ETHERCAT EPICS ARCHITECTURE

Platform

The platform for EtherCAT application is based on the Linux OS, the real-time patch is optional. Two Ethernet ports are basic configuration, one for control system internet and one to connect to EtherCAT devices. In our applications, CompactPCI (Advantech) [6], MXC industrial PC (Adlinktech) [7], UP Squared (Up-board) [8], Raspberry pi [9], all are acceptable. Some existing control systems for ID control are based on the cPCI platform. As a new platform, we select the MXC platform. UP2 and Raspberry are used for special standalone applications. The x86 or x64 works as the EtherCAT master platform.

IgH EtherCAT Master

The IgH EtherCAT Master is an open-source EtherCAT communication software tool [10] for the Linux platform. The devices in the EtherCAT network can be divided into master and slaves. The master acts as the controller of the entire network and is responsible for sending instant packets to each slave to read data or control its behaviour. The Linux OS kernel version must be 2.6 or 3.x. Currently the version of IgH EtherCAT modules is 1.5.2 which was released on 2013-02-12. The performance was studied by Soyeon Kim [11], which shows that the real-time performance is similar to or better than Beckhoff's TwinCAT. The Ethernet driver is divided into the native EtherCAT-Capable Ethernet drivers and the Linux standard Ethernet driver. Users must use the first driver to support real-time communication. After installation of the IgH EtherCAT master, execution of the command "ethercat version" can check if the machine has the EtherCAT driver module loaded, the command "/etc/init.d/ethercat status" checks the status of the IgH EtherCAT master, and the command "/etc/init.d/ethercat stop(start or restart)" controls the behaviour of the IgH EtherCAT master.

DLS EtherCAT EPICS Support

The Diamond Light Source (DLS) EtherCAT is an EPICS support module to interface an EtherCAT bus to EPICS. It uses a scanner process that serves as a server to communicate with EPICS IOC. The PDO (Process Data Objects) and SDO (Service Data Objects) access are supported. The module along with Diamond's Remote I/O was presented in [3]. An EtherCAT device with "REAL" data type of PDO is not supported. We hope to be able to support it in the future. The DLS EtherCAT support consists of four components:

1. Real-time Linux kernel (Optional)
2. IgH EtherCAT device driver
3. Scanner
4. IOCs.

The EPICS environment with asynDriver (asynchronous driver support) and Sequencer (snc/seq) modules are needed for running the DLS EPICS IOC.

Start-up

After the slaves (EtherCAT devices) are connected to the setup, the EtherCAT command line tool can help to visualize it. First, the platform must find the EtherCAT devices

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP11

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

via the command "ethercat slaves". For example, the TPS IU22 insertion device has EtherCAT devices and this command will list all its devices as shown in Fig. 1. Then, using this information, the DLS-EtherCAT chain.xml file is created to configure the sequence and device names. Compiling (Make) the chain.xml file generates the scanner.xml file and operation of the scanner process with the scanner.xml. The scanner process will open up a UNIX socket waiting for communication with the EPICS IOC, thus starting the IOC process to create the PVs.

```
172.20.3.17:~ # ethercat slaves
 0  0:0   OP  +  EK1100 EtherCAT Coupler (2A E-Bus)
 1  0:1   OP  +  EL1819 16K. Dig. Eingang 24V, 10?s
 2  0:2   OP  +  EL2819 16K. Dig. Ausgang 24V, 0,5A, Diagnose
 3  0:3   OP  +  EL3104 4K. Ana. Eingang +/-10V Diff.
 4  0:4   OP  +  EK1100 EtherCAT Coupler (2A E-Bus)
 5  0:5   OP  +  EL3104 4K. Ana. Eingang +/-10V Diff.
 6  0:6   OP  +  EL3104 4K. Ana. Eingang +/-10V Diff.
 7  0:7   OP  +  EL4134 4K. Ana. Ausgang -10/+10V. 16bit
 8  0:8   OP  +  EL9505 Netzteilklemme 5V
 9  0:9   OP  +  EL1124 4K. Dig. Eingang 5V, 10?s, Sensorversorgung
10  0:10  OP  +  EL2124 4K. Dig. Ausgang 5V, 20mA
11  0:11  OP  +  EK1100 EtherCAT Coupler (2A E-Bus)
12  0:12  OP  +  EL3318 8K. Ana. Eingang Thermoelement (TC)
13  0:13  OP  +  EL3318 8K. Ana. Eingang Thermoelement (TC)
14  0:14  OP  +  EL3318 8K. Ana. Eingang Thermoelement (TC)
172.20.3.17:~ #
```

Figure 1: List of EtherCAT devices installed for the IU22 insertion device.

APPLICATIONS

The DLS-EtherCAT EPICS supports several EtherCAT devices with the general data type. Some of the devices are testing, such as Beckhoff, ADLINK, MOONS, and Taiwan Pulse Motion EtherCAT devices. A wide range of applications can be considered for the accelerator, such as insertion device control, power supply, interlock system, feedback control et al and many devices, including DIO and AIO control, temperature monitor, SSI encoder monitor, and et al are employed.

Insertion Device Control System

For insertion devices, EtherCAT modules are used such as digital and analogue I/O for correction power supply control, gauge pressure reading and temperature monitoring. Figure 2 shows the IU22 insertion device control system layout. EtherCAT devices used are shown in Table 1. The MXC-6300 is used as a control platform with Linux kernel 3.10.0 (rt56) CentOS 7, EPICS base 3.14.12.7, and IgH EtherCAT master 1.5.2.

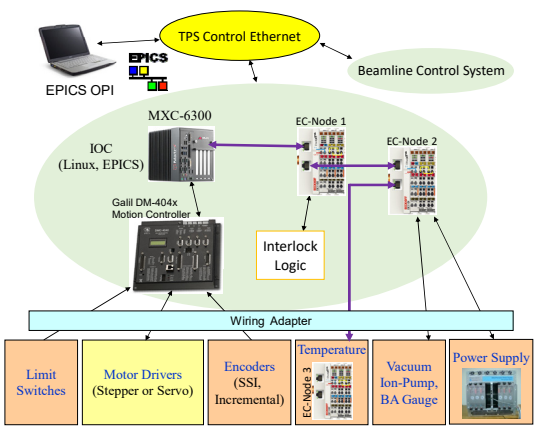


Figure 2: IU22 insertion device control system layout.

Table 1: List of EtherCAT Devices used for the IU22 Insertion Device Control System

	Device	Function
EC-Node 1	EK1100	EtherCAT Coupler
	EL1819	DI for limit switch and subsystem status input
	EL2819	DO for alarm reset and brake open
	EL3104	AI for driver signals monitor
EC-Node 2	EK1100	EtherCAT Coupler
	EL3104	AI for coil power supply monitor
	EL3104	AI for gauge pressure monitor
	EL4134	AO for coil power supply control
	EL9505	Power Supply terminal 5V
	EL1124	DI 5V for coil power supply fault
	EL2124	DO 5V for coil power supply reset
EC-Node 3	EK1100	EtherCAT Coupler
	EL3318	Thermocouple, Ch01-08
	EL3318	Thermocouple, Ch09-16
	EL3318	Thermocouple, Ch17-24

ID Encoder Malfunction Protection

Insertion devices (ID) are essential components in third-generation synchrotron light sources. Reliable operation of insertion devices is important to users of the beamlines. The most unpredictable fault is due to a soft error of absolute optical encoders due to irradiation. There are several solutions to avoid this kind of fault, one is to increase the distance of the encoder from the beam level, the other is a lead cover shielding and the final method is to adopt auxiliary position sensing devices to help recovery from a fault.

The Beckhoff EL3255 (potentiometer reader) module and potentiometer (GEFRAN, PZ-12-S-200 for Gap, PZ-12-S-075 for Phase) are installed in the ID as shown in Fig. 3. After calibration of the potentiometer reading with SSI encoders, the auxiliary absolute position information can be used to cross check the absolute SSI encoder health for malfunction protection purpose. A protection process is developed to compare the position from two sensors, if the difference is too large, the abort motion command will be sent to the controller to stop the motor driver.

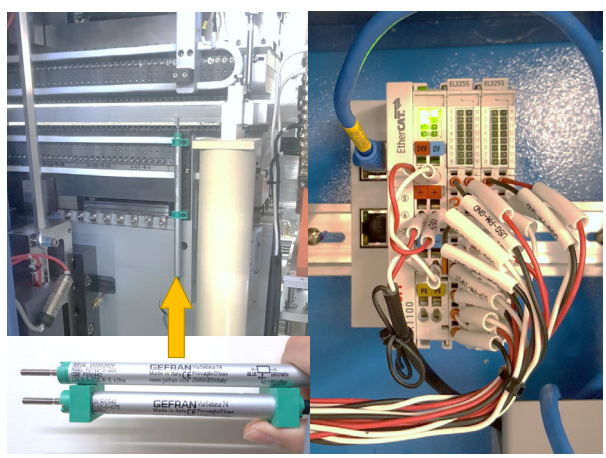


Figure 3: Potentiometer and EtherCAT devices for ID.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP11

SUMMARY

The EtherCAT real-time Ethernet technology is now widely used in the field of industrial automation control. The EtherCAT hardware is introduced into the TPS acceleration control system due to its simple configuration and easy extension. The Diamond Light Source EtherCAT EPICS support is selected for communication with the EPICS environment. It's easy to use and works well with most EtherCAT devices for digital and analog I/O, temperature monitoring, encoder monitoring, and et al. Currently, the new TPS insertion device control system uses EtherCAT devices. Another application in the TPS is to develop a protection process for the insertion device when the optical encoder fails. Thanks to the short cycle time of the EtherCAT protocol, the protection cycle can reach 1 kHz. The features of the EtherCAT in the TPS use the EtherCAT motion controller for motor control applications and adoption of the EtherCAT based data acquisition (DAQ) modules for signal acquisitions are in planning.

ACKNOWLEDGMENTS

The authors are greatly thankful to those who have developed the EtherCAT EPICS support for reading EtherCAT based hardware. And thanks to Ronaldo Mercado (Diamond Light Source) for assistance in resolving the issue of reading/writing the SDO address for this study.

REFERENCES

[1] EtherCAT Technical Introduction and Overview, www.controldesign.com/assets/Media/MediaManager/EtherCAT_Introduction.pdf

[2] Beckhoff Automation, www.beckhoff.com

[3] R. Mercado et al., "Integration EtherCAT Based IO into EPICS at Diamond", in Proc. *ICALEPCS'11*, Grenoble, France, Jul. 2011, paper WEMAU004.

[4] Dragutin Maier-Manojlovic, "Real-time EtherCAT Driver for EPICS and Embedded Linux at Paul Scherrer Institute (PSI)", in Proc. *ICALEPCS'15*, Melbourne, Australia, Jul. 2015. paper MOPGF027.

[5] HE Yongcheng et al., "A method of communication between EPICS IOC and EtherCAT devices", Nuclear Techniques, Vol. 37, No. 11, November 2014.

[6] Advanteck cPCI platform, www.advantech.tw

[7] Adlinktech, MXC platform, www.adlinktech.com

[8] Up-board, Up Squared platform, www.up-board.org/up-squared

[9] Raspberry pi platform, www.raspberrypi.org

[10] Etherlab IgH EtherCAT, www.etherlab.org/en/ethercat

[11] Kim, Soyeon and Euncheol Shin, "A Performance Evaluation of Open Source-based EtherCAT Master Systems", Proceedings of the 4th International Conference of Control, Dynamic Systems, and Robotics (CDSR'17), page 128, Toronto, Canada.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP13

CONTROL SYSTEM UPGRADE FOR THE FFAG ACCELERATOR COMPLEX AT KURNS

Y. Kuriyama*, Y. Fuwa, Y. Ishi, Y. Mori, H. Okita, T. Uesugi
Institute for Integrated Radiation and Nuclear Science, Kyoto University, Osaka, Japan

Abstract

Fixed field alternating gradient (FFAG) accelerator complex has been operated as a proton driver for the experiment of accelerator driven system (ADS) at Institute for Integrated Radiation and Nuclear Science, Kyoto University (KURNS). PLC based control system has been developed and the operator interface has been connected to PLC via network. Originally, a LabVIEW based operational interface was chosen to construct the system because of its easiness. However we met an upgrade problem, and a new control system based on EPICS instead of LabVIEW was introduced in 2010. In the spring of 2018, the replacement from LabVIEW to EPICS has been almost completed except for the beam interlock system and the LINAC control system provided by LINAC production company (AccSys). Also, the EPICS archiving tool (Archiver Appliance) has been invoked and operated at the end of 2017. This presentation reports the details of the current control system and also the upgraded GPIB control and storage system.

INTRODUCTION

FFAG accelerator complex has been operated as a proton driver for the ADS experiment [1]. The construction of this complex had been started since the fiscal year of 2002 and with this complex, the world first ADS experiment was carried out in March, 2009 [2].

The KUNRS complex consists with two ion sources, H^- linac and four FFAG rings. Figure 1 shows the schematic view of KURNS complex.

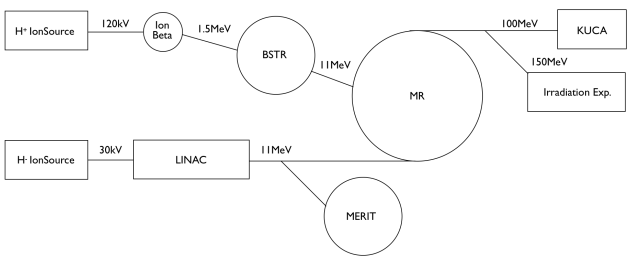


Figure 1: Schematic View of the KURNS Complex.

To control this complex, PLC based control system had been developed. Originally, operator interface had been developed based on a LabVIEW [3]. Since 2010, the aim for the sophistication and stability, we introduced EPICS control system [4]. The immigration from LabVIEW to EPICS had been almost completed in the spring of 2018 except for the beam interlock system and Linac control system.

* kuriyama@rri.kyoto-u.ac.jp

OVERVIEW OF CONTROL SYSTEM

In the KURNS complex, PLC-based systems have been adopted since the beginning of construction. The operator communicates with the PLC via the network, and the control target device such as the electromagnet power supply is operated by receiving a command from the PLC.

About 20 PLC-CPU modules and 20 Linux PCs, 5 Windows PCs and 2 board computers and 2 servers are used. Also, 20~35 network cameras are used for an equipment monitoring and beam profile observation.

These control devices are connected by an independent network for the accelerator control. Figure 2 shows a diagram of control system.

Since all the devices are connected to the same network, high network load is becoming a problem.

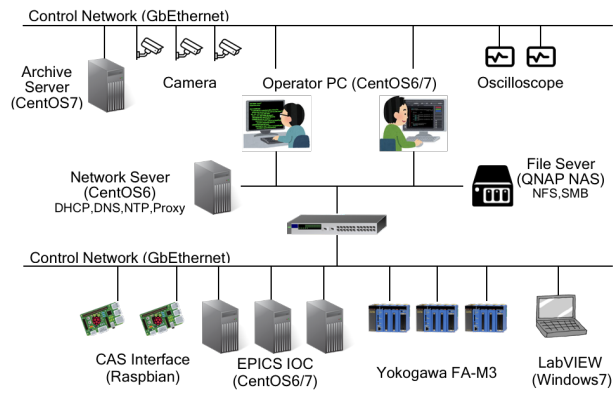


Figure 2: Schematic View of the Control System.

For the device control system, EPICS-based ones and GPIB controlled ones are mixed. EPICS-IOC consists of Yokogawa's PLC FA-M3 and CentOS6 or CentOS7 PC. The number of EPICS-IOC is about 15.

From the point of necessary processing capacity, it is possible to aggregate them into several units. Since accelerator components are often replaced at the KUNRS complex, EPICS IOCs are prepared for each section such as ring and beam line to cope with this. It realizes the flexibility of control system.

RENEWAL OF STORAGE SYSTEM

During the past 10 years, KURNS control system had changed file server three times. One is Linux-based file server and two are mac os based file servers.

Since the period of use was exceeded five years in December 2017, we began to consider updating the file server. Requirements for the new file server are

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP13

- Supported Protocol : NFS v3, SMB 2.0,
- Redundancy : RAID 6,
- Backup : External Storage, Snapshot,
- Ease of replacement : Hot-swappable, General Drive Bay (SATA).

In order to satisfy the above requirements, we decided to shift to a NAS dedicated machine instead of a x86 based PC.

The adopted hardwear is TS-431P2-4G manufactured by QNAP [5]. It has Arm-based CPU, 4 GB memory, and 4 storage bays. The specifications of new file server are summarized in the Table 1.

Table 1: Specifications of the New File Server

Model	QNAP TS-431P2-4G
OS	QTS 4.3.4
CPU	1.7 GHz Quad Core (Arm Cortex-A15)
Memory	4 GB (DDR3) × 1
Storage	3 TB × 4 (RAID6)
LAN	GbE × 2 (Link Aggregation)

We continue to use previous file server as read-only storage for some time and evaluate the durability and reliability of the new file server.

INTRODUCTION OF EPICS ARCHIVING SYSTEM

Until now, we used homemade shell scripts to preserve various parameters. The timing of preservation is left to the judgment of the operator. Therefore, it was not possible to save parameters during non-operation.

In order to deal with this problem, we started experimental use of the EPICS Archiver Appliance [6] from December 2017. Approximately 140 PVs were recorded at the shortest period of 0.1 second. The ArchiveViewer [7] is used to display the acquired data. Figure 3 shows vacuum data taken from January 1, 2018 to February 1, 2018.

Figure 3: Vacuum Data from 1 Jan. 2018 to 1 Feb. 2018 taken by "The EPICS Archiver Appliance". The Viewer is "ArchiveViewer".

Since the EPICS Archiver Appliance was stable over 3 months, we started to use full-scale usage from April 2018. We prepared exclusive PC and EPICS archive server had started to record about 310 PVs. The hardware specification of the EPICS archive server are summarized in the Table 2 and software specifications are summarized in the Table 3.

Table 2: Hardware Specifications of EPICS Archive Server

CPU	4 GHz Quad Core (i7 6700K) × 1
Memory	16 GB (DDR4 PC4-17000) × 4
Storage	1 TB SSD × 1
	6 TB NAS (RAID6)

Table 3: Software Version of EPICS Archive Server

OS	CentOS7
Archiver Appliance	22 June 2017
Java	Oracle Java SE Dev. Kit 8
Tomcat	7.0.86
Apache	2.4

The data acquired by this archiver is directly stored in the NAS. Currently, the event rate is 65/s and the data consumption is 0.1 GB/Day. Since there is sufficient margin in performance, we plan to increase the number of PVs to store.

MIGRATION OF GPIB CONTROL ENVIRONMENT

In the KURNS control system, GPIB is used to control old type power supply and stepping motor controller. Control programs for the GPIB devices were made using the Glade [8] version 2 for GUI creation under the CentOS5 environment.

As the CentOS5 support period expired in March 2017, the immigration of the GPIB control environment became necessary.

In the KUNRS control system, GPIB-USB-HS [9] manufactured by National Instruments has been used as a standard device for connection between the device controlled by the GPIB and the PC. But the device driver of GPIB-USB-HS provided by manufactured is only compatible with CentOS5 or older.

In order to use the CentOS6 or later version, it was necessary to change the device driver. Therefore, we decided to use the device driver Linux GPIB Package [10] which is being developed as open source. In addition, since this device driver was able to cooperate with Python, we decided to use Tkinter as a GUI creation tools, which is Python's standard library.

Before the migration, when relocating programs between PCs, there was a need to recompile. After migration, recompile doesn't need and portability improved with Python. Table 4 shows the results before and after the immigration of the GPIB control environment.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

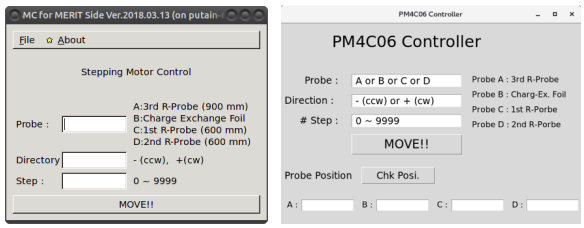
ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP13

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Table 4: Migration of GPIB Control Environment

	Before Migration	After Migration
Hardware	NI GPIB-USB-HS	NI GPIB-USB-HS NI GPIB-USB-HS+
OS	CentOS5	CentOS7
Dev. Driver	NI-488.2	Linux GPIB Package
App. Lang.	C (GTK)	Python (Tkinter)

Figure 4 shows two operator interfaces of stepping motor controller, one is created using GTK and another is using Python(Tkinter).



(a) GTK based OPI. (b) Python (Tkinter) based OPI.

Figure 4: Examples of Operator Interface for the device controlled via GPIB.

NEXT PLANS

Integration of GPIB Control to EPICS

GPIB controlled devices are independent control, not in cooperation with EPICS. For this reason, different operation systems are mixed in control, which is a cause of complicating the operation of the accelerator. In order to solve this problem, we are preparing a bridge program including GPIB control which behaves as EPICS IOC.

Integration of Beam Diagnosis to EPICS

Since problems arise in the accessibility of data acquired by the beam diagnosis device, we would like to implement bulk management using EPICS archive server. We are developing data acquisition system integrated with EPICS.

Update of Linac Control System

We introduced a linear accelerator to the KURNS complex system in 2008. It has been over 10 years since the introduction, and in 2015, we had experienced serious hardware problems with RFQ.

The PC used for control has not been updated since its introduction and failed at startup and sudden shutdown occurred. Although early replacement is necessary, there is no source code of control software in the manufacture. And there is no device driver of the input/output board (for communication with the linear accelerator power supply unit) for the latest OS.

Since there are many uncertain factors regarding the updating work of the control PC, the manufacture cannot estimate the cost of renewal. However, for the continuous operation of the KURNS complex, the replacement of linac control system cannot be avoid.

SUMMARY

The KURNS complex has also exceeded 10 years since the start of construction. During this 10 years, the structure of the accelerator continues to change, and the control system is also changing to cope with the change.

For the purpose of the sophistication and stability, replacement of LabVIEW-based control system to EPICS is almost completed and for the data protection, file servers were replaced fourth times and EPICS archive server is introduced. And beam diagnosis system and GPIB control system are also moving to EPICS environment.

At present, the update plan of the control system of the linear accelerator has not been determined but this is a most urgent task of control group.

REFERENCES

[1] T. Uesugi *et al.*, "FFAGs for the ERIT and ADS Projects at KURRI", in *Proc. EPAC'08*, Geneva, Switzerland, Jun. 2008, paper TUOBM04.

[2] C. H. Pyeon *et al.*, "First Injection of Spallation Neutrons Generated by High-Energy Protons into the Kyoto University Critical Assembly", *J. Nucl. Scie. Technol.*, vol. 46, no. 12, pp. 1091–1093, 2010. doi:10.1080/18811248.2009.9711620

[3] M. Tanigaki *et al.*, "Control system for the FFAG complex at KURRI", *NIM*. A, vol. 612, pp. 354-359, 2010. doi:10.1016/j.nima.2009.11.024

[4] Y. Kuriyama *et al.*, "EPICS Control System for the FFAG Complex at KURRI", in *Proc. ICALEPCS'13*, San Francisco, CA, USA, Oct. 2013, paper THPPC036.

[5] QNAP, https://www.qnap.com

[6] The EPICS Archiver Appliance, https://slacmshankar.github.io/epicsarchiver_docs/index.html

[7] The ArchiveViewer, https://slacmshankar.github.io/epicsarchiver_docs/archiveviewer.html

[8] Glade, https://glade.gnome.org

[9] GPIB-USB-HS, http://www.ni.com/en-us/support/model.gpib-usb-hs.html

[10] Linux GPIB Package, https://linux-gpib.sourceforge.io

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP14

VACUUM CONTROL SYSTEM FOR THE TAIWAN PHOTON SOURCE

Y-C. Yang, J-Y. Chuang, Y-M. Hsiao, Y. Z. Lin, C. K. Chan and C. C. Chang
National Synchrotron Radiation Research Center (NSRRC), Hsinchu 30076, Taiwan

Abstract

The Taiwan Photon Source (TPS) is a 3 GeV storage ring. A NI C-RIO controller, basic to the real-time, EPICS program, is used and designed such as to maintain ultra-high vacuum conditions and protect vacuum components. Pressure readings from ionization gauges are taken as the logic signal to control sector gate valves to protect ultra-high vacuum condition. Monitoring of vacuum components, water-cooling systems and chamber temperatures serves to protect vacuum equipment from radiation power. The evolution and status of the control system is presented in this paper.

INTRODUCTION

Commissioning for the TPS, a low-emittance 3-GeV synchrotron ring, started in December 2014 and is now currently operated in top-up mode at 400mA for users. During past year's operation, the design goal of 500mA beam current was archived on December 2015. Until the last machine shut down in June 2018, a total beam dose of 3631 Ah was accumulated and the beam cleaning effect decreased to 1.43×10^{-11} Pa/mA. Table 1 lists some operational milestones of the TPS in past years.

Table 1: Milestones for the TPS Currently in Operation

Date	Milestone
2014.12	first beam stored
2015.03	100mA beam current archived
2015.12	500mA design goal archived
2016.09	open to user (300mA)
2016.11	1000 Ah beam dose accumulated
2017.11	400mA operational Beam Current

The TPS storage and booster rings are located in the same tunnel. The storage ring with a circumference of 518.4m, is divided into 24 sections, including 24 bending and 24 straight sections. Each section corresponds to one control instrument area (CIA). Figure 1 shows a schematic 3D drawing of the TPS vacuum system with the storage ring (SR), booster ring (BR) and control-instrument area (CIA). The vacuum component connection cable lengths between tunnel and CIA are 20 to 30 meter long.

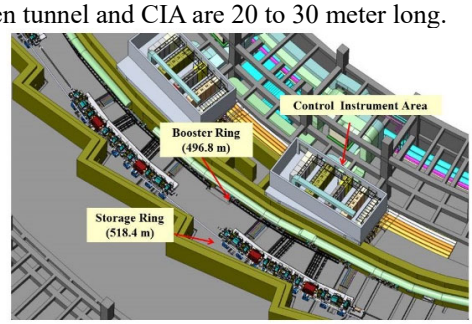


Figure 1: Schematic 3D drawing of the TPS vacuum system.

Figure 2 illustrates the layout for 1/24 of the TPS vacuum system, consisting each of a straight and a bending section with two sector gate valves (SGV), two pumping gate valves (PGV), two front-end valves (FEV), six metal angle valves (MGV), six ionization gauges (IG), ten non-evaporable getter (NEG) pumps, six sputtering ion pumps (IP), and eight turbo-molecular pumps (TMP). The vacuum chambers inside the cell contain two straight ducts S3, S4, and two bending chambers B1, B2; the S1 and S2 ducts are located at both ends of the cell isolated with two SGVs [1].

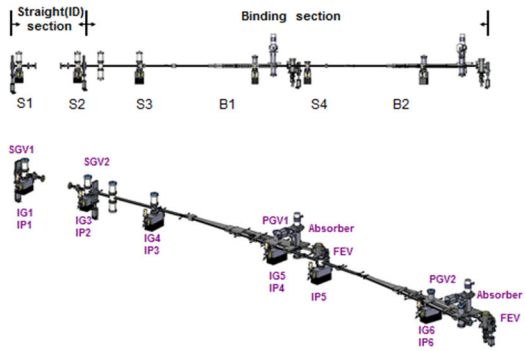


Figure 2: Layout of 1/24 vacuum section.

The mechanism of the vacuum control system is to maintain and protect ultra-high vacuum conditions by controlling and monitoring vacuum components as described above. The safety interlock system is based on the conditions in vacuum components, such as a gauge, ion pump or valve. The following sections describe the design concepts.

VACUUM CONTROL SYSTEM

TPS uses EPICS (Experimental Physics and Industrial Control System) to control and monitor the accelerator machine. EPICS can provide a standard client-server model for a distributed system. In the TPS vacuum control system, a Compact-RIO real-time controller from National Instrument® serves for the vacuum safety interlock, data acquisition and monitoring systems. Between the C-RIO and vacuum system, the interface of I/O communication is used by the I/O connect port of the vacuum controllers , such as vacuum gauges, pumps and meters for cooling water, or directly by the I/O terminals of the vacuum components. The architecture of the control and communications relations is shown in Fig. 3.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP14

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

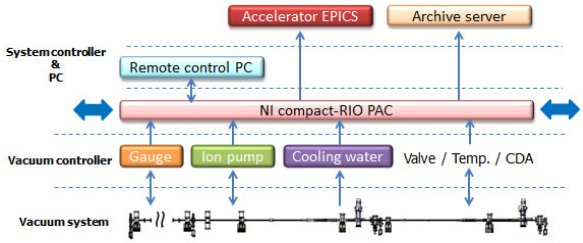


Figure 3: Architecture of the vacuum control and communications system.

Since the vacuum system of the storage ring is divided into 24 sections, with 24 C-RIO controllers distributed into 24 CIA associated with 24 vacuum sections. Each C-RIO collects all signals from one section, including about 48 analogue input signals, 96 digital input signals and 64 digital output signals. Among the analogue signals, the vacuum gauge and pump pressure readings are taken as the basic logic signal for the safety interlock system. The RTD temperature sensor readings serve to monitor the cooling water and vacuum components without special cooling system such as valves, bellows and BPM blocks. Digital input and output signals provide the status , set-point, logic trigger and remote control of the vacuum components [2]. A model of the 9074 C-RIO controller with three analogue input modules, three digital input and two digital output modules installed in CIA is shown in Fig. 4.

Figure 4: A model of the C-RIO controller with modules.

In addition to the storage ring, vacuum signals of the booster ring (BR), the transfer lines between linac and booster ring (LTB) and between booster ring and storage ring (BTS) are connected to adjacent C-RIO controllers, depending on the location, for safety interlock, data acquisition and monitors. Figure 5 shows the number of I/O ports for each vacuum subsystem.

Sub-system	AI		DI	DO
	reading	RTD	status / trigger	control
LTB	3	N/A	22	13
BR	108	N/A	264	192
BTS	6	N/A	32	27
SR	144	360	1152	1008
UTILITY	168	168	480	N/A

Figure 5: I/O ports for the vacuum subsystem.

SAFETY INTERLOCK SYSTEM

The protection of the ultra-high vacuum condition and of vacuum components is the main concern for the safety interlock system. Before the vacuum control system is designed, the properties of the vacuum components must be considered, especially the safety interlock system. Some considerations follow.

(1) Vacuum-gauge protection: In the storage ring of the TPS, an ionization gauge was chosen and taken as the source for a logical signal. During machine operation, the readings of a vacuum gauge may increase to more than 10^{-6} Pa. To avoid vacuum gauges operating in conditions of poor vacuum for an extended period, which would decrease their lifetime, a self-protection mode is set. In this mode, vacuum gauges become automatically switched off when the vacuum pressure increases suddenly to more than 1×10^{-3} Pa, but can be switched on only manually after the pressure recovers.

(2) Ion-pump protection: Similar to the vacuum gauges, ion pumps are switched on only when the local pressure is less than 1×10^{-4} Pa according to the logical output signal of the vacuum gauges. A protection mode of an ion-pump controller is concurrently selected. In this mode, the controller limits the output current and switches off the high voltage when the output current reaches or exceeds a threshold current by more than 0.2 s.

(3) Isolation valve: The mechanism of the safety interlock system is set to control the opening and closing of the sector gate valves (SGV) to isolate a vacuum system with poor pressure. When the pressure increases to more than 1×10^{-4} Pa, which is the trigger output of the vacuum gauges at either end of the valve, the SGV closes to protect the vacuum at the other side. Two properties of the SGV must be considered here: the pressure of compressed air and the closing time. The pressure of compressed air for normal operation is 4~8 bar (4.08~8.16 kg/cm^2). A trigger point of 5 kg/cm^2 for the compressed air is therefore set as the interlock trigger signal for the utility system. The closing interval of the SGV is 4 s, based on sufficient pressure and rate of flow of compressed air to fill the cylinder. To ensure normal operation of the SGVs, independent air piping for each SGV is necessary. If one air pipe supplies more than one SGV, the closing time of the SGVs is delayed, thus affecting the performance of the SGVs.

LOGIC DESIGN

Figure 6 shows the logic diagram of the SGV control mechanism, which complies with the principle of manual on and fail-safety. When the pressure on both sides of an SGV is satisfactory, the SGV can be controlled and switched off automatically as soon as the interlock at either end is triggered. All three gauges are installed between two SGVs. If any two gauges are over the set point or

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP14

malfunctioning, the logic trigger becomes active and the SGVs are then closed. Two front- ends (FE) associated with this vacuum system additionally ensure the completeness of the safety interlock system. Besides, considering the vacuum pressure, the emergency trip for neighbouring valves is added to the interlock system to decrease the risk of a spread of poor vacuum. When any neighbouring valve is out of control or malfunctions, the emergency trip signal becomes active and the SGVs close to prevent the spread of poor vacuum.

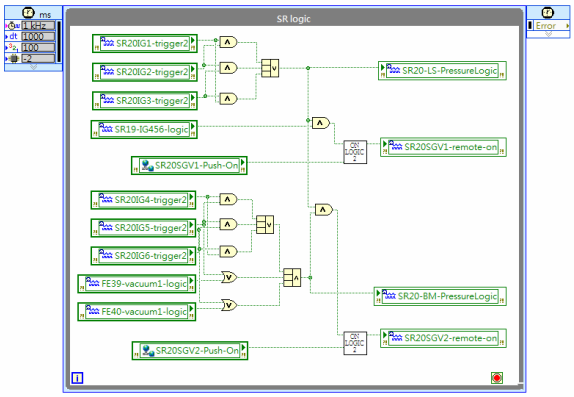


Figure 6: Diagram of the SGV logic control system.

Besides the basic protection of ultra-high vacuum condition, another issue is about vacuum component protection from synchrotron radiation power. The open signal of an SGV and normal status of the cooling water are summed and sent to the machine protection system (MPS) to make sure the vacuum system is ready for machine operation.

During the past years of operation, the interlock logic was optimized. The biggest difference was to avoid a machine trip by a single signal source. Several times during machine operation, false FE vacuum signals caused interrupts from faulty evaluation of signals from FEs and BLs. A vacuum signal in the storage ring adjacent to the FE system was added and a trip signal is given only when these two signals fault simultaneously. The temperature interlock system was also corrected. Countdown and reset functions were added to avoid a trip signal being sent by the signal from a damaged temperature sensor [3].

MONITOR SYSTEM

NI Labview is a system engineering software, which offers a graphical programming approach that visualizes the application. The graphical interface is flexible and easy to use. Web publishing functions in Labview are widely used in the TPS vacuum monitoring system, which transfer Labview front panel pages to web pages easily. Figure 7 displays one example for the monitoring page of the vacuum status, which includes machine operation conditions and insertion devices status acessed from the TPS EPICS IO server and vacuum pressure signal binding from 24 C-RIO controllers. It is easy to be viewed by vacuum group staff on PC or cell phone.

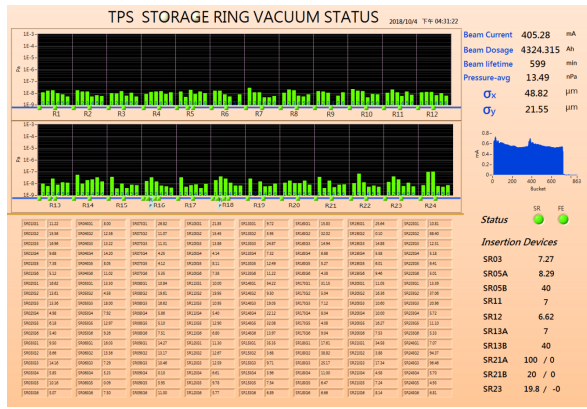


Figure 7: Vacuum monitoring page of the TPS storage ring.

One fake event occurred originating from the Front End (FE) interlock system, which also uses a C-RIO 9074 controller [4]. The reason for the event was not clear, but experience indicats that it was easy occured after long time operation without rebooting or execution of the distributed system manager (DSM) function of Labview. The CRIO 9074 VxWorks OS controllers were replaced to Linux OS versions, since the C-RIO 903X Linux OS controller exhibits a more stable performance, the CPU loading is five times lower than for the 9074 VxWorks.

CONCLUSIONS

The design of the TPS vacuum control system is described above. The vacuum pressure protection function and component protection logics worked well during the past years of operation. All vacuum signals, including vacuum pressure, status of vacuum devices, temperature distribution were displayed on web pages which is easy to monitor. More applications like LINE Notify instant message sending function will be developed in the future.

REFERENCES

[1] G. Y. Hsiung, *et al.*, "TPS Vacuum system", in *Proc. PAC'09*, Vancouver, Canada, May 4-8, 2009, paper MO6RFP018.

[2] Y. C. Yang, *et al.*, "Development of the TPS Vacuum Interlock and Monitor System", in *Proc. IPAC`14*, Dresden, Germany, June 15-20, 2014, paper WEPME051, pp. 2387-2389.

[3] Y. C. Yang, *et al.*, "Two year operational experience with the TPS vacuum system", Journal of physics: conf. Series 874(2017).

[4] J. C. Chuang, *et al.*, "Upgrade for the TPS Fail Safe Front End Interlock System", 9th International Workshop on Radiation Safety at Synchrotron Radiation Sources, Hsinchu, Taiwan, April 19-22, 2017.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP15

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

RECENT DEVELOPMENT OF THE RIKEN RI BEAM FACTORY CONTROL SYSTEM

M. Komiyama, M. Fujimaki, N. Fukunishi, K. Kumagai, A. Uchiyama
RIKEN Nishina Center, Wako, Saitama, Japan

Abstract

We report on the development of the successor to the existing controller devices used for the magnet power supplies in the RIKEN Radioactive Isotope Beam Factory (RIBF). The existing system controlling the magnet power supplies is operated on the Versa Module Europa (VME) computing machines, under the Experimental Physics and Industrial Control System (EPICS) framework. The present controller system has been operated stably for over 10 years. However, it is now commercially unavailable, because the supply of some parts has already ceased. From 2011 to 2016, we have been developing a successor system to achieve essentially the same function as the existing one, but the successor system is designed to run in control systems constructed by programmable logic controller (PLC) modules instead of the VME computing environment, in order to achieve a cost reduction and easily cooperate with other systems.

We set up a test system using this successor and confirmed that a magnet power supply could be controlled in the same manner as the existing system. Now, we plan to begin controlling magnets of beam transport lines using this successor system in the current year.

INTRODUCTION

The RIKEN Radioactive Isotope Beam Factory (RIBF) is a cyclotron-based accelerator facility aiming at the development of nuclear physics, materials, and life science studies. RIBF consists of two heavy-ion linear accelerator injectors and five heavy-ion cyclotrons, including the world's first superconducting ring cyclotron (SRC). Cascades of the cyclotrons can provide the world's most intense RI beams over the whole atomic mass range, using fragmentation or fission of high-energy heavy-ion beams [1]. For example, a 345-MeV/nucleon ^{238}U beam of 70 pnA was successfully extracted from the SRC in 2017. RIBF was constructed as an extension of the old facility commissioned in 1986, by adding three new cyclotrons, and began operation in 2006.

The components of the RIBF accelerator complex, such as the magnet power supplies, beam diagnostic devices, and vacuum systems, are controlled by the Experimental Physics and Industrial Control System (EPICS) [2], with a few exceptions such as the control system dedicated to the radio frequency system of RIBF [3]. However, all the essential operation datasets of EPICS and other control systems are integrated into the EPICS-based control system. In addition, two types of interlock systems that are independent of the accelerator control systems are also operated in the RIBF facility: a radiation safety interlock system for human protection [4] and a beam interlock system (BIS) that protects the hardware of the RIBF accelerator complex from potential damage caused by high-power heavy-ion beams [5].

UPDATE OF CONTROLLER DEVICE FOR MAGNET POWER SUPPLY

Controller Device for Magnet Power Supplies in RIBF

The magnet power supplies are operated both in the old facility section and in the newly added facility section commissioned since 2006 (hereafter, the new facility section). However, there are differences in these controllers according to their introduction times. The magnet power supplies in the old facility section are controlled by our in-house controller device based on Computer-Aided Measurement And Control (CAMAC), a device interface module (DIM) [6]. On the other hand, the magnet power supplies in the new facility section are designed to be controlled by the Network I/O (NIO) system, which is a commercially available control system manufactured by the Hitachi Zosen Corporation. A block diagram of the NIO system is presented in Fig. 1.

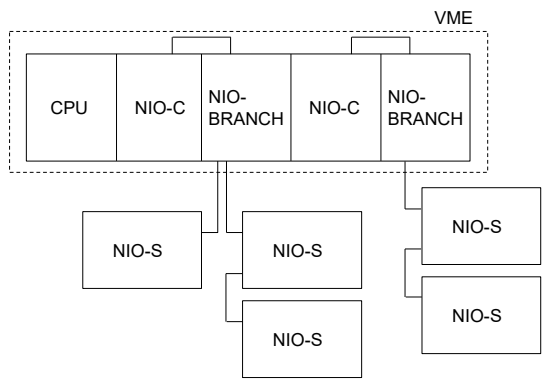


Figure 1: Block diagram of the NIO system under operation.

The NIO system consists of three types of controllers: the NIO-S board, NIO-C board, and branch board. The NIO-S board is a slave board attached directly to the magnet power supply, which controls it according to a signal sent from an upper-level control system through the NIO-C board. The NIO-C board acts as a master board of the NIO-S boards and is designed to operate in the Versa Module Europa (VME) computing machines. The NIO-C board not only has an NIO board, but also an additional board dedicated to High level Data Link

* misaki@riken.jp

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP15

Control procedure (HDLC) communication. The NIO-C and NIO-S boards are connected using an optical fiber cable through a branch board. One NIO-C board can control 43 NIO-S boards via branch boards, and there are 15 NIO-C boards in operation connected to seven VME chassis distributed in RIBF. Because one NIO-S board can only control one magnet power supply, there are approximately 500 NIO-S boards in the RIBF accelerator complex, and this corresponds to 60% of the total magnet power supplies used in the RIBF accelerator complex.

Development and Update to the Successor

DIM has been working stably for over 30 years, with improvements being added as required. However, in addition to its serious aging, it has become difficult to maintain DIM because there are no engineers who can produce or repair it. Considering this situation, we are systematically replacing the old magnet power supplies controlled by DIM to new ones controlled by the NIO. As a result, the number of magnet power supplies controlled by DIM was reduced from over 680 to 375 in 2017. In this situation, the production of NIO was terminated, and we can no longer purchase NIO boards. Therefore, we needed to develop a successor, and we planned to develop this in the order of an NIO-S board, NIO-C board, and branch board. We developed an NIO-S successor between 2011 and 2014. Table 1 lists the hardware specifications of the NIO-S under operation and its newly developed successor.

Table 1: Specifications of NIO-S Under Operation and its Successor

NIO-S	Under operation	Successor
CPU	SH-2 28.33 MHz	SH-2 SH7084 32 MHz
EPROM	128 Kbyte	-
Flash ROM	256 Kbyte	32 Mbyte
SRAM	512 Kbyte	512 Kbyte
SCA	HD64570	TD-HDLCip
Digital input (points)	32 (8 points/ 1 common)	48 (8 points/ 1 common)
Digital output (points)	32 (8 points/ 1 common)	32 (8 points/ 1 common)
Serial interface	RS-485 Optical link	RS-485 Optical link
Debug port	RS-232C	RS-232C

This successor was designed to be compatible with the NIO-S board currently under operation. Communication is the most significant feature of the successor. The HDLC transmission method, which is performed using an adaptor chip used for serial communication in the NIO-S under operation, is replaced in the successor by using the IP in the field-programmable gate array (FPGA). We also increased the number of digital inputs from 32 to 48, in consideration of future expandability.

After the development, we updated the existing NIO-S installed in the three independent magnet power supplies to the successor and began testing its operation from 2014. We can control these as smoothly as other magnet power supplies without modifying anything in the existing control programs.

In the development of a successor for the NIO-C board, the required specifications are essentially the same as for the existing board. However, we decided to design a successor to run on a control system constructed by programmable logic controller (PLC) modules instead of the VME computing environment used currently, in order to achieve a cost reduction and easily cooperate with other systems. The hardware specifications of the NIO-C under operation and its newly developed successor are listed in Table 2.

Table 2: Specifications of the NIO-C Under Operation and its Successor

NIO-C	Under operation		Successor
	NIO-board	Communication-board (HDLC)	
CPU	TMPZ84 C013A-10 6.4 MHz	TMPZ84C0 13A-10 8 MHz	NIOS2 Processor 32 MHz
ROM	28 Kbyte	28 Kbyte	EPCS16 2 Mbyte
RAM	32 Kbyte	28 Kbyte	
DRAM	-	-	32 Mbyte
DPRAM	2 Kbyte	2 Kbyte	16 Kbyte (in FPGA)
SCA	HD64570		TD-HDLCip
FPGA	-		Cyclone4 EP4CE22
Bus controller	- (VME)		A6374LG (Yokogawa)
Serial interface	RS-485		RS-485
Debug port	RS-232C		RS-232C

The NIO-C successor was developed based on the PLC system manufactured by the Yokogawa Electric Corporation (hereafter, FA-M3), following recent trends in the control systems of RIBF accelerators. One of the advantages of adopting FA-M3 is that we can set up a

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP15

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

simple control system by choosing a Linux-based PLC-CPU (F3RP61 [7]) on which EPICS programs are executed. In that case, F3RP61 works not only as a device controller, but also as an EPICS Input Output controller (IOC) [8]. Figure 2 presents a block diagram of the successor system of the NIO system. We designed the NIO-C successor to communicate data and commands with a PLC-CPU via a shared RAM contained on it.

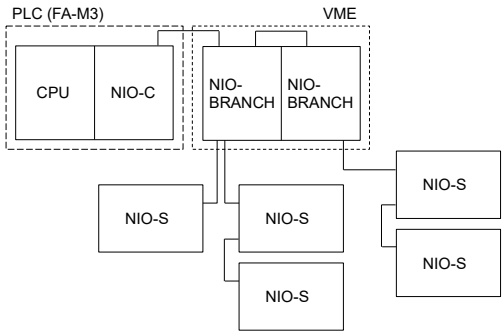


Figure 2: Block diagram of the successor system of the NIO system.

The software of the successor was developed maintaining full compatibility with the NIO-C under operation. The NIO-C successor has 13 kinds of command codes, such as magnet power supply on/off, polarity switching, and pulse output. Most of these command codes inherit those of the NIO-C under operation: there is one updated command code for setting current value to the magnet power supply. In the existing system, there are two ways of setting a current target value for a magnet power supply: one is to give only the target current value to the magnet power supply, in which a default current excitation rate is used; the other is to provide a target current value and specify its current excitation rate (hereafter ramp control). The ramp control command code in the existing system requires setting parameters such as the current target value, step time, and total setting time to reach the target current value. It is assumed that the current value is 0 A, and ramp control is performed from 0 A towards the target current value. In other words, this cannot be used when starting from other current values. Therefore, we now only use former command code in the current control of the magnet power supply. When we need to significantly change the current value, we employ a program developed by implementing ramp control starting from an arbitrary current value, in order to ensure safety and not damage the magnet power supply. Therefore, in developing the software for the NIO-C successor an upgraded command code for ramp control was developed by adding a current starting value to the parameter set with the existing ramp command. We assume that this command code is utilized in the control of a large current magnet power supply, such as for a dipole magnet.

We completed the development of the software of the NIO-C successor in 2016. Subsequently, we have started to develop the control program of the magnet power supply, using F3RP61 as an EPICS IOC. As a first step, we have developed a program that implements equivalent usability to existing control programs by using a test unit containing only F3RP61 and the NIO-C successor, as shown in Fig. 2. The test unit has been attached to one of the NIO stations in operation, which controls the magnets of beam transport lines including dipole magnets, quadrupole magnets, and steering magnets. There is one VME-CPU, five NIO-Cs, and two or three branch boards connected to each NIO-C, and approximately 150 of the magnet power supplies are controlled by this NIO station. For testing the NIO-C successor, we replaced one of the NIO-Cs in operation with the successor and connected this to an existing branch board. This NIO-C is controlling 34 magnet power supplies. As a first step, we tested all 13 command codes by developing test programs, and we confirmed that all commands successfully performed according to their specifications. Based on this result, we have developed an EPICS run-time database and graphical user interface (GUI) for the magnet control based on the existing one. As a result, we successfully controlled all 34 magnet power supplies in the same manner as the existing system. The response to the command input was the same as for the current system. Furthermore, we can control these by using an updated command for the ramp control by setting the parameters of some patterns.

As a next step, we will start the development of a successor to the branch board. The branch board is a non-intelligent board for a star connection between an NIO-C and NIO-S with an optical cable. Although the existing branch board runs in the VME chassis, a necessary input power of 5 V should be fed independently from VME. Therefore, we plan to develop a successor to the branch board with specifications of not using VME, in order to lower the costs and simplify the board.

REFERENCES

[1] O. Kamigaito *et al.*,"Present Status and Future Plan of RIKEN RI Beam Factory", in *Proc. IPAC2016*, Busan, Korea, May 2016, paper TUPMR022, pp. 1281–1283. doi:10.18429/JACoW-IPAC2016-TUPMR022

[2] EPICS, http://www.aps.anl.gov/epics/

[3] M. Komiyama *et al.*, "Recent Improvements to the RIKEN RI Beam Factory Control System", in *Proc. PCaPAC2016*, Campinas, Brazil, Oct. 2016, paper WEPOPRP011 pp. 31–34. doi.org/10.18429/JACoW-PCaPAC2016-WEPOPRP011

[4] H. Sakamoto *et al.*, "HIS: The radiation safety interlock system for the RIKEN RI Beam Factory," RIKEN Accel. Prog. Rep. vol. 37, pp. 281, 2004.

[5] M. Komiyama, M. Fujimaki, I. Yokoyama, and M. Kase, "Status of control and beam interlock system for RARF and RIBF," RIKEN Accel. Prog. Rep. vol. 39, pp. 239, 2006.

[6] K. Shimizu, T. Wada, J. Fujita, and I. Yokoyama, *in Proc. Cyclotrons'84*, pp. 392-395.

[7] FA-M3, http://www.FA-M3.com/jp/

[8] A. Uchiyama *et al.*, in *Proc. PCaPAC2008*, pp.145–147.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP16

CMS ECAL DETECTOR CONTROL SYSTEM UPGRADE PLAN FOR THE CERN LARGE HADRON COLLIDER LONG SHUTDOWN II

R. Jiménez Estupiñán, D. Di Calafiori, G. Dissertori, L. Djambazov, W. Lustermann, S. Zelepukin[1], ETH, Zurich, Switzerland
[1]also at University of Wisconsin-Madison, USA

Abstract

The Electromagnetic Calorimeter (ECAL) is one of the detectors of the Compact Muon Solenoid (CMS) experiment at the CERN Large Hadron Collider (LHC). The ECAL Detector Control System (DCS) software has been implemented using the WinCC Open Architecture (OA) [1] platform. Modifications that require fundamental changes in the architecture are deployed only during the LHC long shutdowns. The upcoming long shutdown (2019-2020) offers a unique opportunity to perform large software updates to achieve a higher modularity, enabling a faster adaptation to changes in the experiment environment. We present the main activities of the ECAL DCS upgrade plan, covering aspects such as the re-organization of the computing infrastructure, the consolidation of integration tools using virtualized environments and the further usage of centralized resources. CMS software toolkits are evaluated from the point of view of the standardization of important parts of the system, such as the machine protection mechanism and graphical user interfaces. Many of the presented features are currently being developed, serving as precursors to the major ECAL upgrade foreseen for the next long shutdown (~2024-2025).

INTRODUCTION

The CMS Electromagnetic Calorimeter (ECAL) detector is composed of a scintillating crystal calorimeter and a lead/silicon preshower. The CMS experiment takes collisions data at the LHC, requiring extremely high reliability and the minimum down-time of the various detectors. The ECAL detector is subdivided in partitions as follows: Barrel (EB), Endcaps (EE) and Preshower (ES). The EB partition consists of 36 Supermodules (SM) each containing 1700 crystals, the EE consists of two endcaps split in four semi-circles (Dees), each containing 3662 crystals, and the ES consists of two circular structures.

The DCS controls and monitors the status of the hardware of each partition: cooling, powering systems, safety systems, environmental monitoring systems and other external interfaces. The DCS software coordinates the interaction between the different subsystems, providing an effective and meaningful way of operating the detector. From the architectural point of view, the DCS software is built by a hierarchy of components, whose main features can be organized into one of the following categories: Systems Configuration (peripheral addresses, database archivers, alarms, notifications, etc.), Finite State Machine (FSM), Automatic Actions (AA) and User Interfaces (UIs).

The CMS ECAL DCS [2] has successfully supported the ECAL operations since the CMS commissioning, more than 10 years ago. The CMS detectors maintenance is mainly driven by the LHC schedule, often restricted to periods of 2 to 5 days, called Technical Stops (TS). Major modifications are postponed to the Extended Year-end Technical Stops (EYETS) or the LHC Long Shutdowns (LS). The next LS will last for about two years, starting at the end of 2018. During this period, also known as the Second Long Shutdown (LS2), multiple activities across the LHC, Injectors and LHC Experiments will be performed [3]. Following the previous re-integration and consolidation of the CMS ECAL DCS [4], the LS2 provides an ideal opportunity to bring new features into operation, improve the maintainability of the CMS ECAL DCS software and increase the level of availability of the systems.

DCS UPGRADE PLAN OVERVIEW

Multiple activities are scheduled for the LS2; however, this paper focuses on those that imply significant changes to the DCS architecture:

- Re-configuration of the low voltage powering system. The device distribution at the Controller Area Network (CAN) level will be modified to increase the overall performance of the system;
- Extension of the protective automatic actions. The plan includes a proposal to increase the granularity of the current actions, the implementation of new ones and the evaluation of existing frameworks;
- Software standardization. The existing development guidelines will be extended and applied during the creation/modification of every component.
- Consolidation of the CMS ECAL DCS test platform. The usage of CMS automated deployment and testing tools will contribute to have faster development cycles, while reducing the time spend on maintenance;
- Improvements in the Programmable Logic Controller (PLC) based protection and safety systems.

Some of the LS2 activities present dependencies that define the order in which they must be executed (e.g. software updates come after the changes in the hardware layout). The LS2 upgrade plan is organized according to the tasks priority and dependencies (See Fig. 1) and it will be refined in the upcoming months, in order to meet all the requirements presented by the ECAL community.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP16

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

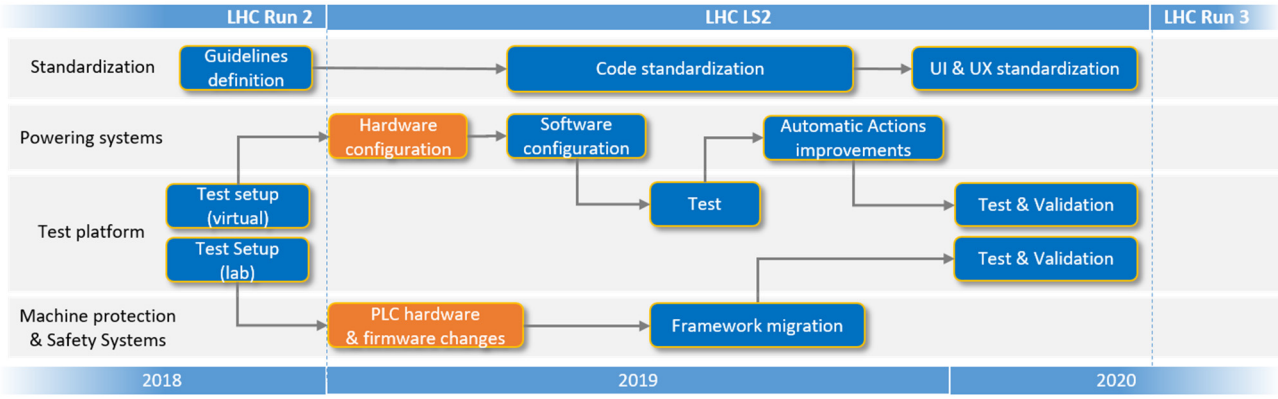


Figure 1: ECAL DCS upgrade plan for the LS2.

RECONFIGURATION OF THE LOW VOLTAGE POWERING SYSTEM

The ECAL powering systems are classified into two categories: Low Voltage (LV) and Bias Voltage (BV) power systems. The LV system for the EB and EE partitions is composed of 136 Wiener [5] Marathon power supplies, distributed across 10 CAN buses. The CAN buses for the LV system are connected to CAN-to-Ethernet interfaces enabling the access over the CMS private Ethernet network. The EB/EE BV and ES LV and BV are based on CAEN [6] power supplies using built-in Ethernet interfaces.

In the past years, the connection between the DCS and the EB/EE LV system has been affected by disruptions in the CMS private Ethernet network. When the communication is disrupted, the driver qualifies the data as invalid, similarly to an internal failure of the power supplies. Upon this condition, the CMS ECAL DCS is programmed to shut down the affected partitions (SMs or Dees) as a preventive action. After a few of such incidents, several tests were performed to understand the data invalidation process with the help of a network disruptor device. The tests focused on the two aspects: the disruption time required by the driver to invalidate the data and the time required to reconnect and restore the communication. Among other things, the tests revealed that the performance of the driver depends on the distribution of devices across the different buses. This means that the most populated bus determines the overall recovery time after such incident. This particular conclusion motivated the following activities for the LS2: extension of the number of buses connected to the CAN bus interface from 10 to 12 buses, redistribution of devices across the buses (maximum 15 devices per bus) and the reorganization of the DCS software according to the new interface layout (See Fig 2.). These changes, together with a new driver's configuration, will help to increase the overall performance of the LV network, making the DCS more tolerant to disruptions. This task requires the assignment of new peripheral addresses, the renaming of multiple structures and a crosswise reconfiguration of the software to ensure the correct integration of the changes.

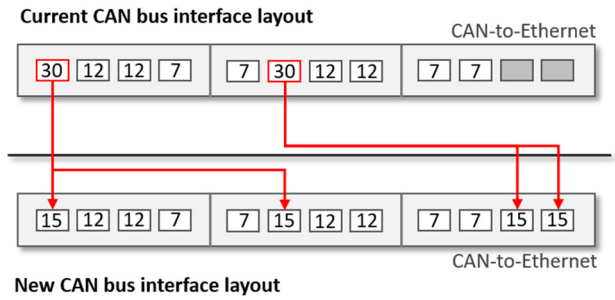


Figure 2: Changes in the CAN-to-Ethernet interface layout. The number of power supplies in each CAN bus is indicated in the white squares.

EXTENSION OF THE PROTECTIVE AUTOMATIC ACTIONS

The automatic actions in the DCS are preventive measures intended to protect the detector before escalating the problem to the safety systems. Actions are programmed to prevent delivering power during harmful conditions (e.g. absence of cooling, high humidity, etc.). The current implementation will be analysed and possibly extended to include the following features:

- Creation of new triggering conditions based on temperature information of the power supplies.
- Finer granularity of the automatic actions for the EE and EB partitions.
- Standardization of the protective automatic actions.

Finer Granularity of the Automatic Actions

Currently, the automatic actions for the EB and EE partitions are executed with the granularity of one SM or Dee. Each SM is powered by 3 LV Marathon power supplies and 34 BV channels. Dees are powered by 7 LV Marathon power supplies and 12 BV channels. This means that a single channel failure in a LV power supply triggers an action over a large number of devices regardless of their status, compromising the CMS data quality. A concrete proposal to increase the granularity of the automatic actions will be presented in the upcoming months, following a thorough analysis to prove the feasibility of this modification.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP16

Standardization of the Protective Automatic Actions

The protective automatic actions in the CMS ECAL DCS are programmed following different approaches [7]. The ECAL Finite State Machine (FSM) embeds most of the automatic actions for the EB and EE partitions, while the rest of the actions are implemented in custom control scripts. Other mechanisms to program automatic actions are also available to the CMS community. The Detector Protection is a software framework using the WinCC OA distribution capabilities and its native locking mechanism for exclusive data access. This framework offers a single interface to configure and monitor actions across distributed systems in a uniformed way. This framework is currently used by multiple projects within CMS and its usage will be considered after evaluating aspects such as the scalability of the software.

CONSOLIDATION OF THE TEST PLATFORM

The current CMS ECAL DCS computing infrastructure is composed of three redundant servers. Each server features 16 CPU cores, 32 gigabytes of RAM and runs the Windows 2008 Enterprise R2 operating system. The DCS software components are distributed among these servers according to their functionality and the hardware connected to them. The computing load and distribution of concerns is considered optimal, resulting in a highly cohesive system with very low coupling between the different software components. The configuration of the control platform (distributed and redundant connections, network parametrization and components layout) is stored in a central installation database.

Since the beginning of 2018, the CMS ECAL DCS team uses a set of CMS installation tools to perform unattended software deployment in the test setups. These tools are able to access the DCS installation databases and build the latest version of the software using the official code repositories. In addition to this, the installation tools are able to monitor changes and to synchronize the development copies with the production systems. The computing hardware used for the test setup is similar to the one used in production (except for the redundant set of nodes) and it will be extensively used during the LS2 for performance analysis, debugging and automated testing.

Virtual Test Environment

In addition to the physical test setup, three virtual machines from the CMS computing infrastructure are also available for testing. Each of them provides with 4 virtual CPUs, 8 GB of RAM and 40 GB of disk storage, sufficient to run an offline copy of the CMS ECAL DCS software. This virtual setup is particularly useful for developing multiple features in parallel, while minimizing the time required for maintaining the infrastructure. The usage of virtual machines permits the reinstallation of the operating system within minutes, having a ready-to-use copy of the CMS ECAL DCS software within a few hours. The virtual environment will be used during the LS2 to fragment and speed up the different software developments (See Fig 3.).

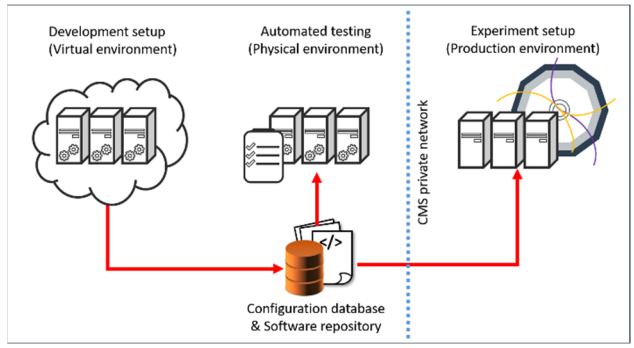


Figure 3: Software deployment process.

SOFTWARE STANDARDIZATION

The CMS ECAL DCS software was written by multiple developers, using different design criteria and programming styles. During the latest re-integration, the software was reorganized following a set of recommendations for controls systems development, in compliance with the existing JCOP framework [8] and CMS Central DCS integration guidelines [9]. Part of this reorganization work focussed on merging applications, migrating the DCS to the latest generations of software/hardware technologies, while adapting the architecture to run in fewer but more powerful servers. The resulting architecture was successfully adjusted and simplified, enabling the reduction from fifteen to three servers. Many software components execute similar tasks over different domains and areas of the detector (e.g. configuring and monitoring hardware, triggering protective automatic actions, etc.). However, the different implementations may differ substantially from each other, making the overall architecture sometimes difficult to evolve or maintain. One of the goals for the LS2 is to extend the existing guidelines, homogenize and incorporate new functionality by means of common libraries. The new guidelines will put the focus on specific naming conventions and programming best practices to improve the ECAL DCS software in the following areas:

- Code styling: Uniform naming and declaration rules for functions and variables will be introduced incrementally to improve the clarity, maintainability, extendibility of the code;
- Shared functionality: Similar tasks across components will be abstracted and encapsulated (e.g., resolution of hardware dependencies, storage of mappings, etc.)
- Debugging and diagnostics tools: New functionality will be introduced to extract run-time information from the systems and to improve the application message logging;
- User interfaces: Some of the user interfaces will be redesigned to improve information handling and to ease the support service during the on-call operations.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP16

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

MACHINE PROTECTION AND SAFETY SYSTEMS

The CMS ECAL detector features Siemens PLC [10] based safety systems, in addition to the CMS Safety System (DSS). The environmental conditions inside the EB and EE partitions are monitored through custom readout units, transmitting the sensors data to the PLC through RS485 buses. The environmental conditions inside the ES partitions are directly monitored by two other controllers, whose core functionality is implemented using the CMS Tracker PLC framework. The Tracker PLC framework is a compilation of tools to develop PLC applications for particle detectors. The framework permits a certain level of parametrization (alarm limits, deactivation of broken sensors, etc.) from the DCS supervisory layer. At the moment, these features are present in the two ES PLCs and some of them will be activated and integrated into the DCS interface. Further steps in the migration to this framework are also foreseen for the EB and EE Safety PLC, aiming to provide a higher level of support using the CMS ECAL DCS user interfaces.

CONCLUSION

Starting from a well consolidated software, the CMS ECAL DCS team seeks now for an efficient, extensible and more maintainable architecture. The adoption of common frameworks, development guidelines and tools will help to reduce the maintenance efforts while improving the support of critical parts of the system. In addition to this, the changes in the CAN bus interface and the protective automatic actions will contribute to increase the overall performance of the detector by providing a more reliable and robust system for the next physics operations in 2020.

ACKNOWLEDGEMENTS

The authors would like to thank the Swiss National Science Foundation for the financial support, as well as the CMS Central DCS team and the CMS Tracker DCS team for sharing some of the tools mentioned in this paper.

REFERENCES

[1] SIMATIC WinCC OA, http://www.etm.at/index_e.asp

[2] D. R. S. Di Calafiori *et al*., "The CMS ECAL Detector Control System", in *Proc. ICALEPCS'09*, Kobe, Japan, Oct. 2009, paper THA002, p. 633.

[3] M. Bernardini *et al*., Long Shutdown 2 @ LHC, CERN, Geneva, Switzerland.

[4] O. Holme *et al*., "Re-integration and Consolidation of the Detector Control System for the Compact Muon Solenoid Electromagnetic Calorimeter", in *Proc. ICALEPCS'13*, San Francisco, USA, Oct. 2013, paper MOPPC035, p. 154.

[5] Wiener Electronics for Research & Industry, http://www.wiener-d.com/

[6] CAEN Tools for discovery, http://www.caen.it/

[7] R. Jimenez Estupinan *et al*., "Improving the safety and protective automatic actions of the CMS Electromagnetic Calorimeter Detector Control System", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, paper THPHA109, p. 1639.

[8] JCOP Framework team, "Joint Controls Project (JCOP) Framework sub-project guide-lines and conventions", CERN, Geneva, Switzerland, EDMS ID:1100577, Oct. 2010, https://indico.cern.ch/event/151538/attachments/155675/220269/jcopFrameworkGuidelines.pdf

[9] R. Gomez-Reino, CMS DCS Integration Guidelines, 2007, [online], https://twiki.cern.ch/twiki/bin/view/CMS/DCSIntegrationGuidelines

[10] Siemens PLC, https://www.siemens.com/global/en/home/products/automation/systems/industrial/plc.html

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP17

EXTENDING THE REMOTE CONTROL CAPABILITIES IN THE CMS DETECTOR CONTROL SYSTEM WITH REMOTE PROCEDURE CALL SERVICES

R. Jiménez Estupiñán, ETH Zurich, Zurich, Switzerland
Attila Racz, Christian Deldicque, Christian Wernet, Christoph Schwick, Cristina Vazquez Velez, Dainius Simelevicius [1], Diego Da Silva Gomes, Dinyar Rabady, Dominique Gigi, Emilio Meschi, Frank Glege, Frans Meijers, Hannes Sakulin, Jeroen Hegeman, Jonathan Richard Fulcher, Luciano Orsini, Maciej Gladki, Marc Dobson, Michael Lettrich, Philipp Brummer [1], Thomas Reis, CERN, Geneva, Switzerland
Ulf Behrens, DESY, Hamburg, Germany
Audrius Mecionis [2], Jean-Marc Andre, Mantas Stankevicius [1], Nicolas Doualot, Petr Zejdl[3], Remigius K. Mommsen, Srecko Morovic, Valdas Rapsevicius [1], Vivian O'Dell, FNAL, Chicago, Illinois, USA
Christoph Paus, Georgiana-Lavinia Darlea, Guillelmo Gomez-Ceballos, Zeynep Demiragli, MIT, Cambridge, Massachusetts, USA
Andrea Petrucci, Rice University, Houston, Texas, USA
Ioannis Papakrivopoulos, Technical University of Athens, Athens, Greece
Samim Erhan, UCLA, Los Angeles, California, USA
Andre Holzner, James Branson, Marco Pieri, Sergio Cittolin, UCSD, San Diego, California, USA
[1]also at Karlsruhe Institute of Technology, Karlsruhe, Germany, [2]also at Vilnius University, Vilnius, Lithuania, [3]also at CERN, Geneva, Switzerland

Abstract

The CMS Detector Control System (DCS) is implemented as a large distributed and redundant system, with applications interacting and sharing data in multiple ways. The CMS XML-RPC is a software toolkit implementing the standard Remote Procedure Call (RPC) protocol, using the Extensible Mark-up Language (XML) and a custom lightweight variant using the JavaScript Object Notation (JSON) to model, encode and expose resources through the Hypertext Transfer Protocol (HTTP). The CMS XML-RPC toolkit complies with the standard specification of the XML-RPC protocol that allows system developers to build collaborative software architectures with self-contained and reusable logic, and with encapsulation of well-defined processes. The implementation of this protocol introduces not only a powerful communication method to operate and exchange data with web-based applications, but also a new programming paradigm to design service-oriented software architectures within the CMS DCS domain. This paper presents details of the CMS XML-RPC implementation in WinCC Open Architecture (OA) Control Language using an object-oriented approach.

INTRODUCTION

CMS DCS applications are implemented using the SIMATIC WinCC OA [1] platform, mostly written in its native programming language called control (CTRL) language. The CTRL language is a C-like language with a very poor type definition syntax, not suitable for programming complex software architectures. The CMSfwClass [2] is a programming framework to build object oriented (OO) applications using CTRL language. The CMS XML-RPC toolkit relies on the CMSfwClass framework, which permits a better implementation of the software architecture after an extensive planning and design phase. The toolkit provides a set of software classes to build client-server architectures (See Fig. 1), enabling heterogeneous software entities to act as service providers (servers) or service requesters (clients).

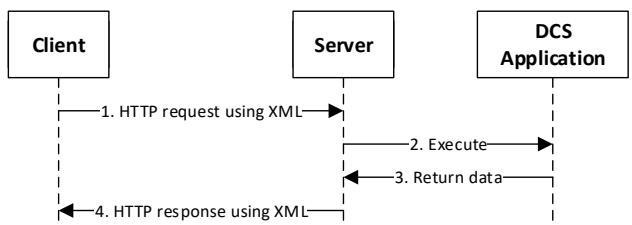


Figure 1: Client-server activity diagram on XML-RPC.

USE CASES

During the design phase, the software models were prepared to support at least the following scenarios: Remote procedure calls from the DCS user interface (UI), service-oriented collaboration between DCS applications, and DCS web services.

Remote Procedure Calls from the UIs

The CMS DCS is a large distributed environment, organized in a hierarchy of nodes and accessed from remote lo-

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP17

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

cations. In this context, the code associated to UIs is executed in the client machines while the events and data processing happens in the control nodes. Certain operations need access to resources that are only available in a remote control node (server). The CMS XML-RPC toolkit allows programmers to encapsulate server-only operations in the form of service objects that can be invoked from the UI. This feature allows the UIs to access resources, which are typically not accessible on the client side (E.g. local server files, system calls, etc.). In addition, the OO implementation hides the server details from the UIs; delimiting the concerns of the software and resulting in a cleaner implementation, compared to the classic procedural approach.

Service-oriented Collaboration

Many of the DCS applications are implemented as standalone applications, which are not ready to share functionality without further development. The CMS XML-RPC toolkit introduces a mechanism to model DCS functionality as a service. Different applications can be prepared to act as producers (servers) and consumers (clients) of services, permitting a well-structured collaboration interface between remote peers. In addition, this mechanism provides an abstraction layer to connect applications on different platforms.

DCS Web Services

The CMS XML-RPC toolkit complies with the standard XML-RPC specification using HTTP over TCP. This means that any application implementing these protocols can make use of the features exposed by DCS web services, including standard web applications. The CMS online portal (https://cmsonline.cern.ch) is a web portal hosting DCS related applications. Some of its applications connect to DCS web services; offering a new set of control features to the CMS users worldwide.

ARCHITECTURE

The CMS XML-RPC toolkit does not provide ad-hoc functionality. Instead, it provides a software model that can be used or extended to build new tailored client-server applications. The model includes two abstract classes describing the basic functionality of a service dispatcher and a service client. These classes are unaware of the final implementation details, as well as the message-encoding format, permitting the further extension of the architecture. In addition, the model includes concrete classes to build different client-server architectures.

Service Classes

The service classes are the core structures for building service based applications. The toolkit includes three subclasses derived from the abstract service class. Each of them extends the initial service implementation.

- Service class: This class implements the basic service functionality. Objects of this class offer a list of procedures to be executed remotely in the DCS context.
- Service router class: This class groups multiple service objects; offering a single entry point to clients. Objects of this class are particularly useful to concentrate and minimize the number of connections between the DCS and other platforms.
- Proxy server class: This class implement a message forwarding mechanism. Objects of this class can intercept, filter and divert requests to other service providers.

Service System Interface

The service system interface describes a list of popular RPC functions. The system interface is not part of the XML-RPC protocol itself, but it is a common feature supported by most of the available RPC servers. Using the system functions, clients can request the full list of available methods, their signatures or even the encapsulation of multiple calls to a remote server into a single request.

Client Classes

The CMS XML-RPC toolkit includes a set of service client classes, derived from an abstract service client class. Each of them implement different ways of invoking services.

- Client class: This class implements a basic method invocation by forwarding requests to specific service objects. Objects of this class use the internal WinCC OA run-time database to pass the messages to the service objects.
- Web client class: This class forwards the requests to a web service. Objects of this class listen to specific host ports using the transmission control protocol (TCP).
- Http client class: This class extends the web-service client functionality by using the HTTP protocol over TCP.

Proxy Server Configuration

By combining the server and client functionality into a single class, objects can configure a proxy server capable of serving, requesting, and therefore forwarding requests to other peers (see Fig. 2). A proxy server object can be configured to administer a group of resources, acting as a firewall between the CMS DCS web server and external clients.

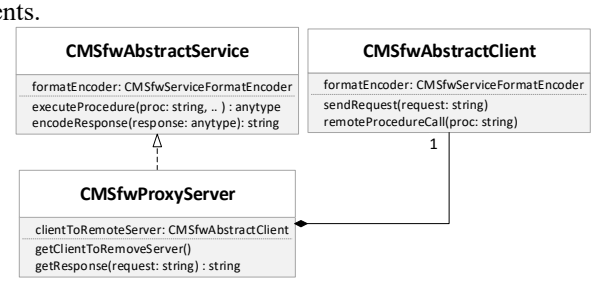


Figure 2: Service-client composition.

Message Formatting Classes

The information exchange between clients and servers require the encoding and decoding of messages. Clients should encode their request and transmit them using one of the available formats. Servers will decode the messages and process the request. The results are sent back to the

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP17

client, repeating the encoding-decoding operations on both sides. The CMS XML-RPC toolkit delegates the format encoding operations to objects implementing the message format. The different formatting classes implement the same interface; which defines the common methods for encoding and decoding messages. Thanks to this model, new formats can be easily included or extended without altering the final architecture.

IMPLEMENTATION DETAILS

The CMS XML-RPC toolkit is composed of 13 classes, 8 class interfaces and a Web service application handler. Once the framework is installed, objects can be created using a factory script or any of the available serialization mechanisms. The configuration of the objects and the relations between them is what determines the final software architecture. In total, the framework comprises around 3000 lines of code, in addition to the code required for building the final application.

Web Service Application Handler

WinCC OA provides an application programming interface (API) for building a standard HTTP server using CTRL language. The CMS XML-RPC toolkit uses the HTTP server API to implements a web-service application handler. Instances of any service class can be passed as parameters to the application handler to run a fully functional web service.

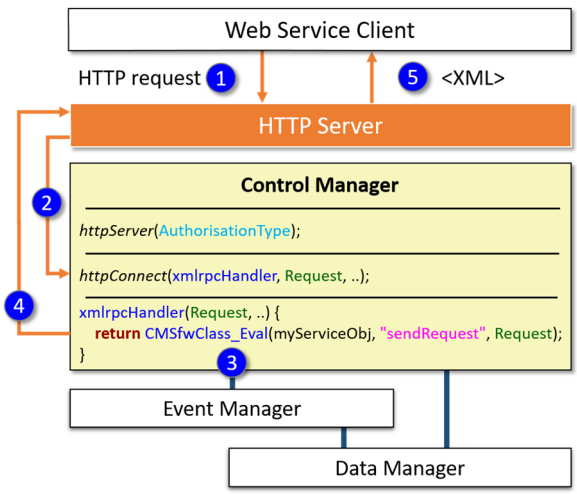


Figure 3: CMS XML-RPC application handler.

As shown in Fig. 3, web clients initiate the procedure with a HTTP request to the server. The application handler configures the HTTP server to listen to a particular port number in the control node. When the request arrives to the server, a call-back function processes it and delegates the execution to a service object. The service object decodes the request, executes the requested operation and encodes the results in the appropriate format. Results are passed back to the HTTP server, which replies to the corresponding client.

Message Format

The CMS XML-RPC toolkit provides two format-encoding classes. The first formatting class complies with the standard XML-RPC protocol specification. The second class implements a variant where responses are formulated in JSON. The usage of the JSON format is only available to web clients, which are able to decode this type of messages. This format is particularly useful in the context of JavaScript applications, since the encoding is part of the language notation. Messages encoded in JSON are lighter than the ones in XML (see table 1) requiring less space and processing time. Using one of the XML-RPC services, we have measured the length of the messages in the different formats (See Table 1).

Table 1: Size of the Messages

Type of message	Format	Bytes
Client request	XML	111 bytes
Server response	XML	1,042 bytes
Server response	JSON	597 bytes

A single query to the system returned a list of 25 items with 534 bytes of content data. The format of the XML message took more than 50% of the total message space. By contrast, the JSON format required only 1% of the message space. For this reason, JSON is the preferred format to exchange data with online applications.

INTEGRATION WITH CMS ONLINE

CMS Online is a web portal using the Oracle WebCenter Portal [3] technology to access technical information of the CMS experiment. The CMS online portal extends the DCS capabilities by offering a set of tools to monitor and administer parts of the DCS. Initially, the data exchange between the CMS Online and the DCS was limited by the usage of databases. Now, the CMS XML-RPC toolkit adds a new connection method to exchange data and perform remote operations from the CMS Online platform. At the moment, two web applications using the CMS XML-RPC toolkit are available:

- Online parametrization browser: This web application allows the users to inspect and change certain parameters in the control systems.
- Online DCS log files browser: This web application exposes the different logs available in the remote nodes, and implements some formatting and filtering features to facilitate the readability.

CONCLUSION

The implementation of the CMS XML-RPC toolkit started as a prototype to prove the feasibility of the XML-RPC protocol within the DCS domain. With the time, after several iterations to extend and refine its design, the toolkit became a consolidated part of the CMS DCS software. As result, the online applications based on the XML-RPC have also become an important tool for the CMS DCS community, allowing experts to access DCS technical information with the only help of a web browser.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP17

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ACKNOWLEDGEMENTS

The authors would like to thank the Swiss National Science Foundation for the financial support.

We would like also to thank our colleague L. Masetti, who provided insight and expertise in the topics discussed in this paper.

REFERENCES

[1] SIMATIC WinCC OA [online], http://www.etm.at/index_e.asp

[2] R. Jimenez Estupiñan et al. "Enhancing the Detector Control System of the CMS Experiment with Object Oriented Modelling", ICALEPCS'15, Melbourne, Australia, October 2015, paper MOPGF025.

[3] Oracle WebCenter Portal [online], https://www.oracle.com/technetwork/middleware/webcenter/portal/overview/index.html

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP19

TINE RELEASE 5.0: A FIRST LOOK

P. Duval, J. Szczesny, T. Tempel, DESY, Hamburg, Germany
S. Weisse, DESY, Zeuthen, Germany
M. Nikolova, EMBL-Hamburg, Germany
J. Bobnar, Cosylab, Ljubljana, Slovenia

Abstract

The TINE [1] control system evolved in great part to meet the needs of controlling a large accelerator the size of HERA, where not only the size of the machine and efficient online data display and analysis were determining criteria, but also the seamless integration of many different platforms and programming languages. Although there has been continuous development and improvement during the operation of PETRA, it has now been 10 years since the last major release (version 4). Introducing a new major release necessarily implies a restructuring of the protocol headers and a tacit guarantee that it be compatible with its predecessors, as any logical deployment and upgrade strategy will entail operating in a mixed environment. We report here on the newest features of TINE Release 5.0 and on first experiences in its initial deployment.

INTRODUCTION

Originally a spin-off of the ISOLDE control system [2], TINE is both a mature control system, where a great deal of development has gone into the control system protocol itself, offering a multi-faceted and flexible API with many alternatives for solving data flow problems, and it is a modern control system, capable of being used with both cutting-edge and legacy technology. In addition to publish-subscribe and client-server transactions offered by many other control systems, TINE supports multi-casting and contract coercion [3]. As the TINE kernel is written in straight C and based on Berkeley sockets, it has been ported to most available operating systems. Java TINE, with all of its features, is written entirely in Java (i.e. no Java Native Interface). All other platforms, from .NET to Matlab to LabView to Python, make use of interoperability with the primary TINE kernel library. Furthermore, any client or server application based on TINE and its central services does not require any non-standard or third party software (i.e. there are no LDAP, MySQL, Oracle, Log4j, etc. dependencies).

The transition to TINE Release 4.0 was reported some time ago [4], where numerous features of TINE were enumerated, some of which (e.g. multicasting, redirection, structured data) set it apart from other control systems in common use. In addition, TINE offers a wide variety of features designed for efficient data transport and communication in large systems.

A series of meetings in 2012 identified long-term goals and established a roadmap for the future Release 5.0. Many of these goals have been realized over the past several years, showing up in new minor release versions of TINE, the last being version 4.6.3. What sets Release 5.0 apart and warrants a new major release number are some necessary changes to the protocol headers.

In the following we will identify and discuss those relevant embellishments which have ensued since the 2012 meetings and have culminated in TINE Release 5.0.

RELEASE 4 ISSUES

As noted in the introduction, a general collaboration meeting in 2012 identified certain aspects which needed to be addressed. These include the following.

Protocol Issues

The TINE protocol makes use of Berkeley sockets and TINE Release 4 originally did not properly support IP version 6 (IPv6), as the socket API calls used were all IPv4 centric. Although there is no mad rush to use IPv6, it does offer advantages which could be of interest in the not too distant future.

Header Issues

Several nice-to-have features, which potentially make life easier for administrators tracking connectivity problems, could only be added by expanding the existing protocol headers (and thereby requiring a new major release). For instance the process ID and application type of a connected client are not available under Release 4.

In addition, some supported features required workarounds under some circumstances, which could also only be ironed out by additional information not currently available in the Release 4 protocol headers. For instance, a generic client making a request to a server for a property's canonical data set can ask for the DEFAULT data set (and thus avoid an independent query to obtain the property characteristics). The returned data header will in fact provide the proper data format, but not explicitly give the correct data size. The latter can usually be inferred from the number of data bytes returned. However, if the request in question was truncated by the server, then the property data size which *should* be used in a request is an unknown quantity.

Finally, large data sets often require packet reassembly in the TINE kernel. For example, IPv4 jumbo datagrams can have a maximum length of 64 Kbytes. Any larger data set will require assembling multiple packets. In Release 4, the request and response headers hold the total message size in bytes in an unsigned short, i.e. precisely the 64 Kbytes of an IPv4 jumbo datagram. TINE transfers can of course use a TCP stream, or shared memory, rather than datagrams, but the same packet reassembly exists.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP19

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

That in itself is not a problem, except that it is often useful to specify a larger number for the message size in bytes, necessitating a 4-byte integer in the transport headers, rather than the current 2-byte integer.

Other Issues

A TINE server developer can choose among a variety of platforms on which to write his server, including Java, Python, LabView, Matlab, and .NET, not to mention the operating system. Nevertheless, a number of production servers are written in C or C++, making direct use of the C library API. C++ developers are most likely to make use of Standard Template Library (STL) or Microsoft Foundation Classes (MFC) libraries and headers. If this is indeed the case, then certain measures must be taken to avoid namespace collisions when *tine.h* is included in the same code module as the STL or MFC headers. This primarily has to do with macro definitions attempting to override e.g. a class name and cannot be trivially solved by using a namespace wrapper around *tine.h*.

RELEASE 5 SOLUTIONS

Protocol Issues

TINE Release 4.5.0 introduced the standard IPv6 socket API to the TINE library and by Release 4.6.3 the TINE libraries in both C and Java were fully implemented. The general strategy is for clients and servers to make use of a dual stack if possible, where a single bound listening socket can support either protocol. IPv4 clients will then only ever see an IPv4 address. Likewise an IPv6 client will always see an IPv6 address, be it a real one or a mapped IPv4 address (with a leading '::ffff:'. Thus this aspect was concluded prior to the advent of Release 5.0.

Aside from removing the administrative headaches involved in making use of private networks and exhausting the IPv4 address space, IPv6 also offers jumbo datagrams up to 4,294,967,295 bytes.

Header Issues

The TINE Release 5 request headers have indeed been modified to pass a client's process ID and application type to a server, along with associated diagnostics which pass this information along (see Figure 1). The application type is composed of an 8-character string identifying the principal kind of client making the call. A middle layer server acting as a client will supply the text "FEC", for instance, whereas a Python client will supply the text "PyTine". A client's process ID is perhaps of little or no use if the client is a command line tool such as *tget* used in a script. However, for persistent clients it is a useful identification number which can expedite the search for a specific client application should it become problematic.

The new response headers also categorically supply a contract's canonical data size as well as the size in bytes and in elements returned in the call.

Entities such as the message size or MTU are also now categorically 4-byte integers.

A contract response header also continues to supply additional system and user *stamps* as 4-byte unsigned integers. These are in addition to the associated data's time stamp, and are typically used to provide an event or cycle number tag to the associated data. That is, these were specifically *not* upgraded to 8-byte integers, primarily to avoid issues on 32-bit (or 16-bit) platforms which do not support them. A quantity such as an event number will wrap only every 14 years or so even if incremented at 10 Hz, so this should not present a problem in the short term, and will essentially never present a problem if the said *event number* is reset at the beginning of a run. In the long term, these quantities can be upgraded at some future time, should the need arise.

Both the request and response headers also provide information on the endianness of the host machine and the character encoding in use in the data provided. In the current release (5.0.0) the endianness is fixed as little endian and the character set is fixed as ASCII standard. One could argue that a modicum of efficiency could be squeezed from the system if only one of a client-server pair engaged in byte swapping when it needed to. However, as a Release 5 server will need to support Release 4 clients, which expect little endian payloads, it would have to make the decision to swap or not at the point of delivery. This is currently problematic for some data types, such as a user-defined tagged structure and some non-fixed length data types and requires extensive refactoring of code. Therefore this issue has been postponed for some future release, which of course will have to make the identical swapping decision based on a client having a release number greater than e.g. 5.2, etc.

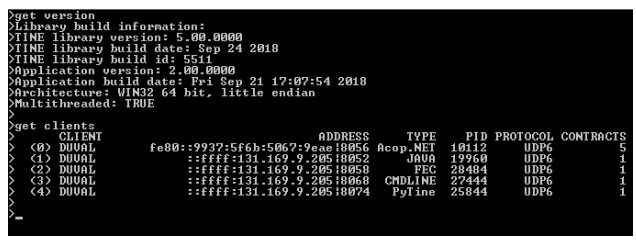


Figure 1: An example of a server's console command to show its attached clients. New to Release 5 are the PID and TYPE columns.

Other Issues

The problematic macro definitions (largely error/status codes) have essentially all be replaced with enumerations in Release 5. This has the advantage (as in the case of a macro definition) of not requiring constant variables in a program's data segment. Furthermore enumerations easily lend themselves to being used within a C++ namespace wrapper. Thus using the TINE API directly in C++ code no longer requires any extra measures to avoid namespace collisions. A namespace wrapper around the TINE header file *tine.h* is entirely optional.

UPGRADE STRATEGY

Any control system component making use of TINE Release 5 must be fully compatible with earlier releases

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP19

of TINE (predominately of Release 4 vintage). Release 5 servers must seamlessly interface with Release 4 (or Release 3) clients. And Release 5 clients must likewise be able to access earlier vintage servers. With this as an ansatz we can contemplate upgrading the control system elements adiabatically, with the expectation that legacy components will remain operational for months, if not years.

There is no *best moment* to roll out a new major release such as this, other than perhaps during a long shutdown, where there is often a prolonged re-animation of the machine. This happens infrequently. In any event, in the case of the PETRA III complex, no amount of unit testing will catch all compatibility issues, largely due to the multicultural aspects found in machine control there. For instance, there are critical Java servers running on both Windows and Linux hosts. There are 32-bit and 64-bit servers running not only on Windows and Linux hosts, but on VxWorks and LabView (also Windows) as well. Client applications are liable to be rich client Java applications using ACOP beans or jDDD (with its own complexities) or rich clients using Matlab, LabView, or using ACOP.NET [5].

As TINE is feature rich, there tends to be a wide variety of *ways to do things*. This in itself tends to increase the general entropy in a test environment.

Thus the path to general deployment was to test as much as possible, making use of the TINE unit server and client in combinations of Release 4 interfacing with Release 5, and then to deploy and react during the machine studies following a *mini* shutdown and prior to a user run. Here one can see which hiccups occur during normal operations and either rollback if necessary or find and fix (if the operators can tolerate the hiccup during a bit of extreme programing).

In fact, there were surprisingly few hiccups - three to be exact, two of which led to a rollback. Nonetheless, at the conclusion of the machine studies, TINE Release 5 was in place as the de-facto standard, although there will be a mixed scenario for some time to come. And to be sure, one still must continue to be on the lookout for hiccups and be ready to react.

LESSONS LEARNED

With the rollout of TINE Release 5, one generally hopes that, as far as the users and customers are concerned, *nobody notices anything*. That is to say, there are no new bells and whistles that would make transfers more efficient or offer new paradigms for application development. The API is basically unchanged. On the other hand, developers (especially server developers using the C library) will appreciate many of the new embellishments. Likewise administrators will find it easier to track communication problems.

The TINE core team will also be able to navigate through both the Java and C library code more easily, due to extensive refactoring. And as the latest TINE transport headers are extensible, it should prove to be a straightforward task to add fields to the existing headers at some time in the future should they be needed.

One somehow anticipates that by the mere act of shaking things up, i.e. not letting sleeping dogs lie, so to speak, various real problems (e.g. hidden race conditions) will be exposed. In the initial phase of the ensuing users run, two further upgrade issues in fact became apparent. One of these, a long-standing TCP issue which might occur when large input data sets are being collected at the server side, had *almost* no chance of expression in the Release-4 world and only became visible when the contract request headers increased in size in Release 5. This issue surfaced on a particular server and led to a local rollback until it was understood. The second issue involved a check on multi-channel contract coercion logic versus the minor release and revision numbers, which suddenly jumped back to 0 and 0. This latter issue had no *visible* consequences and was only noticed in that certain applications appeared to suffer in certain aspects of transfer efficiency. Both of these problems were promptly dealt with and neither had any direct bearing on the user run.

Introducing a new major release (or any systematic upgrade, for that matter) is not something one takes lightly at any time. In the absence of a full-blown mock facility which is actually used under *real* conditions, there is virtually no way to *catch things* other than to deploy and standby in extreme programming mode. All in all, there were surprisingly few hiccups due to specific software problems.

An additional hiccup was due to the non-synchronized deployment of system libraries. Although it had nothing to do with any software problems or Release 4/Release 5 compatibility issues, it would of course not have arisen had there been no attempt at an upgrade. Yet, this in itself exposed an existing problem (in this case, a misunderstanding in the software deployment on Windows hosts).

The moral of the story is: It sometimes takes a user run to expose problems! By now the dust has settled, so to speak, and one is gradually beginning to breathe more easily.

REFERENCES

[1] TINE website; http://tine.desy.de

[2] R. Billings *et al.*, "A PC Based Control System for the CERN ISOLDE Separators", in *Proc. ICALEPCS'91*, Tsukuba, Japan, Nov. 1991.

[3] P. Duval and S. Herb, "The TINE Control System Protocol: How to achieve high scalability and performance", in *Proc. PCaPAC'10,* Saskatoon, Canada, Oct. 2010, paper WECOAA02.

[4] P. Duval *et al.*, "TINE Release 4 in Operation", in *Proc. PCaPAC'08,* Ljubljana, Slovenia, Oct. 2008, paper MOX01.

[5] P. Duval *et al.*, "ACOP.NET: Not Just Another GUI Builder", presented at PCaPAC'18, Hsinchu, Taiwan, Oct. 2018, paper THCB1, this conference (and references therein).

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP21

INJECTION CONTROL OF THE TPS

J. Chen, Y. S. Cheng, C. Y. Wu, C. Y. Liao, K. H. Hu, K. T. Hsu
National Synchrotron Radiation Research Center, 30076 Hsinchu, Taiwan

Abstract

Injection control application served for Taiwan Photon Source (TPS) to help commissioning and operation of the machine. Top-up injection functionality is available from machine commissioning stage to accelerate vacuum conditioning. During last two years, several updates have been done to enhance flexibility for the injection control. The injection control includes foreground and background processes to coordinate the operation of e-gun, linear accelerator, booster synchrotron, storage ring by the help of event based timing system. Lifetime calculation of the storage ring is also synchronized with the injection process. Detail of the implementation will be presented in this report.

INTRODUCTION

The Taiwan Photon Source TPS [1] is a latest generation of high brightness synchrotron light source which is located at the National Synchrotron Radiation Research Center (NSRRC) in Taiwan. TPS consists of a 150 MeV electron linear accelerator (linac), a 3 GeV booster synchrotron, a 3 GeV storage ring, and experimental beamlines. Ground breaking for civil construction was held on February 2010. The civil construction completed in April 2013. Accelerator system's installation and integration started in later 2013. The control system environment readied in mid-2014 to support subsystem integration and test. After 4 months of hardware/software testing and improving, the commissioning of booster and storage ring was started in December 2014. First synchrotron light shines in the last day of 2014 [2].

Injection control application with top-up functionality was deploy at early period of commissioning phase to release loading of operator and increase beam dosage accumulation to accelerate vacuum conditioning. Constant current with lower and upper limit top-up injection was used for user service since September 2016. However, time interval between injections vary slightly which caused by beam lifetime variation affected by machine conditions such as coupling, nonlinear beam dynamics, insertion device parameters (gap and phase) changed, etc. User could hardly predict the timing of injection even hardware gating signal provided. The experimental data deteriorated by the injection transient. Some beamlines could use hardware gating signal to exclude the transient but most of the instruments and data acquisition systems which cannot gated by hardware gating signal were sensitive to the injection transient. The top-up injection control had been revised to fixed time interval between injections in early 2018. The time of next injection is predictable to suspend data acquisition during the injection window. Maximum stored beam current is set to 405 mA at the moment. The storage ring is refilled every 4 minutes. Beam current drops around 4 mA during this period, it takes about a few seconds (< 5 sec) to refill the stored beam current back to 405 mA.

INJECTION SUPPORT SYSTEM

The injection process need coordinate various subsystem such as electron gun, buncher, RF system of linear accelerator, injection and extraction pulse magnets of the booster synchrotron, and injection septa and injection kickers of the storage ring.

Linac and Booster Synchrotron

The TPS injector consists of a 150 MeV linac and a 3 GeV booster synchrotron. The 150 MeV linac was available for booster injection in the 2nd quarter of 2014. Measured linac beam parameters are well accepted with its specifications. The LTB transfer line has been successfully commissioned with beam and the magnet settings agree with designed expectation [3].

The booster synchrotron was commissioning in the third quarter of 2014 successful. Optimization were done in performance for routine operation. These optimizations consist of system upgrade to improve its weakness and the improvement of the ramping procedure to increase the capture and ramping efficiency of the beam charge, optics characterization [4].

Pulse Magnets and Pulsers

The pulse magnets of TPS consist of one booster injection septum and one booster injection kicker, two booster extraction septa and two booster extraction kickers, two storage ring injection septa and four storage ring injection kickers, and two storage ring pingers for diagnostic purpose [5]. Coordinate operation of these pulse magnets except pingers are necessary for storage ring injection.

Event Based Timing System

Event based timing system was adopted to coordinate operation of TPS include injection. The event system consists of event generator (EVG), event receivers (EVRs) and timing distribution fiber network [6]. Machine clocks (repetition rate, revolution clock of BR and SR, synch clock, etc.) are distributed by distributed bus. While trigger events for the injection are management by the sequence RAM inside of the EVG which installed the timing mater node which is the sources of major timing events. The 125 MHz event rate will deliver 8 nsec coarse timing resolution.

A control page for the EVG configure and testing as shown in Fig. 1, The interface provides a tool to configure event generator such as clock rate, sequence RAM, DBUS,

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP21

signal mapping, etc. To operate the accelerator efficiency, the injection control sequence was coded by state notation language (SNL) and execute by the EPICS sequencer running at the timing master as shown in Fig. 2. An injection interface page was implemented to provide a convenient interface to do beam injection for the storage ring. The injection sequence program running at the timing master EPICS IOC which communicate with the injection interface via process variables. Injection program is a state machine to help users to control the injection is shown in Fig. 3.

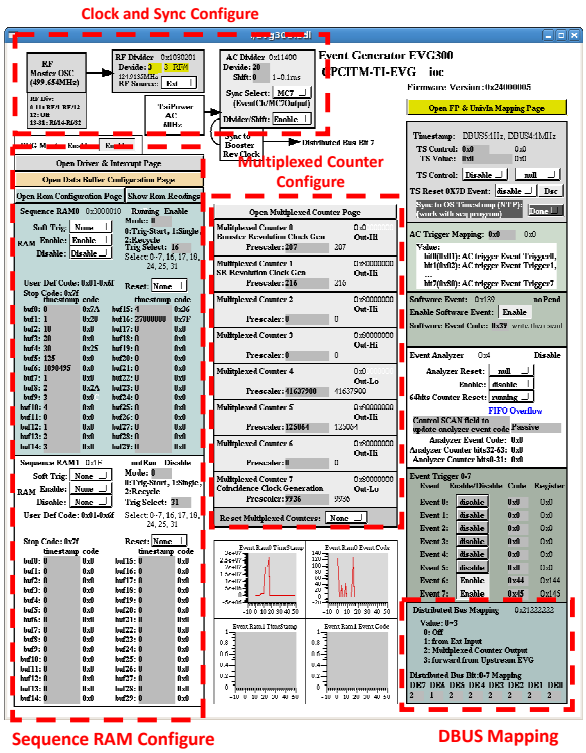


Figure 1: Event generator configuration graphical user interface.

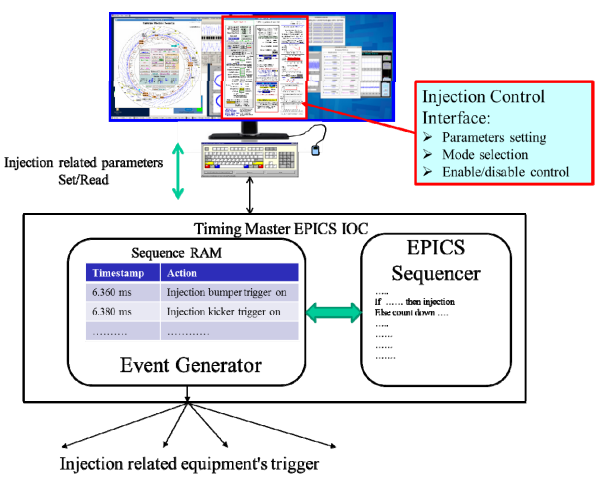


Figure 2: Injection control page communicate with the injection sequence program running in timing master EPICS IOC.

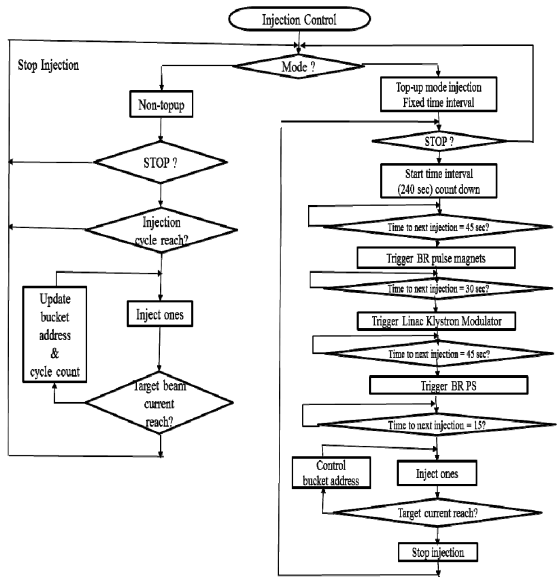


Figure 3: Sequence for injection control.

Filling Pattern Measurement

Filling pattern is an important factor which will affect storage ring operation performance such as lifetime, instability, etc. Keep the desired filling pattern are important during user operation which continue eleven-days operation every two weeks of current shift plans. Filling pattern should be monitor continuously. The acquired waveform from button signal by a dedicated oscilloscope with 40 GS/sec effective sampling rate trigger by revolution clock are used to extract bunch current with high resolution. The filling pattern analysis code extract bunch relative intensity accompany with total beam current measured by DCCT to calculate bunch current of individual bunch.

INJECTION INTERFACE

An injection control EDM page was implemented to support parameters configuration and operation of the injection as shown in Fig. 4. All of injection modes and sequences are managed by the content of sequence RAM which updated by the injection control EPICS sequencer program every injection cycle (0.333 second).

The left side of the page provide electron gun mode control, trigger enable, and dump beam setting. Timestamp stored in the sequence RAM is also listed.

Middle part of the injection control page consists of injection parameter setting, such as bucker address, injection cycle, target current setting, top-up control current range and time interval. The Top-up mode can select either multi-bunch mode or hybrid mode. If hybrid mode selected. it need specify the bucket address of isolated bunch. The injection process will be multiplexing the injection for multi-bunch and isolated bunch in hybrid mode as shown in Fig. 5.

Right parts provide synoptic display of beam current, filling pattern, beam lifetime, front-end status, and booster beam information.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP21

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

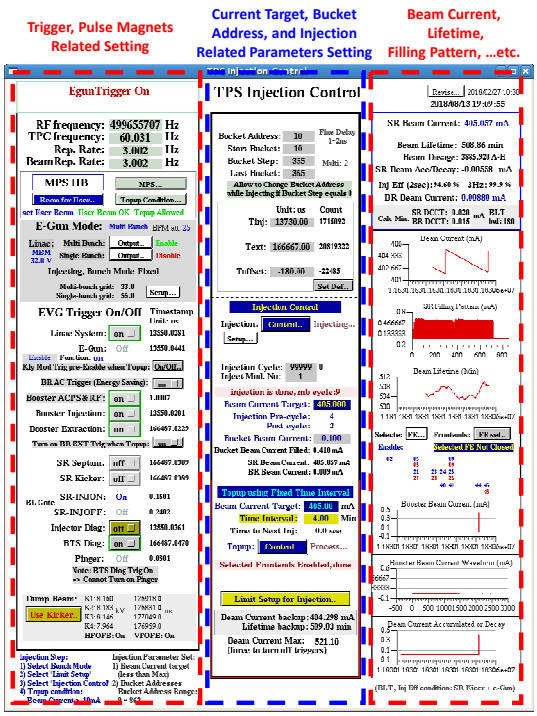


Figure 4: Injection control page.

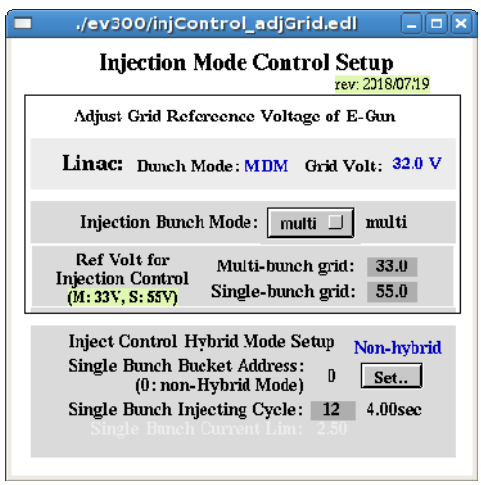


Figure 5: Top-up mode selection page.

Typical beam current and lifetime history, and filling pattern are shown in Fig. 6. Since the booster synchrotron can provide bunch-train near 1 μsec, current injection the electron gun and booster bunch train length as 700 nsec (350 bunches) in multi-bunch injection. The storage ring injection is only switch back and forth between two specific bucket location for bunch train injection until desired current accumulated. The gap of the lifetime data is due to lifetime calculation is stop during beam current accumulation. It need time to acquire sufficient data for regression before the first lifetime data available. So, the lifetime during this period is no updated.

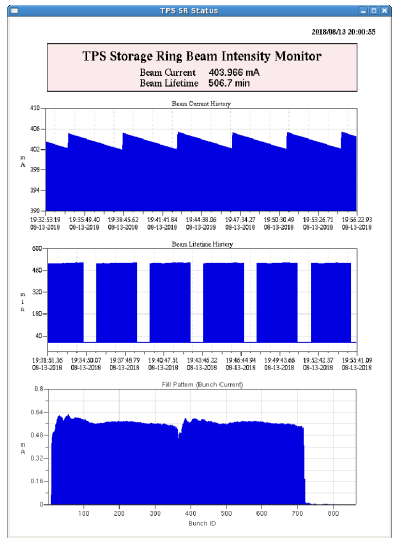


Figure 6: Beam current, lifetime, and filling pattern in typical user service shifts.

SUMMARY

Injection control for the TPS was implemented and revised several times since machine commissioning in late 2014. Event based timing system support synchronized action of various injection/extraction elements. Sequence RAM in event generator is used to configure the sequence of injection/extraction processes. Constant current mode was adopted from beginning of commissioning in late 2014 and user service started in September 2016. However, to help synchronize of experiment data acquisition with the injection process, constant time interval injection was chosen in early of 2018. The injection control functionality for TPS is working well and continue functionality revision will be performed to satisfy various requirements in future.

REFERENCES

[1] C. C. Kuo *et al.*, "Accelerator Physics Issues for TPS", in *Proc. IPAC'10*, Kyoto, Japan, May 2010, paper MOOCMH01, pp. 36-38.

[2] C. C. Kuo *et al.*, "Commissioning of the Taiwan Photon Source", in *Proc. IPAC'15*, Richmond, USA, May 2015, paper TUXC3, p. 1314.

[3] H. P. Chang *et al.*, "TPS Linac Relocation and Beam Test of the LTB Transfer Line", in *Proc. IPAC'15*, Richmond, USA, May 2015, paper TUPJE049, p.1731.

[4] H. J. Tsai *et al.*, "First Year Performance of the TPS Booster Ring", in *Proc. IPAC'16*, Busan, Korea, May 2016, paper MOPOR017, p. 634.

[5] C. Y. Wu *et al.*, "Control Interface of Pulse Magnet Power Supply for TPS Project", in *Proc. IPAC'15*, Richmond, USA, May 2015, paper MOPTY077, p.1120.

[6] C. Y. Wu *et al.*, "Integration of the Timing System for TPS", in *Proc. IPAC'14*, Dresden, Germany, June 2014, paper TUPRI113. p. 1833.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP22

THE LENS EFFECT IN THE SECONDARY EMISSION BASED SYSTEMS OF JOINT SEARCHING IN EBW*

A. Medvedev†, K. Blokhina, M. Fedotov, A. Starostenko, Y. Semenov, M. Sizov, A. Tsyganov
BINP SB RAS, 630090 Novosibirsk, Russia
A. Medvedev, BINP SB RAS, 630090 Novosibirsk, Russia

Abstract

The results of the developed scan lines generator for the magnetic correctors system are presented. Get the dependency between various types of scan lines and distribution of the allocated energy in the electron beam welding facility. The lens effect in the secondary emission based system of joint searching, using 3-fragment linear scan line is received. The accuracy of the joint searching system (the error of the positioning system) is 0.05 mm, the lens effect can decrease this value several times. The requirements for the creation full calibrated system of joint searching are listed.

INTRODUCTION

Electron beam technologies based on using beam source (electron gun) with accelerating voltage about some decades kilovolts and power from several watts to several decades kilowatts. The joint search system is very important part of the electron beam welding facility, it allows scan the surface of the target and sees the map of scanning area with asperities of the surface, in particular, the system allows see the joint.

Basically, the principle of functioning the joint search system is reflection and reemission the part of electrons, which interact with the target surface. The back current is measured by the isolated electrode. Different areas of the surface have different reflection coefficient, so we can see the different current values depending on the surface traits. In practice, we need the beam reflecting system with a reading of values from sensors.

EXPERIMENTAL SETUP

The experiments were conducted in the Budker Institute of Nuclear Physics (Siberian Branch of the Russian Academy of Science). The experimental setup has the vacuum chamber, the electron gun (up to 60 kV) with current ability up to 750 mA [1], two-coordinate reflection system based on cos(theta) coils, magnetic amplifiers (bipolar current sources up to 1.5 amperes), electrode, amplifier, and processing block of the joint search system.

Processing block include several channels of digital-analog converter which form the signal for magnetic system, one channel of analog-digital converter, which measure the value from the sensor, FPGA based device for synchronous communication between DAC and ADC, and the single-board computer, which provides the network access for staff of the facility, and realizes the interface to change settings of scanning (by changing internal values in FPGA). Figure 1 shows the main equipment of joint searching system, and interaction point with the rest of facility equipment.

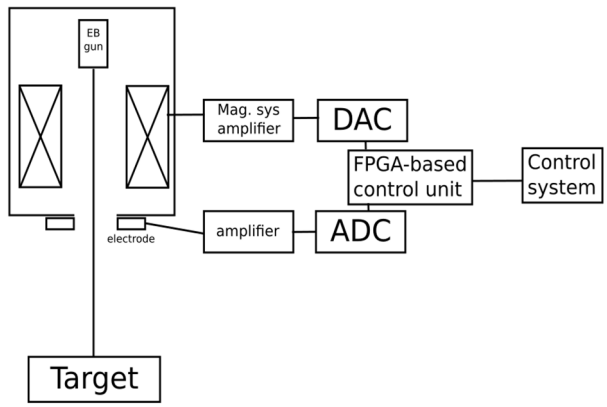


Figure 1: Scheme of the joint searing system.

THE SCANNING PROCEDURE

Despite the scanning system has two coordinates, usually for practice purpose we need only one of them. The beam intersects the joint each period of scanning, and the target movement system moves the detail along the joint. As a result, we can see the position error for all length of the joint. The dual scan uses when we need to get the frame of the scanning area. The experimental setup allows creating this type of scan.

We can illustrate the signal formed by the scanning system. In the simplest case, this signal is sawtooth. Usually, we use a sawtooth-like signal with positive and negative ramps (triangle wave). Using of the one-side sawtooth signal may give us negative effects with quality and appearance of the seam (in the case, when we weld the details with scan), so we use symmetric types of scan.

The scanning system forms the map of the surface, and the size of the map corresponds to the size of the scanning area. We can change the amplitude of the scan, depending on the view, which we want to see. It is possible, to get the ratio between scan amplitude and width of the scanning area. Due to the incline of the beam, the ratio will be changing for each new vertical position of the target. This problem can be solved by using 2 magnetic correctors, one above the other, with the birefringence of the beam. This moment we have the second pair of magnetic correctors but haven't made hardware and software modifications in the control system to get this opportunity. So we fix the distance between the beam gun and target in order to get

* supported by Russian Science Foundation (project N14-50-00080)
† alexey.m.medvedev@gmail.com

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP22

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

repeatability of experiments (approximately 100 mm from the end of the gun).

The quality of the plotted map depends on the noises level in reflected beam current, the geometry of the collector electrode, and the amplification factor. Besides, we can use programming methods to increase the quality of the image. One of them is filtering methods applying to the big set of data measured by special function in automatic mode. The user interface allows seeing the map in real time, with noises. It is very useful for a primary setting of the target position.

THE LENS EFFECT

The architecture of the joint searching system allows setting any data sequence to form the signal for the magnetic system. We develop the 3-stage linear signal, with the ability to set different incline and different size of each stage. Two parameters set the form of signal, and the program generates it automatically. The most interesting form shown on the Fig. 2. The beam moved with small velocity on the central part of the scan line and has high velocity at the edges of the scan line. The same time, the surface map shows the signal evenly in time, so the central part of the scan line has a bigger segment in the surface map. We named this "the lens effect".

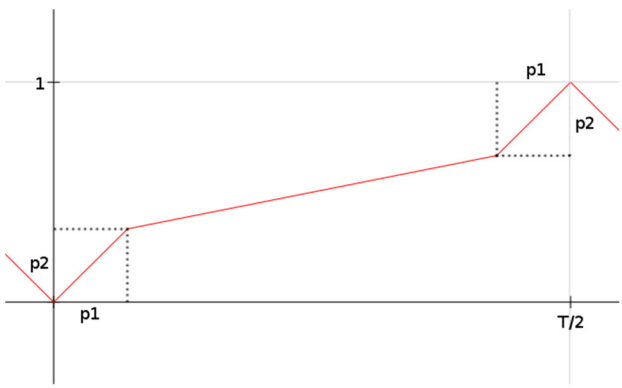


Figure 2: Three-stage linear signal.

The parameters of this digital lens could be adjusted by the scan line generator. The shape could be transformed in various waves with different lens effect (from triangle wave to straight line or square wave).

The shape of the signal influences to power distribution in the welding application. The form with lens effect in the central part of the scanning area increases energy release in the joint. Thus the depth of the seam is increasing and we can improve welding quality and weld with less beam current.

ACHIEVABLE ACCURACY

This section describes the boundaries and restrictions of the joint search system and says about accuracy, which we can reach on the facility. First, the beam has it's own size, the lens of magnetic optic system of the beam could reduce the size of the cross-section of the beam, but the minimum size is limited by gun geometry, beam quality and influence of space charge of the beam. Gun of BINP EBW facility has beam minimum size about 0.5 mm (with current used for scanning).

Second, the control signal is formed by 14 bit DAC. The amplitude of the beam trace with using maximum range of the DAC is 2 cm (could be increased by using another model of DAC). One LSB corresponds to 1.25 um of beam trace; this is the extreme achievable value of minimal beam movement. For scan shape, we use an array of forming a line with 500 values. The length of the trace segment between two neighboring values with maximum scan amplitude is 40 um. By reducing the amplitude or by using the mode with lens effect we can increase the resolution.

Besides the spatial properties, there are temporal characteristics of the scanning system. The frequency of the scan is limited by Foucault currents and capabilities of the DAC. Practically, we can achieve 1 kHz frequency of the scan.

CONCLUSION

We developed the joint search system based on secondary emission effect. Possibility to set any form of scan line allows getting the scanning system with lens effect. The minimum size the system can show, limited by the beam size, on the electron beam welding facility this value is about a one-tenth millimeter.

ACKNOWLEDGEMENTS

This work has been supported by Russian Science Foundation (project N14-50-00080).

REFERENCES

[1] Yu. I. Semenov, P. V. Logatchev *et al.*, “60 keV 30 kW electron beam facility for electron beam technology”, in *Proc. EPAC’08*, Genoa, Italy, Dec. 2008, paper TUPP161.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP23

CONTROL SYSTEM USING EPICS TOOLS AT TARLA LINAC

O.F. Elcim†, Institute of Accelerator Technologies, Ankara U., Ankara, Turkey

Abstract

The first accelerator based research facility of Turkey-TARLA is under commissioning at Institute of Accelerator Technologies of Ankara University. It is designed to generate free electron laser and Bremsstrahlung radiation using up to 40 MeV continuous wave (CW) electron beam. The control system of TARLA is based on EPICS and are being tested offline. TARLA also has industrial control systems such as PLC based cryoplant and water cooling system. Its control system is under development, it benefits from the latest version of EPICS framework, i.e. V7. In other words, TARLA control system uses existing demonstrated tools of EPICSV3 as well as pvAccess which comes with EPICSv4 for transferring the large data through control network. Archive (CSS BEAUTY) and alarm (CSS BEAST) system have been set up to detect stability and prevent failures. Operator interfaces have been designed using CSS BOY. Currently, CCDs, PSS (Personel Safety System), MPS (Machine Protection System), Superconductive Cavities, RF Amplifiers, microTCA based LLRF system are being integrated into distributed control system. In this proceeding we summarize the current status and future plans of TARLA control system.

TARLA CONTROL SYSTEM

Control system is one of the main issues. TARLA approach so far was to have sub-systems as independent as possible (including all business logic) with local control system while exposing status read-backs and configuration parameters via TCP/IP interface (slow control only).

TARLA LINAC control will be performed by an EPICS based system operating under the Centos7 operating system on industrial rack mount. EPICS was selected as the main medium due to its commitment to new era accelerators as well as distributed structure of architecture and high performance characteristics. The latest addition to EPICS base, EPICS V4 [1] (EPICS 7 was released in August 2017), was also included in the design goals.

The development environment already used:

- CentOS 7
- EPICS base 3.15
- EPICS V4 4.6
- CSS 4.5.6 (basic)
- MySQL
- Git

Hardware platform for slow control is not finalized. United Electronics hardware platform is being used as IOC for analogue and digital interfaces:

- DNR-6-1G: Compact (3U), 6-slot, rugged, Gigabit Ethernet Data Acquisition and control rack
- UEIPAC 600R: Real-Time, GigE, programmable automation controller (PAC) with 6 I/O slots
- UEPAC 300-1G: Real-Time, GigE, programmable automation controller (PAC) with 6 I/O slots

"In-house" developed TCP/IP based general purposed I/O cards or other custom developed hardware will also be used, while product from other manufactures are also possible.

Standard IOC for devices with TCP/IP interfaces is MOXA DA-682A, x886 2U rackmount computer with 6 gigabit Ethernet ports and 2 PCI expansion slots.

When designing the TARLA Control System, priority and purpose is to create an easy to use/maintain, soft IOC-style, fast, stable and extendable control system suitable for the control and monitoring requirements of all auxiliary systems as well as for laser creation and beam diagnostics.

Architectural Design

Control systems for experimental physics facilities are usually structured in three tiers (see Fig. 1):

User Interfaces. These can be either graphical or non-graphical (command line based) and are usually located in the control room. However, there could be some user interfaces used elsewhere during commissioning and maintenance.

Central Systems and Services. Systems like Timing, Machine and Person safety and computer services that need to run continuously irrespective of user activities, e.g., for archiving acquired data, monitoring alarm states, user authentication services etc.

Equipment Interfaces. This tier is responsible for interaction with equipment and devices. It provides an abstract representation of equipment to higher layers through which the equipment can be monitored and controlled.

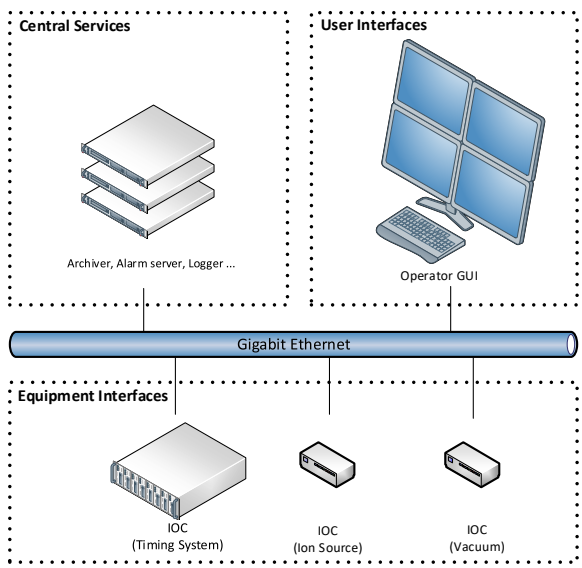


Figure 1: Three tier structure in TARLA control system.

∗ Work supported by Ministry of Development of Turkey
† omer.faruk.elcim@tarla.org.tr

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP23

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

A vertical column (see Fig. 2) is an abstraction which describes how a concrete device is integrated into the control system. It covers everything from the device sensors to the GUI on the operator machine.

It can be viewed from the two perspectives, presented on EPICS example in Figure 2:

- Looking at hardware it follows signal from the sensors/actuators through device controller to the computer running IOC (EPICS server) and finally to the operator's workstation (EPICS client).
- Software vertical column defines operating system(s), device drivers, required EPICS modules, device specific GUIs etc.

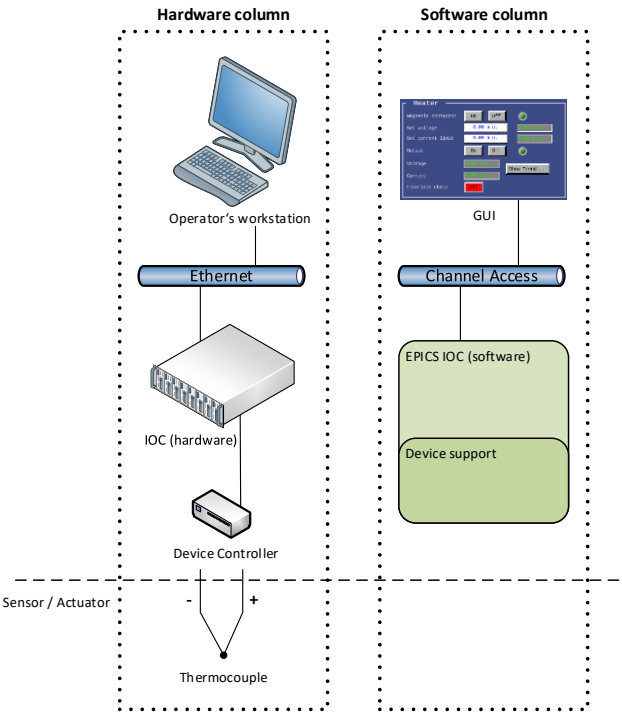


Figure 2: Hardware and software vertical column in EPICS.

Alarm Handler

Alarms are special kind of messages that indicate conditions that need operator awareness and additional actions, supervised by operator. Alarms may range in criticality level.

Alarms are hierarchal structured into levels. Number of levels should be limited and criticality of each level documented, considering the potential effect of the underlying cause of the alarm on operational runtime, equipment damage and safety of personnel.

An alarm server should be responsible for collecting all alarms, storing them in a dedicated database and dispatching them to the operator display.

Most promising alarm server and viewer in EPICS community is BEAST, which is important part of the Control System Studio [2], which is also used in TARLA Control System (see Fig.3).

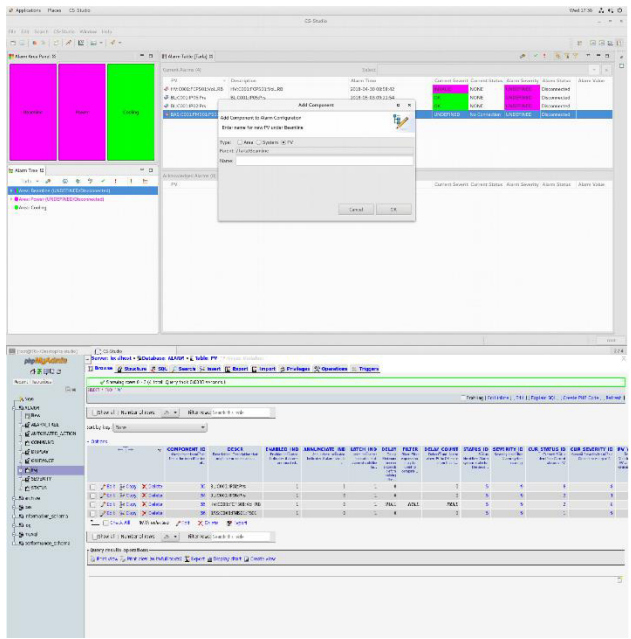


Figure 3: Alarm handler of TARLA control system.

Archive Engine

Providing historical data about the linac operations is up to archiving service. This data is used to evaluate TARLA linac performance, diagnose issues and failures and establish correlations between measurements and several settings for tests. It is important to design of archiving application and the accessibility of the archived data, they will heavily affect machine diagnostic efficiently.

Archiver service for TARLA is EPICS Archiver Appliance (see Fig. 4), which is among others used by ESS and SLAC. It stores data in the binary files on a hard drive, SSD or RAM.

The archiver appliance is written in Java and can run on most common Linux distributions.

One of the main advantages of the Archiver appliance is the built-in support for data storage on short term storage, medium term storage and long term storage.

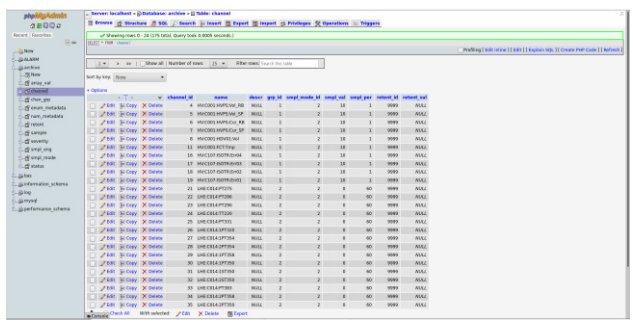


Figure 4: Archive engine of TARLA control system.

Viewer for the archived data is "Data Browser" perspective in the Control System Studio (see Fig. 5). Data can be retrieved from the Archiver Appliance to the Control System Studio with out of the box support.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP23

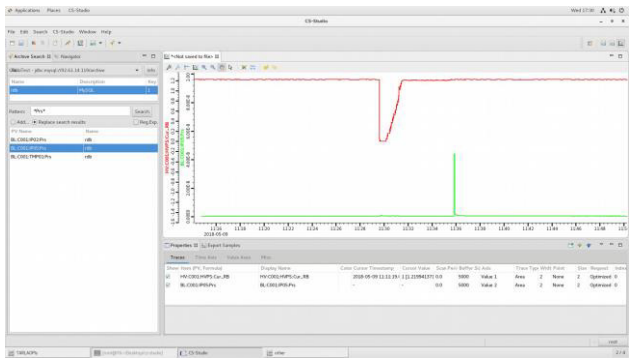


Figure 5: CSS data browser, viewer of archived data.

IocLogServer

Logging provides engineers and developers with insight into the current behaviour of TARLA CS components. It is an important tool for diagnostics when TARLA control system's functionality does not fit with set necessities and allow developers and engineers to resolve problems.

The logging service should collect time-stamped log entries from the control system and store them in a central location, where efficient queries can be performed.

EPICS already provide services for logging such as "iocLogServer" and "iocLogClient" components. Apart from EPICS logs other log files generated by computer nodes (e.g. Linux log files) can also be centrally managed with off the shelf services such a "syslog". Logs provided by the "iocLogServer" are text based.

JAVA-based software for saving IOC's error records to the RDB directory. The modified version of EPICS iocLogServer can read in configuration file in start-up time. This configuration file specifies how the server must process messages from particular IOC (files where messages must be saved, log file limits etc.). Flexible format of configuration file allows to define different methods of message processing. Now three methods are implemented: basic, non-stop and monthly. Each IOC can have its own configuration[3].

OLOG Logbook

Olog is implemented as a REST style web service. Its intended use is as an electronic logbook for accelerator operations [4].

It is a JAVA-based application that enables practical observation of important observations, notes, break-downs.

Control System Studio has an interface for viewing OLOG logs, which can be accessible from path Window > Open Perspective > Other < Log Viewer Perspective (see Fig 6.).

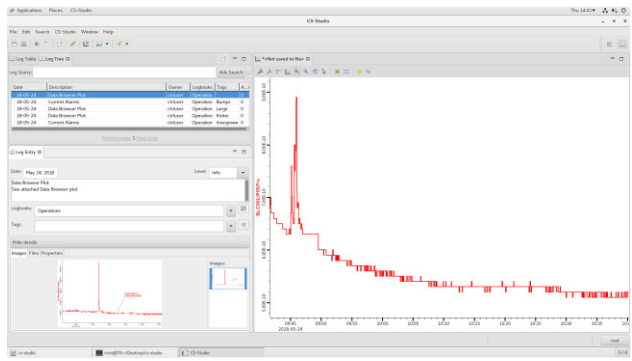


Figure 6: OLOG CSS interface.

CONCLUSION

Control system for TARLA e-gun is capable of producing and controlling the electron beam in DC/CW or macro-pulsed mode at the moment[5].

Device integrations to the EPICS-based control system and infrastructure improvements of the control system continue parallel to the installation of the line. The integration of the injector line control and diagnostic devices of the linac will be completed in the first quarter of 2019.

The purpose of the TARLA team as well as control system group is to build a state-of-art system, fully featured and debugged by the first laser beam generation.

REFERENCES

[1] http://epics-pvdata.sourceforge.net/index.html

[2] http://cs-studio.sourceforge.net/docbook/

[3] http://www-mks2.desy.de/

[4] http://olog.github.io

[5] Kazanci *et al.*, "Current Status of TARLA Control System", in *Proc. IPAC'14*, Dresden, Germany, Jun. 2014, paper THPRO127.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP24

THE STATE MACHINE BASED AUTOMATIC CONDITIONING APPLICATION FOR PITZ

D. Melkumyan*, P. Boonpornprasert, Y. Chen, J. Good, M. Gross, H. Huck, I. Isaev, D. Kalantaryan, M. Krasilnikov, O. Lishilin, G. Loisch, A. Oppelt, M. Otevrel[1], B. Petrosyan, H. Qian, Y. Renier[2], F. Stephan, G. Trowitzsch, G. Vashchenko, DESY, 15738 Zeuthen, Germany

[1]also at CEITEC, Brno University of Technology, 612 00 Brno, Czech Republic
[2]also at DPNC, Geneva University, 1211 Geneva, Switzerland

Abstract

The Photo Injector Test Facility at DESY in Zeuthen (PITZ) was built to test and to optimize high brightness electron sources for Free-Electron Lasers (FELs). In order to achieve high accelerating gradients and long RF pulse lengths in the normal conducting RF gun cavities, an extensive and safe RF conditioning is required. A State Machine based Automatic Conditioning application (SMAC) was developed to automate the RF conditioning processes, allowing for greater efficiency and performance optimization. The SMAC application has been successfully applied to RF conditioning of several gun cavities at PITZ.

INTRODUCTION

The PITZ [1] has been established to develop, test and optimize sources of high brightness electron beams for FELs like FLASH [2] and the European XFEL [3]. Essential requirements of an electron injector for FELs are the ability to generate a reliable electron beam with a very small transverse emittance (e.g. <1mm-mrad, 1nC bunch charge) and a reasonably small longitudinal emittance. The primary goal at PITZ was to facilitate stable and reliable operation with 60 MV/m accelerating gradient at the photo cathode at 650µs RF pulse length and 10Hz repetition rate. To achieve this, a new RF feed system with two RF windows was installed at PITZ in 2014.

The RF Setup

The gun prototype 4.5 conditioning setup consisting of a 10-MW multi-beam klystron, an upgraded RF waveguide distribution system with SF_6 and Air parts, two 10-MW THALES [4] vacuum RF windows, directional couplers, Ion Getter vacuum Pumps (IGP) and a Pressure Gage (PG), photomultipliers (PMT) and electron detectors (e-det) located around the gun coupler is shown in Figure 1.

The Gun

The PITZ photo electron gun is a 1.6 cell normal conducting L-band cavity, with cathode located at the back wall of the half-cell. The electron beam is generated at the cathode by a laser pulse and then accelerated by RF fields and focused by the solenoid fields.

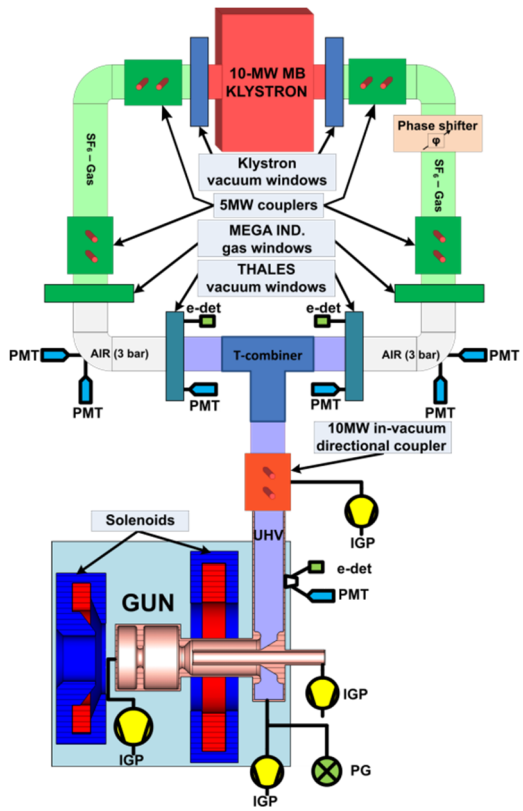


Figure 1: The PITZ RF setup (version 4.5, 2018).

Conditioning

The main goal of the conditioning process is to improve the vacuum in the cavity and the surface condition of the cavity, while applying the RF power is slowly increased in steps up to the operating gradient. During the conditioning process breakdowns may happen with a sharp rise of the pressure in the cavity. In such event the voltage in the cavity is dropped to some initial value and then slowly increased again. The final goal of the gun conditioning is to achieve stable operation at the maximum power level in the gun with solenoid fields.

The State Machine based Automatic Conditioning application (SMAC) was developed to automate the conditioning process for the RF cavities. The application was designed to replicate the operator behaviour during conditioning an RF cavity, allowing for greater efficiency and performance optimization. The interface for the SMAC application has been designed to be user-friendly and intuitive. It provides the user control of the conditioning process and relevant monitoring data.

* david.melkumyan@desy.de

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP24

THE STATE MACHINE BASED AUTOMATIC CONDITIONING APPLICATION

SMAC is written in Java and uses State Chart XML (SCXML) [5-7] as the finite-state machine execution environment based on Harel state-charts [6]. It employs the Distributed Object-Oriented Control System (DOOCS) [8] and Three-fold Integrated Networking Environment (TINE) [9] for the communication with the control systems of PITZ such as RF System, Interlock System [10], Alarm system, Water Cooling System (WCS), etc. The graphical user interface (GUI) is created by using the Java Swing toolkit and available via Java Web Start (JWS) [11] which provides a simple way to launch an application via a network. Communication between GUI and SXML processing layer is performed via Document Object Model [12] (DOM) events. A pure Java design makes the application extremely portable across computing environments. In the end SMAC produces very detailed logs in order to see how the workflow has progressed from step to step and observe any errors that may have occurred. The overall structure and data flow of SMAC application is shown in Figure 2.

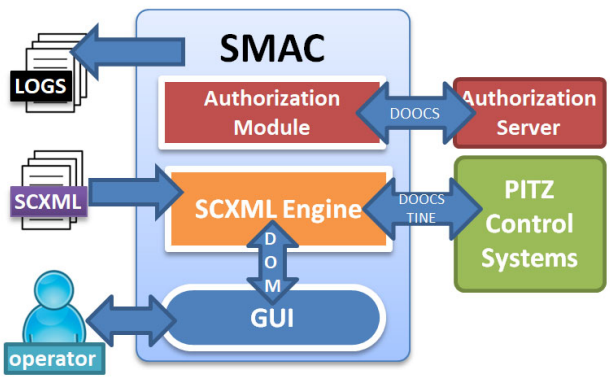


Figure 2: Overall structure and data flow of SMAC.

The Authorization Module

The authorization module with a help of the Authorization server guaranties that only one instance of SMAC is working at the same time. This will prevent concurrent conditioning processes that would lead to application malfunction and even cavity damage.

The SCXML Engine

SCXML engine capable of executing a state machine defined using a SCXML document that describes the application flow. Figure 3 shows a SCXML file segment that defines top-level states of the RF conditioning process. The main advantages of the SCXML approach are listed below:

- Legibility: the whole application flow is easily visible by looking at the flow definition files.
- Flexibility: using only the SCXML document the application control logic can be extracted out of the source code and specify very modular components into an SCXML.
- Standards-based solution: specified by W3C the SCXML is becoming a standard way to represent state charts in XML. It is part of Unified Modeling Language (UML) [13].
- Any SCXML can be easily parsed and interpreted by other software. An SCXML document can be created even using various graphical tools.

```
<state id="rf_configure" src="../conf/rf.config.sc.xml">
<state id="idle_rf_ramping">
<parallel id="process_rf_ramping">
<state id="error_rf_ramping">
<state id="interrupted_rf_ramping">
    <onentry>
        <log label="interrupted_rf_ramping" expr="'RF ramping is in
    </onentry>
    <transition event="resume.rf.ramping" target="process_rf_rampin
    <transition event="terminate.rf.ramping" target="terminated_rf_
</state>
<state id="terminated_rf_ramping">
```

Figure 3: Segment of SCXML code.

The GUI

Figure 4 shows a screen snapshot of the SMACs interface whilst running.

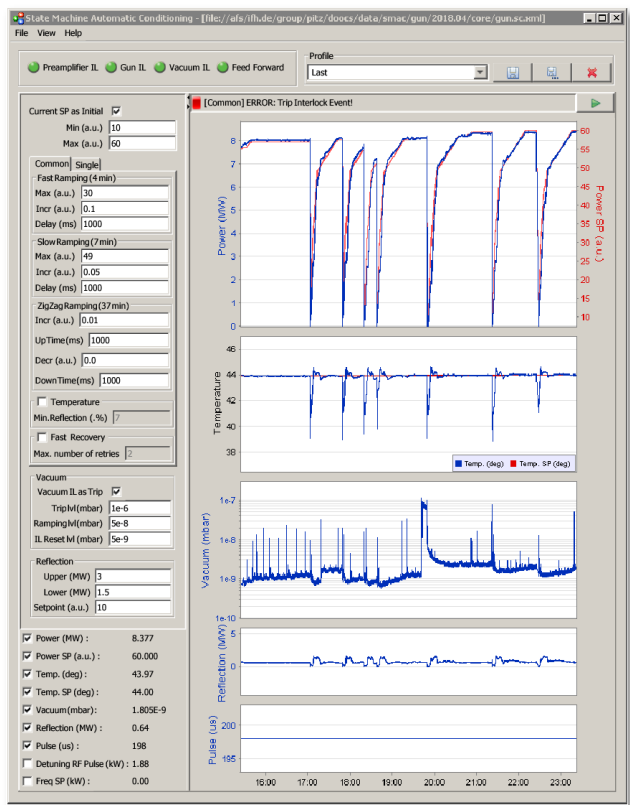


Figure 4: A screenshot of the GUI.

The panel on top-left shows the feed-forward signal and the Interlock system statuses. The profile panel on the top right allows operator to pre-configure the conditioning settings in order to quickly apply them to a new run. The plots on the right present the history of the RF power, water temperature, vacuum pressure, reflected power and the RF pulse length. The bottom left displays the current values of monitored parameters. Parameters of the condi-

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP24

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

tioning procedure can be set at the middle left of the screen. The main parameters set by the user are:

- RF power range and ramping speeds.
- Vacuum upper/lower thresholds.
- Reflected power thresholds.
- Enabling WCS temperature control.
- Enabling "Fast Recovery" [14] procedure after each breakdown.

Default values for these parameters were set based upon prior experience of the operators.

CONDITIONING ALGORITHM

The conditioning algorithm consists of gradually increasing the RF power and the RF pulse length but keeping a low rate of vacuum spikes in the cavity in order to prevent any damage from breakdowns. The algorithm based on conditioning requirement of the THALES windows and adopted for the gun cavity conditioning. It follows simple rules:

- RF pulse length: 10 – 650μs.
- Vacuum pressure: below 10^{-7} mbar.
- RF power ramping speed: maximum 0.2MW every 15 minutes for each new RF pulse length, when the RF window has seen this power level the first time.
- In the case of significant vacuum spikes or breakdowns:
  - Reset the RF power to initial value.
  - Reset pulse length to minimum value (10μs).
  - After reaching the maximum RF power, increase the pulse length in reasonable steps.
- Initially, the RF gun solenoid is off.
- Achieve the maximum RF power level with a negligible breakdown rate (aim for less than one breakdown per week).
- Finally repeat the conditioning procedure with sweeping RF gun solenoid currents.

However such a conditioning algorithm might be complicated since it involves simultaneous monitoring and controlling various operating parameters (e.g. interlock/feedforward/feedback statuses, RF power, gun temperature, RF frequency, etc.). Figure 5 shows a segment of the RF ramping flow diagram.

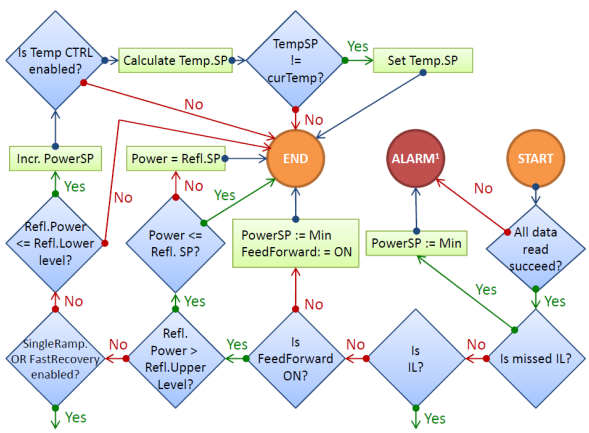


Figure 5: Segment of the RF ramping flow diagram.

The current version of the SMAC application implements two ramping modes:

1. Single mode: based on the "Fast Ramping" algorithm [14]. Thus the RF frequency is changed to follow the resonance frequency and the temperature of the gun cavity is continuously adjusted for slightly overheated operation. This process is continued until a certain power level is reached or a breakdown occurs.
2. Common (or continues) mode is shown in Figure 6: RF power is steadily increased until a significant vacuum spike or until breakdown occurs. The common mode consists of several stages ("fast", "slow" and "zigzag") with different RF ramping speeds. In case of relative high level of pressure in the cavity or high level of RF reflected power:
   - Suspends increasing of RF power until suitable ramping conditions.
   - Decreases RF power to initial value, if reflected power or pressure in the cavity is too high.
   - After each breakdown SMAC attempts to reach the last power level applying the Fast Recovery procedure if enabled.

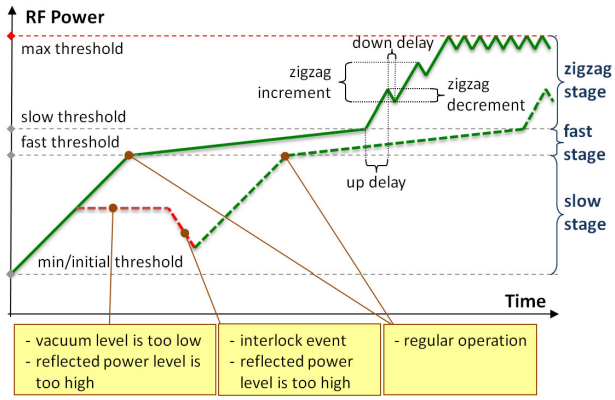


Figure 6: Common mode of the RF power ramping.

CONCLUSION

In this paper, we have presented the main design features and the current implementation status of the SMAC application. This implementation was intended as a proof of concept, applying a state-chart approach toward the development of the automatic conditioning application. The state-chart approach has been found particularly useful because of its appealing hierarchical, communication and concurrency constructs. The use of Java SCXML engine was very encouraging and we intent to explore the use of it with advantages offered by the Java technology. The SMAC application was brought into operation in 2010 and has been used at PITZ very successfully for all RF cavities. Since then the application is left to run unattended overnight. The RF conditioning strategy is based on recommendations from physicists of the PITZ group. The SMAC continues to be improved by feedbacks and suggestions from the physicists.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP24

REFERENCES

[1] PITZ, "Photo Injector Test Facility at DESY," http://pitz.desy.de

[2] W. Ackermann *et al*., "Operation of a free-electron laser from the extreme ultraviolet to the water window," Nature Photonics, vol. 1, pp. 336-342, 2007.

[3] M. Altarelli *et al.,* "The European x-ray free-electron laser technical design report," DESY, Hamburg Report No. DESY 2006-097, 2007.

[4] Thales Group, 45 rue de Villiers, 92526 Neuilly-sur-Seine, Cedex, France www.thalesgroup.com

[5] W3C, "State Chart XML (SCXML): State Machine Notation for ControlAbstraction," http://www.w3.org/TR/scxml

[6] SCXML, "The Jakarta Project Commons SCXML," http://jakarta.apache.org/commons/scxml

[7] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, June 1987.

[8] DOOCS, "DOOCS is a distributed control system," http://doocs.desy.de

[9] TINE, "Three-fold Integrated Networking Environment," http://tine.desy.de

[10] M. Penno *et al.,* "A Configurable Interlock System for RF Stations at XFEL," in *Proc. PCaPAC'08*, Ljubljana, Slovenia, Oct. 2008, paper WEZ03.

[11] Java Web Start, "Java Web Start Overview," http://www.oracle.com/technetwork/java/javase/overview-137531.html

[12] DOM, "Document Object Model Events," https://www.w3.org/TR/WD-DOM-Level-2/events.html

[13] UML, "Introduction to OMG's UML," http://www.uml.org/what-is-uml.htm

[14] Y. Renier *et al.,* "Fast Automatic Ramping of High Average Power Guns," in *Proc. IPAC'17*, Copenhagen, Denmark, May 2017. doi:10.18429/JACoW-IPAC2017-TUPIK052

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP25

EVOLUTION AND CONVERGENCE OF ALARM SYSTEMS IN TANGO

S.Rubio-Manrique, G.Cuni, F.Fernandez, R.Monge
ALBA Synchrotron, Cerdanyola del Vallès, Barcelona, Spain
G.Scalamera, Elettra-Sincrotrone, Basovizza, Trieste, Italy

Abstract

The technology upgrade that represents Tango 9 has triggered the evolution of two of the most used Tango tools, the PANIC Alarm System and the HDB++ Archiving. This paper presents the status of the collaboration between Alba and Elettra Synchrotron sources for the convergence of its both alarms systems under the IEC62682 [1] standard, and the usage of HDB++ tools for logging and diagnostic of alarms. Relevant use cases from the user point of view has been added to the paper as a validation of the benefits of this control system evolution.

INTRODUCTION

Alarms in Control Systems

Alarm Systems have been a common part of control system toolkits for decades. In the Synchrotron community some of the most common tools are PANIC [2] and AlarmHandler [3] for Tango Control System [4], as well as BEAST Alarm System for EPICS.

PANIC and AlarmHandler systems have coexisted within the Tango community for years, but at some point new members of the community asked whether to choose one or the other for their specific domain. This question triggered an effort to compare both systems, extract the best features of them and explore how they could complement each other.

This effort required from us to redefine what an Alarm System was, and what it was expected to do.

What's an Alarm System?

SKA and Elettra institutions proposed to the Tango community to adopt a common terminology and behaviour based on an international norm, the IEC 62682:2014 "Management of Alarm Systems for the Process Industries"[5-6]. The norm states that:

- The primary function of an Alarm System will be to notify abnormal process conditions or equipment malfunctions, and support the operator response.
- The Alarm System is NOT part of the protection nor safety systems, which must have separate tools following its own regulation.
- The Alarm System is part of Operator Response, thus it's part of the HMI (including the non-graphical part of it).

These three statements clarified what our Alarm Systems were expected to do; providing a common ground to start the collaboration on merging both projects.

Alarms within the Tango Control System

It is needed to introduce several concepts on how alarms are developed within a Tango Control System. The following terms describe the object hierarchy:

- Tango Host or Database: the central database where all devices are registered and configuration stored.
- Device Server: Each independent software process distributed across the system, managing one or several Tango devices.
- Device: a Tango Device is an entity that can be typically identified to a hardware piece or software process (e.g. a vacuum pump, a PLC, a motor, a voice synthesizer). Each device exports to the control system its Attributes (process variables), Commands (actions), Properties (for configuration) and States.
- Attributes: Each of the process variables exported by a Tango Device. They support both synchronous or asynchronous reading and writing. At each attribute reading it exports its value, timestamp and quality.
- Attribute Quality: The attribute quality accompanies each value to express the process conditions. Quality can be VALID, INVALID, WARNING or ALARM and it can be set on runtime by a Tango user specifying which ranges of operation/warning/alarm will trigger a quality change.

Quality-based vs Formula-based Alarms

The most primitive scope of Alarm Systems in Tango just included the logging of those attributes qualities in ALARM. But this approach didn't apply when it was required the interaction between multiple devices.

The Tango Alarm System (AlarmHandler device server and its alarm database) was developed [3] to enable logical and arithmetical operations on attribute values, thus extending the alarm triggering from the simple matching of an attribute value within valid ranges.

PANIC [7] was developed by ALBA Synchrotron in 2007 [8] as a Python [9] alternative to the Tango Alarm System. It was based on similar principles but applying a distributed architecture (see Fig. 1) and trying to add annunciator features and more flexibility [10] in alarms declaration, allowing to execute python code for both the formula and the resulting action [11]. This enabled the usage of wildcards for attribute selection, and reusing the data from the alarm evaluation to generate rich-text emails or SMS messaging.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP25

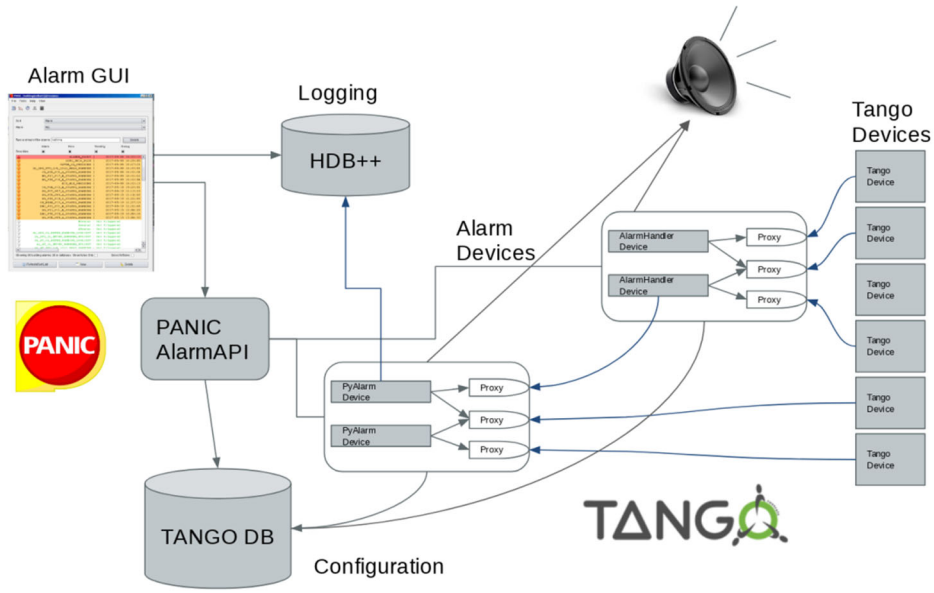


Figure 1: PANIC Architecture, showing alarm device servers, configuration and logging databases, and annunciators.

ALARM SYSTEMS CONVERGENCE

Existing Alarm Systems in Tango have been already described in previous papers [2-3], as well as the changes required [12] to adapt them to the IEC62682. This paper focus instead in the changes that allowed their convergence into a single toolkit.

Unifying the AlarmHander and PANIC device servers forced us to unify criteria on:

- Alarm Formula database
- Alarm attributes
- Classification of alarms
- Annunciators specification and triggering
- Priorities
- Exporting attributes to HMI
- Alarm Summaries

The criteria for convergence have been triggered by the chosen norm (that fixed terminology and state conditions), the Tango control system architecture (centralized database / distributed devices) and the possibilities to expand the HMI to the web (choosing JSON as the default data format for Alarm Summaries).

Intended to reduce the number of different tools required for having a functional alarm system, the Alarm database has been dropped to use Tango Properties instead. Keeping the alarm formulas and configuration in the Tango database helps to unify the quality-based and formula-based alarm approaches, thus storing the configuration for both types of alarms in the same database.

Following the same principle, it has been discarded the idea to maintain a separate Alarm logging, thus reusing the new HDB++ archiving for Tango [13].

HDB++ archiving allows to record alarms on state change, and at the same time records all the events received by the attributes involved in the formula (including both value and quality changes). Thus, alarm changes and attribute changes can be queried and represented from the same database using already existing tools.

ALARMS IMPLEMENTATION

Reactive Alarms vs Polled Alarms

The PANIC API is currently connecting to two types of device servers, PyAlarm, developed in Python by ALBA Synchrotron, and AlarmHandler, developed in C++ by Elettra Sincrotrone.

Both device servers acquire lists of formulas from the Tango Database Properties, connect to the attributes appearing on the formulas and evaluate the results on each attribute value change; triggering actions when required by an alarm state change. Devices differ on how they connect to attributes and react to value change.

PyAlarm is polled-based, so it means that attributes are read periodically, updating the formula result at each attribute reading. Attribute changes are cached and alarm is not triggered until the result is True for a number of iterations (the AlarmThreshold property). With a threshold of 1, response time can be as short as 100 ms; but still this minimal latency time is needed. In exchange, complex formulas are enabled and the polling buffer allows to use the average, the max peaks or the delta change of the last N values acquired.

AlarmHandler instead is event-based. It means that the device is just waiting for events sent from the Tango Control System, reacting immediately when a change event is received. In this case, latency is minimal, and actions are executed immediately. Alarm formula parsing is done in C++, and it is restricted to basic arithmetical and logical operations between attribute values.

As demonstrated in the field, both systems are complementary, as the AlarmHandler can be dedicated to fast reaction on critical conditions while PyAlarm flexibility

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP25

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

provides its best usability on interacting with multiple devices or evaluating attribute evolution in time.

Alarm System Scalability

Another factor that triggered convergence of Alarm Systems in Tango has been the need to increase system scalability and performance for the new SKA project.

Alarms Systems can be scaled using different approaches. PANIC allows scaling by exporting each evaluated alarm as a new attribute. Thus, new alarms can be written using the result of the previous ones already evaluated. It allows having a detailed alarm for each subsystem of a sector, and then summarize all values in a single sector alarm. As example, vacuum alarms will contribute to the alarm state of a sector, and at the same time are grouped in a vacuum alarm for all sectors.

Alarm summaries can be done at client level (Alarm View) or at device server level (Alarm Group). When created at alarm level, it will behave as any other alarm, triggering actions and exporting a new attribute, so the system can be scaled to the next level. They can be accessed even from other Tango Control Systems, so the hierarchy can be expanded indefinitely. Propagation times at each level can be tuned using the AlarmThreshold and PollingPeriod properties.

Alarm Annunciators

Besides its initial features (email, SMS, logging), the current version of PANIC allows now to trigger any kind of Tango command or python code script and provides the capability of reusing alarm data and values in the execution of these commands. As example, it allows to send alarm description to speakers, include attribute values in an email or html report, move a motor a fixed number of steps depending on a calculation, etc.

It also expands the logging mechanisms. HDB++ and the Tango DB are used to keep history of alarm and attribute changes and its configuration; but via external commands other messaging or logging systems can be easily integrated. It allowed PANIC to interact with some popular applications like Telegram or Kibana and generate web reports based on JSON files [14].

CONCLUSIONS AND FUTURE

PANIC is a key tool for the daily operation of ALBA Synchrotron Accelerators and Beamlines, it is the first diagnostic tool used in order to check any incidence. It offers an overview of currently triggered alarms, allowing a full comprehension of the problem. Its second potential resides in the customization of the alarms applications that allows to suit the specific needs of each user group and, in its integration with Taurus [15-16], eases a fast response and troubleshooting in a user-friendly environment.

Having a common interface with Elettra's AlarmHandler allow to expand PANIC functionality to a new level, as now two complementary Alarm engines are provided and scalability can be achieved combining performance and flexibility at each level.

PANIC Use Cases

This are some practical examples of PANIC use cases that have been enabled by the changes introduced in the latest releases:

Protection of infrared mirrors; in this case an alarm was required to, in case of a fast increase of temperature, to be able to move a mirror outside of the vacuum chamber in less than 200 ms. For this application alarms on delta change, response to events and precise motor movement execution were required.

Injection Permits: In the last upgrades of our Synchrotron facility, the number of injection modes are increasing as well as the complexity of the rules that allow to inject in the storage ring. PANIC has been used as an easy tool to compile permission rules and convert it into attributes that can be used as enable/disable of different operations.

Vacuum Systems: it is very important the continuous monitoring of the pressure and equipment state, but besides this, there are many variables belonging to other sub-systems which could affect the vacuum status. PANIC allows to extend the views and notifications of each alarm to include additional information or parameters that help to diagnose the problem. It also allows to crosscheck temperatures against different ranges depending on the current operation status (injection, maintenance, bakeout); checking not only the value but its proper behaviour.

Testing and Deployment

Testing of PANIC is critical in to ensure its reliability and scalability, guaranteeing that any modification in the evaluation engine does not break the compatibility with previous versions. A test suite has been developed [17] and its currently expanded with the help of tango-simlib project. Testing will be introduced in our continuous integration/deployment cycle based on Debian packages.

ACKNOWLEDGEMENT

The latest release of PANIC contains contributions from many people: MaxIV, ESRF, ALBA, Elettra and Solaris institutes as well as some private companies like S2Innovation, IK and TCS.

The work of validating and suffering each new release is shared with our users, the Operation and Vacuum groups at ALBA. They contributed with their experience to refine and improve alarm formulas and syntax.

REFERENCES

[1] *Management of alarms systems for the process industries,* IEC-62682, IEC, 2014.

[2] S. Rubio-Manrique et al., "PANIC, A Suite for Visualization, Logging and Notification of Incidents.", in *Proc. PcaPAC'14*, Karlsruhe, Germany, Oct. 2014, paper FCO206.

[3] L. Pivetta, "Development of the Tango Alarm System", in *Proc. ICALEPCS'05*, Geneva, Switzerland, Oct. 2005, paper WE3b.1-70.

[4] TANGO, http://www.tango-controls.org

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP25

[5] *Engineering Equipment and Material,* Users'Association (EEMUA) issued publication, 191, University of Manchester, United Kingdom, 1999.

[6] M. Tennant, "Implementing Alarm Management Per the ANSI/ISA-18.2 Standard", *Control Engineering*, September 2013.

[7] PANIC, https://github.com/tango-controls/panic

[8] S. Rubio-Manrique et al. "Extending Alarm Handling in Tango", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, paper MOMMU001.

[9] D. Fernández et al. "Alba, a Tango based Control System in Python", in *Proc. ICALEPCS'09*, Kobe, Japan, Oct. 2009, paper WEPMU005.

[10] S. Rubio et al., "Dynamic Attributes and other functional flexibilities of PyTango", in *Proc. ICALEPCS'09*, Kobe, Japan, Oct. 2009, paper THP079.

[11] http://www.pythonhosted.org/panic/recipes.html

[12] G. Scalamera, L.Pivetta, S.Rubio-Manrique, "New developments for the Tango Alarm System", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, paper TUPHA165.

[13] L. Pivetta et al., "New developments for the HDB++ Tango Archiving System", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, paper TUPHA 166.

[14] M. Broseta, D. Roldan, S. Rubio, A. Burgos, G. Cuni, "A web-based report tool for Tango Control Systems via websockets", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, paper TUPHA 173.

[15] S. Rubio et al., "Unifying all Tango Services in a single Control Application", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015. paper WEPGF148.

[16] Taurus Library, http://www.taurus-scada.org

[17] S. Rubio-Manrique et al., "Reproduce Anything, Anywhere: A Generic Simulation Suite for Tango Control Systems", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, paper TUDPL01.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP26

THE DEVELOPMENT OF VACUUM GAUGE MONITORING AND CONTROL SYSTEM USING Web2cToolkit

H. Ishii †, T. Kosuge, H. Nitani
High Energy Accelerator Research Organization(KEK), 1-1 Oho, Tsukuba, Ibaraki 305-0801, Japan

Abstract

The Photon Factory is an accelerator-based light source facility that is a part of the High Energy Accelerator Research Organization (KEK) in Japan. At the Photon Factory, a message transferring software named Simple Transmission and Retrieval System (STARS) is used for beamline control. STARS is suitable for small scale remote-control systems using TCP/IP sockets and works with various types of operating systems.

STARS is applicable to various control systems; we developed the vacuum gauge monitoring and control system with STARS in this work.

Web2cToolkit is a framework developed by Deutsches Elektronen Synchrotron (DESY) that provides a user-friendly human machine interface, and developers can easily create web-based graphical user interfaces (GUIs) with Web2cToolkit.

Web2cToolkit supports various types of protocols (e.g., TINE, DOOCS, EPICS, TANGO), including the STARS protocol. We adopt Web2cToolkit as a framework for a front-end GUI application of our vacuum gauge monitoring and control system. Although the application development is still in progress, some features have already been implemented.

OVERVIEW OF STARS

Simple transmission and retrieval system (STARS) [1] is composed of at least one server program "STARS server" and one client program "STARS client". Each STARS client is connected to the STARS server through a unique node name and handles text-based messages through a TCP/IP socket (Fig. 1).

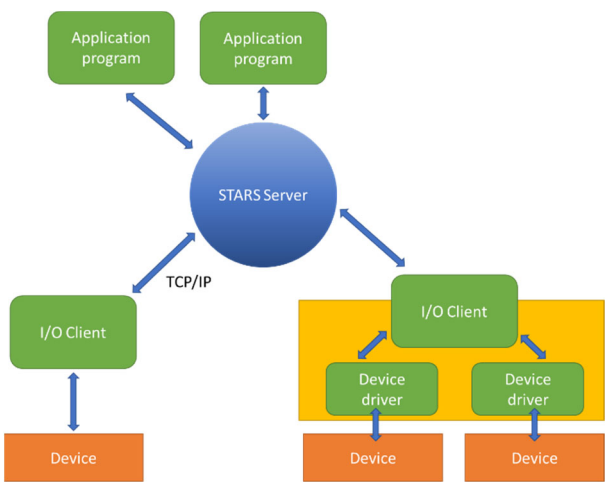


Figure 1: Communication between STARS clients and a STARS server.

The STARS server program is written in Perl. Therefore, it can be executed on different operating systems, such as Windows, MacOS, and Linux. STARS has been introduced as a beamline control system in more than 20 beamlines of the Photon Factory. In addition, an interface program of the beamline interlock systems has been developed using the STARS software.

VACUUM GAUGE CONTROLLER

A large number of vacuum gauges, exceeding 300, are installed in the beamlines at the Photon Factory. These vacuum gauges are connected to various types of controllers that monitor the degree of vacuum of the beamlines. Each controller has a relay output port that either completes or disrupts an electric circuit according to the degree of vacuum, and this port is used for the beamline interlock system. The vacuum environment of the beamlines at the Photon Factory is managed by this beamline interlock system (Fig. 2). The thresholds for these relays can be configured using the switches on the front panels of the controllers or through their respective computer interfaces.

Most of the vacuum controllers have computer interfaces for monitoring the degree of vacuum and operating the controller. Data can be obtained or set through these interfaces and include the following:

- Getting the current value of vacuum;
- Turning the sensor device on or off;
- Setting and Getting the thresholds of the relay output function.

The interface of each vacuum controller has similar functions; however, the different types of communication protocols depend on hardware type or manufacturer.

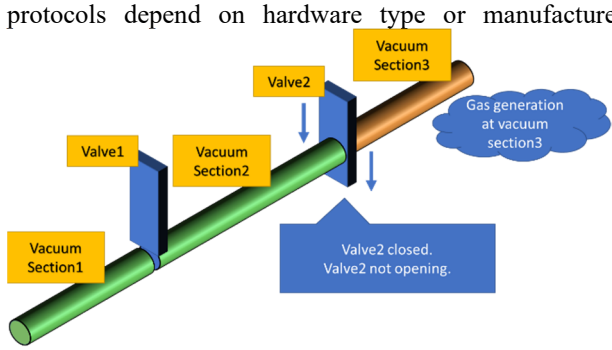


Figure 2: Beamline interlock system and vacuum monitoring.

VACUUM MONITORING SYSTEM

The beamline interlock system uses only the relay output functions; therefore, it is impossible to protect the beamline before a vacuum hazard occurs. The vacuum hazard will interrupt all experiments via leaked synchrotron

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP26

radiation, so it is important to detect abnormal situations even if is the changes may be insignificant. A vacuum data monitoring system has many benefits, such as prediction of vacuum-related issues of the beamline components.

Therefore, we developed the vacuum monitoring and data logging system shown in Fig. 3. We installed STARS on the vacuum gauge monitoring system, and all vacuum gauge controllers were connected to the STARS server through a TCP/IP socket with a serial interface device server.

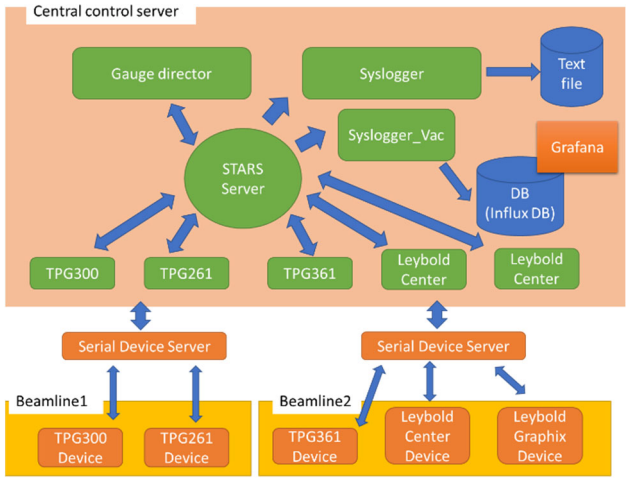


Figure 3: Vacuum monitoring system.

To transmit and receive messages between each vacuum gauge controller and the STARS server, a client soft-ware was developed, and the following client program was installed on each vacuum gauge controller (Table 1). Each client software instance has a unique identifier that consists of the beamline and gauge controller names (e.g., "bl1ga1," where "bl1" is the beamline name and "ga1" is the gauge controller name).

Table 1: List of Vacuum Gauge Controllers Used in Photon Factory

Controller Type	Computer Interface	Manufacturer
TPG300	RS-232C	Pfeiffer Vacuum, GmbH
TPG261	RS-232C	Pfeiffer Vacuum, GmbH
TPG361/362	USB	Pfeiffer Vacuum, GmbH
Canter series	RS-232C	Leybold Vacuum, GmbH
Graphix series	RS-232C	Leybold Vacuum, GmbH

DEVELOPMENT OF SOFTWARE

We developed individual STARS client software for the gauge monitoring system, and each client software has different functions, which are described below.

Vacuum Gauge Control Client

Typically, the original commands of the vacuum gauge controller are not compatible for another type of controller. This is inconvenient for developers because the soft-ware requires modification in accordance with the type of controller. To resolve this problem, the client software has command converters between the original commands and the unified commands, which are defined for other STARS client softwares. For example, some common unified commands are as follows: "Get-Value," "SetSFunction," "GetSwFunction," "SPS," and so on. Using these commands, we can obtain information such as vacuum data, device setting data, and device condition.

Gauge Director Client and Data Logger Client

A gauge director client automatically sends commands according to the configuration file, similar to the Cron utility in a Unix system. The configuration parameters are the sending interval, transmission command, and target node name. The gauge director client sends the "GetValue" command to all vacuum gauge control clients every 30 s in our vacuum monitoring system. The data logger client creates a log file for all STARS communication messages, including vacuum information, and sends the log data to an external database server. The log file is a text-based archive, and InfluxDB is used as the external database system.

Graphical Interface

Grafana is an open source software for data visualization. We adapted Grafana to visualize vacuum logging data, which are recorded in the InfluxDB server. The log viewing interface is called "Dashboard" and can be easily edited on using a web-browser. Grafana has different kinds of graphical components such as graphs, tables, and so on. We used a graph component to display the time variation of the degree of vacuum in the beamlines (Fig. 4).

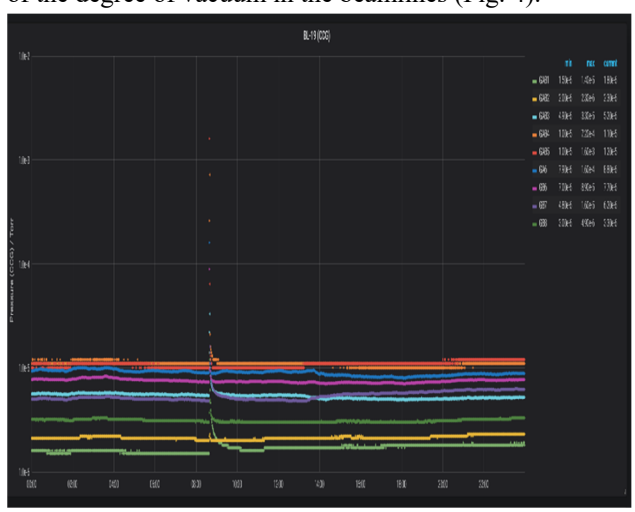


Figure 4: Example of "Dashboard" created with Grafana.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP26

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

STARS FOR WEB2C

The Web2c viewer is a configurable, browser based, dynamic, and bidirectional internet application for live displays (Fig. 5).

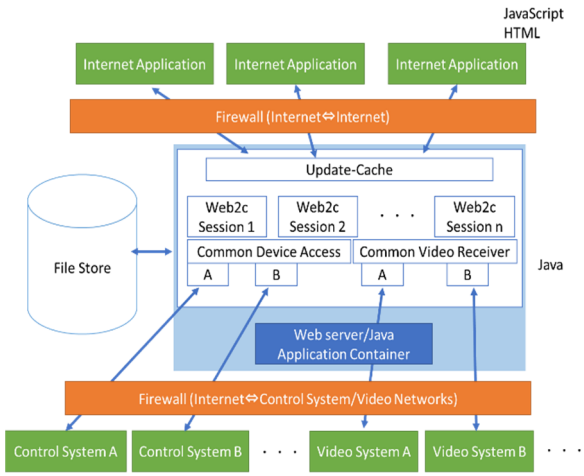


Figure 5: Web2c viewer.

Each application page is easily created using the Web2c viewer wizard or by directly editing the XML file. We created the application page to obtain the information from the beamline vacuum gauge controller using Web2c viewer wizard. The application page consists of various components listed in a corresponding configuration file [2]. The properties of the components are tooltip text, font size, arrangement, signal source URI, and so on. We used the "Web2cViewerText" viewer component in this case (Fig. 6). Currently, only a few Web2c viewer components are available on STARS. For example, "Web2cViewerButton, " "Web2cViewerLabel, " and "Web2cViewerText."

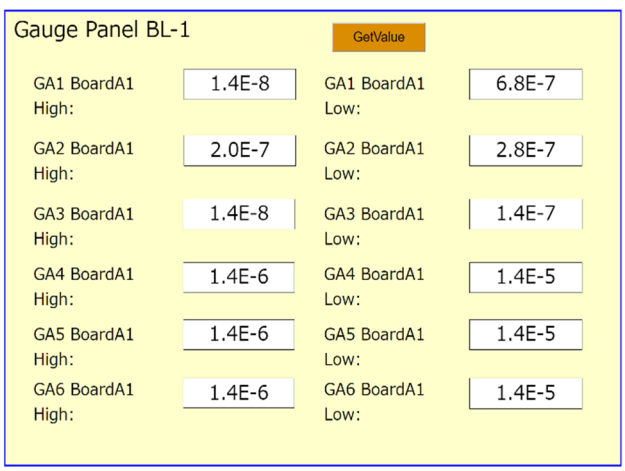


Figure 6: Web2c application page for the vacuum gauge controller.

In Fig. 6, the threshold values of the relay function of the vacuum gauge controller, which is obtained through STARS, are displayed in the text box. When the "GetValue" button is pressed, the threshold values are updated to present values. The Web2c application communicates with STARS via the "STARS proxy." The STARS proxy is the communication protocol between Web2c and the STARS client, and it converts the Web2c message to the STARS message of the gauge controller client. Both Web2c application and STARS proxy are connected to the STARS server as a STARS client (Fig. 7).

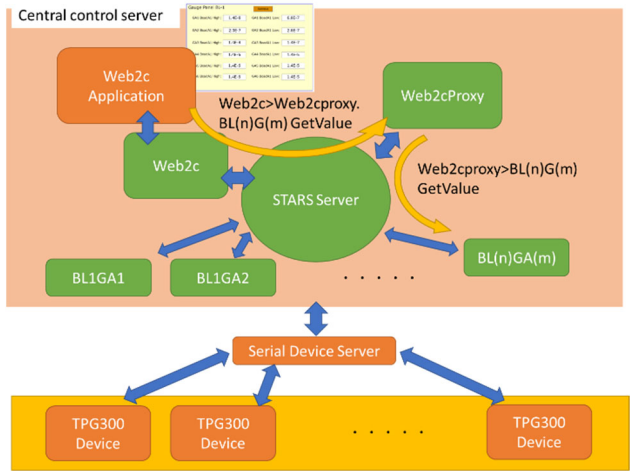


Figure 7: STARS server for Web2c.

CONCLUSION

We constructed a vacuum monitoring system in conjunction with STARS and Web2c. This system supports various types of vacuum gauge controllers. These vacuum gauge controllers connect to the STARS server through a TCP/IP socket and a have unified command system. The vacuum data thus recorded can be displayed using graphs on a web browser. Web2c is a browser-based internet application that can obtain the threshold of a vacuum gauge controller through STARS. We are refining the STARS client program, which is associated with the Web2c framework, and intend to advance the development of other Web2c viewer components through STARS.

REFERENCES

[1] STARS, http://stars.kek.jp
[2] Web2c Toolkit Framework for Web-based Controls Clients, http://adweb.desy.de/mcs/web2cToolkit/web2c_home.htm

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP27

DEVELOPMENT OF THE MALFUNCTIONS DETECTION SYSTEM AT VEPP-2000

O. S. Shubina†, A. I. Senchenko
BINP SB RAS and Novosibirsk State University, Novosibirsk, Russia

Abstract

In 2007, the creation of the electron-positron collider VEPP-2000 (Fig. 1) was completed at the Institute of Nuclear Physics of the SB RAS. The VEPP-2000 collider facility consists of various subsystems, and a failure of any subsystem can lead to the incorrect operation of the complex for several hours or even days. Thus, there is a need to create software that will warn about possible malfunctions. To accomplish the task, software was developed consisting of three modules. The first performs automatic verification of compliance data obtained from the accelerator complex and rules describing the correct subsystems operation. The second module is a user-friendly web interface that displays information about the state of the complex in a convenient way. The third module acts as some intermediary between the first and the second. It processes messages arriving at the message queue and redirects them to all subscribed clients via the web socket. This article is devoted to the development of test software, that is currently running on the VEPP-2000 control panel.

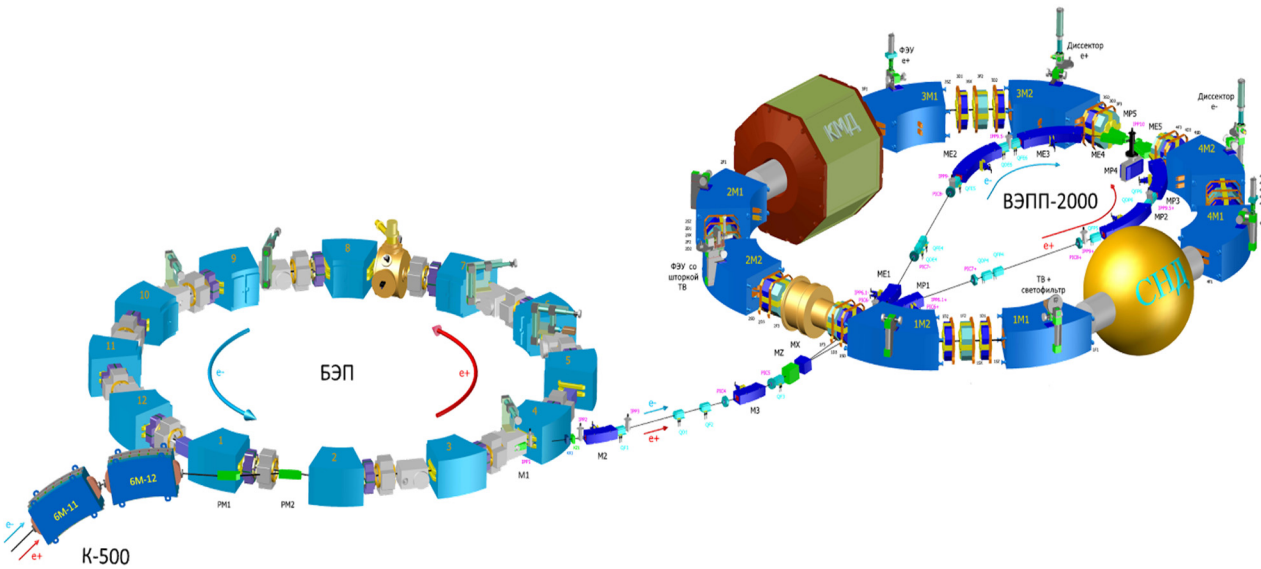


Figure 1: VEPP-2000.

INTRODUCTION

The VEPP 2000 is an electron-positron collider, that was commissioned at the Budker Institute of Nuclear Physics. The VEPP-2000 acceleration complex consists of a few main subsystems: BEP booster ring and VEPP-2000 collider ring. Beam diagnostics is based on 16 optical CCD cameras that register the synchrotron light from either end of the bending magnets and give the full information about beam positions, intensities and profiles. In addition to optical BPMs [1], there are also 4 pick-up stations in the technical straight sections and one current transformer as an absolute current meter. The VEPP-2000 control system [2] has a complex structure and consists of about 4000 channels for monitoring and control. This complicates the timely detection of incorrect operation of the complex or any malfunctions. To solve this problem, software was developed consisting of various parts. The core of the software is a troubleshooting module, that performs automatic validation of rules stored in special configuration files. The following component is responsible for delivering information about the status of the complex to subscribers via the web sockets. Also for the timely notification of new subscribers about the state of the complex uses caching of the state snapshot in the Redis database. Finally, the graphical interface provides a convenient view of all the necessary information.

MODULE FOR CHECKING THE STATE OF THE COMPLEX

The hardware part of the VEPP-2000 accelerator complex has a complicated architecture; therefore, a group system has been developed for the convenience of troubleshooting. Each group is a collection of several channels or

† olgashubina2011@gmail.com.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP27

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

subsystems of the complex, united according to some principle. Thus, during processing, the system operates directly with these groups, rather than with individual channels. Information about groups is stored in a special configuration file. After initialization of all groups and obtaining the necessary data (point 2), the process of troubleshooting is started in accordance with the rules specified in the configuration file. First of all, the system creates a set of objects corresponding to each individual group and storing the information about the rule that is applied to this group, critical and boundary values and a reference to the corresponding data buffer. Each object is an instance of a specific class in which these or other troubleshooting rules are implemented. Then, with a periodicity, a cyclical group check is performed for the presence of malfunction, as well as for the activity of the data source. After the check, a report on the complex status is generated and sent to the message broker RabbitMQ.

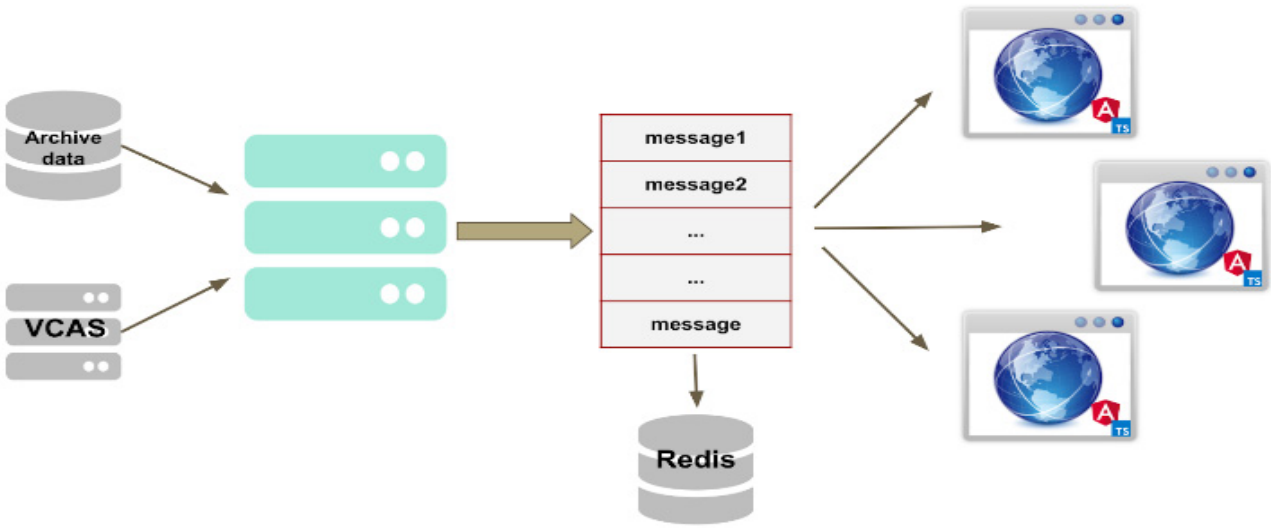


Figure 2: General scheme.

CREATE AND UPDATE DATA BUFFER

Before starting the troubleshooting process, a data buffer is created in the system. Data enters the buffer from the database through a specially developed API for interacting with the accelerator complex archive data. The buffer is filled with data on all the necessary channels for a certain period of time, that is specified in the configuration file groups.

After creating the buffer, it starts the automatic update process. The system subscribes to the VCAS server to obtain online data. Data storage is organized in the form of a ring buffer, i.e. each time, when new data is received, they are saved to the buffer, and the data that is outdated is deleted from the buffer.

The main problem that arose when designing this software is the use of memory. There was a need to create a common memory buffer with which two processes will work in parallel: automatic buffer update and troubleshooting process. At the same time, the solution had to have good performance and be easily incorporated into the developed software and the management system as a whole. This problem was solved by queuing in distributed memory from the multiprocessing library of Python.

MESSAGE BROKER AND COMPLEX STATUS SNAPSHOT

For the distribution of information between customers, a special service, developed on the basis of the message broker RabbitMQ [3], is responsible. It is a web socket server. Any user or software can subscribe this server and receive information on the state of the complex with a certain periodicity.

After passing the next check iteration, a report on the state of the complex is generated and sent to the message broker. Then, the report is processed and sent to all existing subscribers. To reduce traffic and the load on the system, information about the state of the complex is not sent at every iteration of the test, but only when the state changes. Therefore, there may be a problem that the new subscriber does not receive information about the state of the complex for a long time.

To solve this problem, it was decided not only to transmit information online, but also to save the state cast in-memory data structure store Redis. The storage scheme of the state cast has the structure shown in Fig. 2. And now, when adding a new subscription, it will read up-to-date information from the repository, and then receive updates via the web socket when the state changes.

Each author should submit the PDF file and all source files (text and figures) to enable the paper to be reconstructed if there are processing difficulties

MALFUNCTION DETECTION METHODS

Currently, several methods of troubleshooting have been developed to test this system (Fig. 3), in part for current

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP27

and vacuum systems. To troubleshoot current systems, three simple rules are used:

- check of the equality of the set current to the specified parameter
- check of the difference between the set and obtained value
- the standard deviation of the normalized value

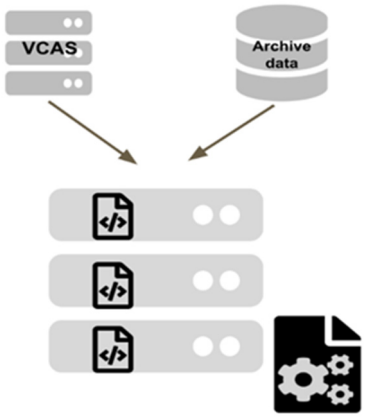


Figure 3: Malfunction detection scheme.

The second fault finding algorithm was developed for those subsystems in which measurements are rarely made to detect a malfunction in time. And in this case, the algorithm performs:

- interpolation of values by the most optimal function
- extrapolating values for current or future time

Templates are provided for recommended software and authors are advised to use them. Please consult the individual conference help pages if questions arise.

WEB GUI

For the convenience of displaying information about the state of the complex, a graphical web interface (Fig. 4) was developed based on the Angular 5 framework [4]. The first consists of colored indicators of each group. Here, red means that an error has occurred on this subsystem, yellow indicates a situation close to critical, green indicates a correct operation. Also, if the data source for any of the subsystems is unavailable, then it is displayed in gray. In addition to the color indication, there is a module in that an informational error message is displayed indicating the time of the error and the subsystem on that it occurred.

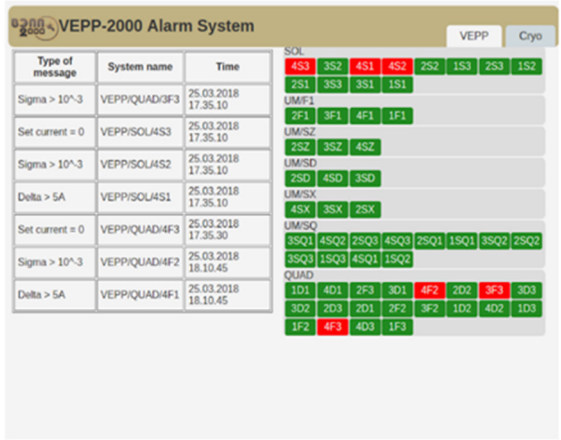


Figure 4: WEB GUI.

CONCLUSION

At present, a prototype of a troubleshooting system at the pulping VEPP-2000 has been implemented and launched in test mode. As part of this work, several algorithms for checking the state of the complex were implemented and connected to an automatic check. In the future we plan to expand the set of rules, as well as review the way they are stored. In addition, the task is to improve the performance of this system, and solve a number of problems associated with distributed memory and timely notification of the state of the complex. It is also planned to expand the scope of their research not only as applied to troubleshooting, but also to their automatic prevention.

REFERENCES

[1] Yu. Shatunov *et al.*, "Project of a New ElectronPositron Collider VEPP-2000", in *Proc. EPAC'00*, Vienna, Austria, p.439.

[2] A. Senchenko *et al.*, "VEPP2000 Collider Control System", in *Proc. PCaPAC'12*, Kolkata, India, Dec. 2012, paper FRCB04.

[3] RabbitMQ official site, `https://www.rabbitmq.com`

[4] Angular5 official site, `https://angular.io`

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP28

DEVELOPMENT OF SOFTWARE FOR ACCESSING THE VEPP-2000 COLLIDER FACILITY ARCHIVING SYSTEM

O. S. Shubina†, A.I.Senchenko, P.Yu. Shatunov
BINP SB RAS and Novosibirsk State University, Novosibirsk, Russia

Abstract

The VEPP-2000 is an electron-positron collider, that was commissioned at Budker Institute of Nuclear Physics. The VEPP-2000 acceleration complex consists of a few main subsystems: BEP booster ring and VEPP-2000 collider ring. Data from accelerator complex are recorded regularly with a frequency at 1 Hz. There is often a need to obtain already stored data for analysis or modeling. In addition, you must provide remote data access to optimize the workflow. The solution of this problem must be universal, and it must be easily adapted to various databases and installation modes. To solve the task, the software was developed based on the client-server architecture. The server part is responsible for processing data in automatic mode according to the developed algorithm. The client part allows to view the data in a user-friendly form. This article talks about the development of software, simplifying access to the VEPP-2000 archiving system, that is launched on the VEPP-2000 control panel.

INTRODUCTION

The VEPP 2000 is an electron-positron collider located at the Budker Institute of Nuclear Physics. The VEPP-2000 acceleration complex (Fig. 1) consists of two main subsystems: the BEP booster ring and the VEPP-2000 collider ring. Beam diagnostics system is based on 16 optical CCD cameras that register the synchrotron light from both ends of the bending magnets and provide full information about beam positions, intensities and profiles. In addition to optical BPMs [1], there are also 4 pick-up stations in the technical straight sections and the one current transformer working as an absolute current meter.

Over than 1200 control channels and 2400 monitoring channels and their joint usage impose rigid restriction on the control system [2]. The architecture of VEPP-2000 software is based on traditional three-layer structure. The important application in the middleware layer is an elaborately designed Log Server [3]. Its purpose is to store all necessary automation system data to the storage.

Every day various accelerator-based experiments are carried out. Sometimes researchers need to obtain the past data for the analysis and calculations. Moreover, it will be useful to have remote access to this data.

Thus, there is the problem of access to the data stored on the hard disk. Also, the solution of this problem have to be universal, and easily adapted to various databases and installation modes.

THE GENERAL SCHEME OF THE APPLICATION

To solve the task, the software has been developed based on the client-server architecture (Fig. 2). HTTP has been chosen as a transport layer protocol, since it is the most common method of data transmission. And also, the REST approach has formed the basis of this application.

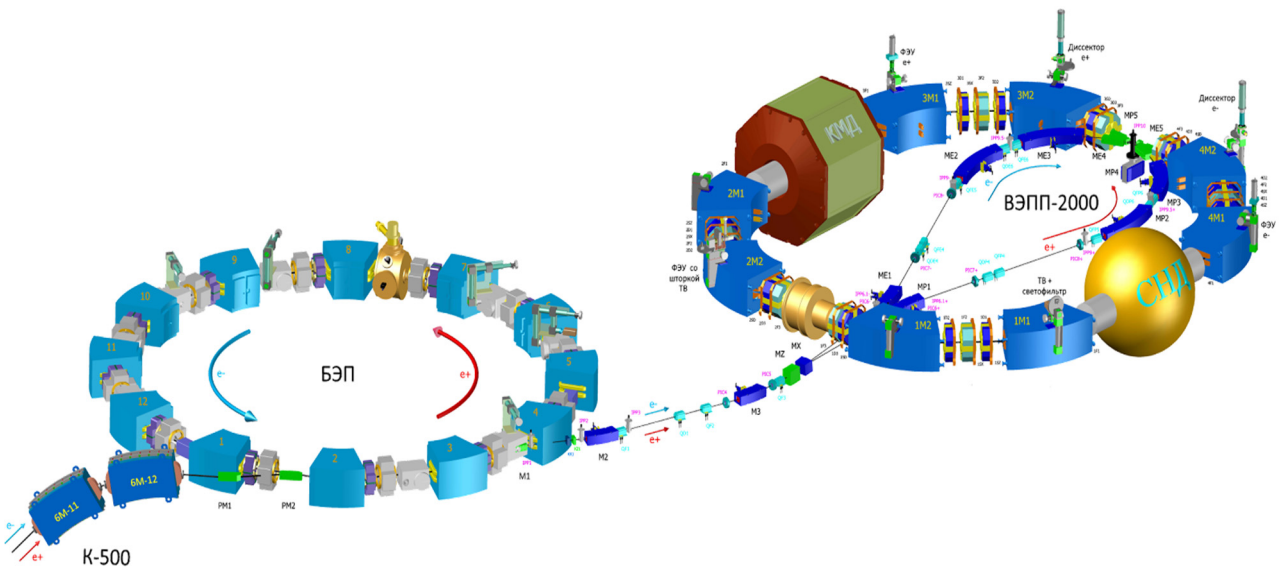


Figure 1: VEPP-2000.

† olgashubina2011@gmail.com.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP28

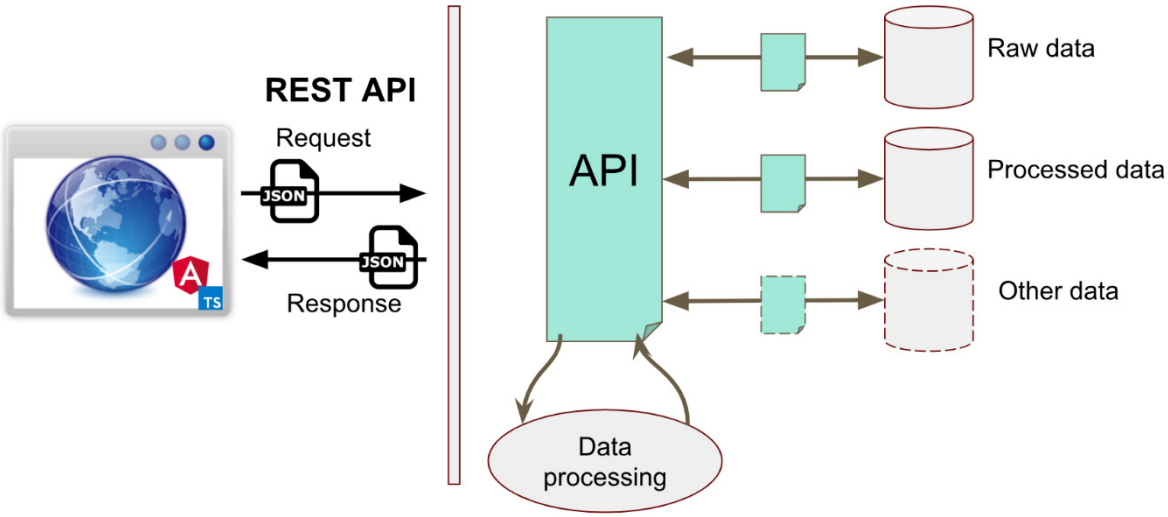


Figure 2: General scheme.

The interaction between the client and the server is carried out using standard HTTP methods. When the user enters all necessary data, the client part of the application generates a special request and passes it as a JSON format message, which has the following structure:

- t1 - the beginning of the requested time interval;
- t2 - the end of the requested time interval;
- channels - the list of channels;
- fields - the method of data processing;
- level - the level of compression.

Next, the server processes the request and sends the response also in JSON format. Then the client works on the received data and provides it to the user. Below we will consider in more detail how the client and server parts of the software are arranged.

THE SERVER PART OF THE APPLICATION

To create the server part, the Flask web framework was used in the Python programming language using the Jinja2 template engine. This framework was chosen because it is designed to create small web applications that have basic capabilities, which are sufficient for our system.

The server part (Fig. 3) provides a single-entry point for obtaining the necessary data via the URL identifier. Its main task is to handle requests coming from the client and to issue data that satisfy the query parameters. Rest approach allows to hide from the client the implementation of specific procedures describing the structure of data storage, as well as procedures for interacting with it.

The server has several main components. The first component is responsible for provision of data from the database. Since it was necessary to create the most universal tool for interacting with data, it was decided to implement a unified software interface for accessing data from various sources. It allows to operate data through small interfaces that are implemented for a specific storage source.

The VEPP-2000 archiving system stores all the necessary data in the object-relational database of PostgreSQL. For example, for a month, the amount of data stored is about 900 million records, which is equivalent to 62 GB of disk usage. Thus, there is the problem of convenient representation, a large amount of data for quick viewing. Therefore, the component responsible for processing the data according to the developed algorithms was created. Since, amount of raw data is growing, there is a need to average new data regularly. To solve this problem, the Celery [4] task manager was used.

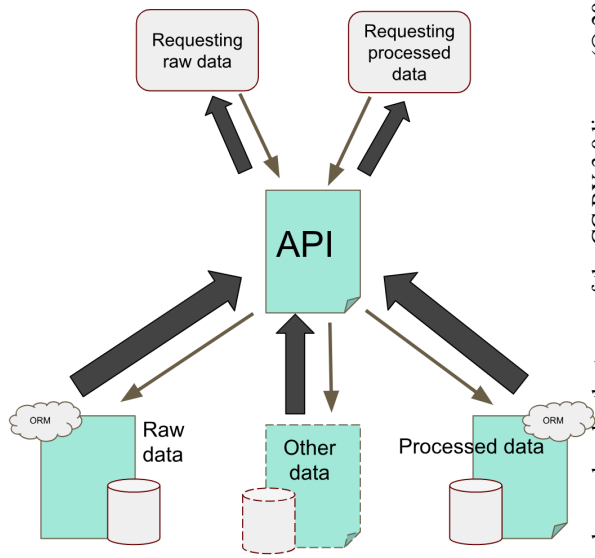


Figure 3: Server part scheme.

In the VEPP-2000 control system, currently, there are two types of channels: the first accepts text values, the second accepts a floating-point number. Depending on the type of data received, two data processing scenarios were developed. For numerical values, this is a finding for a certain period of time:

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP28

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

- arithmetic mean;
- the minimum value;
- the maximum value;
- the median value;
- root mean square deviation.

And for text:

- the left boundary value;
- right boundary value;
- the most significant value;
- central importance.

The last method processes data with both textual and numerical values. Data is processed by the methods described above at three different intervals. Thus, we obtain a group of tables consisting of two processing methods for three levels of compression.

THE CLIENT PART OF THE APPLICATION

The next stage of our work was the creation of the client part. The graphical interface (Fig. 4) was developed on the basis of the Angular5 [5] framework using the TypeScript language.

The main tasks of the GUI are the providing a convenient viewing of channels with various details, and the providing an interface to access data. For the convenience, the channel list was organized as a drop-down tree, similar to a hierarchical file system. For example, the path to the VEPP / Currents / FZ channel will look like this:

- VEPP
  - Currents
    - VEPP / Currents / FZ

Since only one selected value is drawn when displaying the processed data graph, for viewing convenience, when hovering to a specific point, the system displays the remaining values calculated for the given time.

Also, for a more comfortable viewing of the resulting graphs it is possible to scale the graphics and switch between different levels of compression. And also, it provides a convenient navigation on the graphic of mouse movement.

Since data from channels that have textual values cannot be represented in a graph similar to a graph with numerical values, it was customary to visualize information with text values using a table. This method is a kind of graphical interpretation of the tables stored in the database. Also, for more convenience, in the form of a table, it's possible to display data with numerical values.

Despite the fact that the graphical interface was faced with the task of providing fast and comfortable viewing of data, sometimes there might be a need to get raw data for a long period. In this case, the application provides a system for generating a script in Python. It has the same interface as for receiving data, only thus, the user is given a script displayed, which, also, is available for download. This script, when it is launched, connects to the server and uploads the data to a file located on the computer from which it is running.

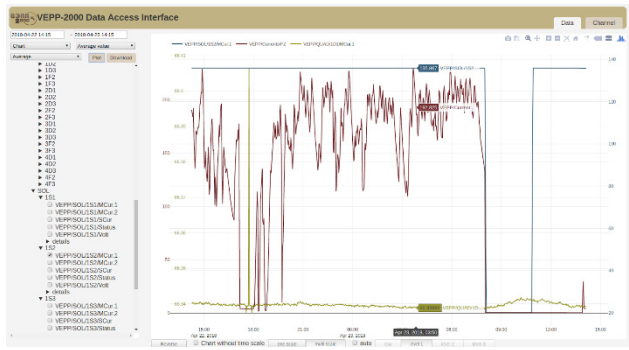


Figure 4: Web GUI.

CONCLUSION

As a result of this work, the software, that provides remote access to the data of the archiving system of the VEPP-2000 accelerator complex, has been developed. Using the REST approach provided a single point of access to data from various sources. As a result, a sufficiently flexible system was obtained in which the client's implementation does not depend on the implementation of the server, which allows, if it is necessary, to make minor or even significant changes to one of the parts without affecting the other at all.

For the convenience of viewing large amounts of data, the algorithms for processing them were implemented. According to the conducted tests, these algorithms allowed to increase the speed of data acquisition on about 90 times. Since the raw data will be regularly replenished, a task manager has been created that performs automatic data processing, according to the developed algorithms. Also, the approach, that allows to avoid the problems of reconnection, has been implemented.

REFERENCES

[1] Yu. Shatunov *et al.*, "Project of a New ElectronPositron Collider VEPP-2000", in *Proc. EPAC'00*, Vienna, Austria, p. 439.

[2] A. Senchenko *et al.*, "VEPP2000 Collider Control System", in *Proc. PCaPAC'12*, Kolkata, India, Dec. 2012, paper FRCB04.

[3] A. Senchenko, D. Berkaev, "VEPP2000 Logging System", in *Proc. PCaPAC'12*, Kolkata, India, Dec. 2012, paper WEPD14.

[4] Celery official site, http://www.celeryproject.org

[5] Angular5 official site, https://angular.io

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP30

DESIGN OF RELIABLE CONTROL WITH STAR-TOPOLOGY FIELDBUS COMMUNICATION FOR AN ELECTRON CYCLOTRON RESONANCE ION SOURCE AT RIBF

A. Uchiyama†, T. Nagatomo, Y. Higurashi, J. Ohnishi, T. Nakagawa, M. Komiyama, N. Fukunishi
RIKEN Nishina Center, Wako 351-0198, Japan
H. Yamauchi, M. Tamura, K. Kaneko, SHI Accelerator Service, Ltd., Tokyo 141-0032, Japan

Abstract

In the RIKEN Radioactive Isotope Beam Factory (RIBF) project, a superconducting linear accelerator has been implemented to enhance the beam energy necessary for promoting super-heavy element search experiments. A new 28-GHz electron cyclotron resonance ion source (ECRIS) has been installed upstream of it. Its control system has been planned to comprise the Yokogawa FA-M3V series, which is a programmable logic controller (PLC) with Experimental Physics and Industrial Control System (EPICS) because basically the same control system has been successfully operated for our existing ECRIS control system. However, the existing ECRIS control system with PLCs has a disadvantage of low reliability for communications between PLC stations. In addition, higher expandability is required because some devices, such as a power supply for an oven, will be changed depending on ion species produced at the ion source. In the new system, we have designed the control system by utilizing a star-topology fieldbus for communications between the PLC stations to establish safety and expandability.

INTRODUCTION

The RIKEN Radioactive Isotope Beam Factory (RIBF) accelerator facility consists of five cyclotrons, including a superconducting ring cyclotron, and two linear accelerators [1]. In the RIBF, we constructed a distributed control system based on the Experimental Physics and Industrial Control System (EPICS) for electromagnet power supplies, beam diagnostic instruments, vacuum control systems, etc. [2]. In FY2016, we started a new project at RIKEN RIBF to further advance synthesis of super-heavy elements with atomic numbers greater than 119, and the main points are as follows [3]:

A superconducting linear accelerator (SRILAC) is newly installed in the downstream part of the RIKEN linear accelerator (RILAC) to enhance the beam energy [4]. To increase the beam intensity for RILAC, the existing 18-GHz electron cyclotron resonance ion source (ECRIS) [5] is also upgraded to a new superconducting ECRIS (SC-ECRIS) [6].

Of these two, the new SC-ECRIS has the same structure as the RIKEN 28-GHz SC-ECRIS installed in the upstream of RILAC2 [7], which is one of the other injectors of the RIBF currently being operated. Therefore, as the devices to be controlled are almost the same as the RIKEN 28-GHz SC-ECRIS, the new SC-ECRIS control system should be constructed based on the current RIKEN 28-GHz SC-ECRIS control system with several improvements to overcome the limitations of the present system. In this proceeding, we discuss disadvantages of the current RIKEN 28-GHz ECRIS control system and report the design of the newly constructed control system in detail.

† a-uchi@riken.jp

RIKEN 28-GHZ SC-ECRIS CONTROL SYSTEM

System Concept

As the main feature of the RIKEN 28-GHz SC-ECRIS control system, Programmable Logic Controllers (PLCs) of the Yokogawa FA-M3 series and EPICS are utilized. The detailed system chart is shown in Fig. 1. The control system is mainly divided into two parts. One is F3RP61-2L, which is a CPU running Linux, to provide the operation services such as controlling of gas valves and power supplies [8]. In the case of the F3RP61-2L-based PLC station, EPICS is installed on the Linux-running system as the PLC CPU's operating system and the EPICS Input/Output Controller (IOC) is implemented as the middle layer for operation services, the so-called embedded EPICS. The other is the part of the implementation of the interlock function as a safety system [9]. Because the sequence CPU-based PLC station has the real-time feature, it is suitable for constructing the safety system, in which not so fast response, such as less than 1 ms, is required. Therefore, the interlock function is realized by the sequence CPU-based PLC station independently from the Linux-CPU-based PLC station, and the state of interlock is monitored by another external EPICS IOC via the TCP/IP network.

As the interlock, the system realizes the function of safely turning off the high-voltage power supply at the time of door opening, turning off the radio frequency (RF) at the time of vacuum abnormality.

Communication between PLC Stations

In the RIKEN 28-GHz SC-ECRIS control system, the main station, which is the installed Linux PLC CPU, manages four substations connected by fieldbus communication electrically isolated by optical fibers. In the case of the Yokogawa FA-M3 series, the fieldbus is called the FA bus. They communicate through FA-bus modules. Because heavy ions generated by an ion source are extracted to the low-energy beam transport by high voltage,

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP30

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

substations also need to be implemented at the high-voltage stage in some cases. As a typical example, the power supplies for the high-temperature oven method [10] and BIAS disk method [11] need to install the PLC substation on the high-voltage stage. Therefore, the optical fiber, being an insulator, is adopted for fieldbus communication between the main station and the substations. Consequently, the signal exchange between the sequence PLC CPU-based station for interlock and the Linux CPU-based PLC station needs to use EPICS Channel Access (CA) via the TCP/IP network or electrical signal.

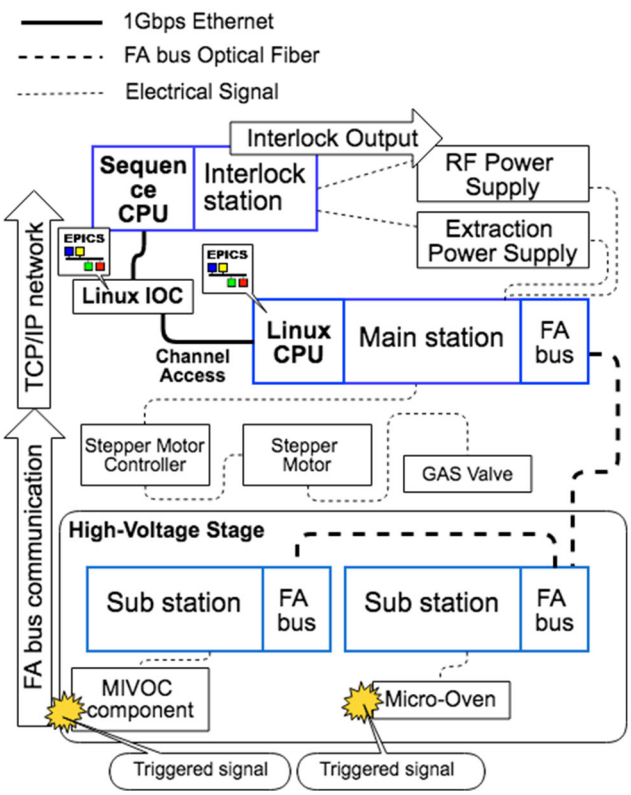


Figure 1: System chart of RIKEN 28-GHz SC-ECRIS control system. CA protocol is utilized for operation services and triggered signal.

Reliability of TCP/IP-based Interlock

There are two methods for sending the interlock signal from the Linux CPU PLC station to the sequence CPU PLC station, viz., sending an electrical signal to the sequence CPU PLC station via the Linux PLC's output module and sending the interlock signal by EPICS CA via TCP/IP. When using state information of the substation implemented in the high-voltage stage as an interlock signal, this system has the means to exchange signals only by the EPICS CA via the TCP/IP network. In general, the reliability of the CA-based interlock system is not high because of the failure of the network switch in the network route, the problem of slow signal transmission speed compared with the bus access, and the problem of reliability of the EPICS IOC. Thereby, the reliability of the signal through TCP/IP is lower than that of the electric signal, and thus the reliability of the interlock is also not relatively high.

NEW SC-ECRIS CONTROL SYSTEM

System Design

In the new project, a new SC-ECRIS is installed upstream of RILAC to increase the beam intensity for RILAC. The photograph of the new SC-ECRIS for SRILAC is shown in Fig. 2. Considering the new SC-ECRIS for SRILAC, the control system should follow the current RIKEN 28-GHz SC-ECRIS control system upstream of RILAC2, because the RIKEN 28-GHz SC-ECRIS control system has achieved success in the RIBF project. Thus, in the case of the new SC-ECRIS control system, we have adopted the Yokogawa FA-M3V series, which is the upgraded FA-M3, for system construction. On the other hand, we should also solve the disadvantage of low reliability for interlock features in the RIKEN 28-GHz SC-ECRIS control system when constructing the new SC-ECRIS control system. Accordingly, to solve the disadvantage, a new SC-ECRIS control system has been designed by implementing two different types of CPUs in the main PLC station. Essentially, the sequence PLC CPU in the first slot and the Linux PLC CPU in the second slot have been implemented in the same PLC base module. In the sequence PLC CPU, the ladder program runs for the interlock system, and the Linux CPU runs the EPICS IOC and provides operation services to users via the EPICS CA protocol. Currently, the new SC-ECRIS control system consists of a main PLC station and five PLC sub stations with star-topology fieldbus communication using optical FA bus modules. The detailed system diagram is shown in Fig. 3.

At present, this control system does not include the control of superconducting electromagnet power supplies. Control of superconducting electromagnet power supplies is implemented in a system consisting of another controller and client without EPICS IOC. The system implementation test with EPICS is in progress now.

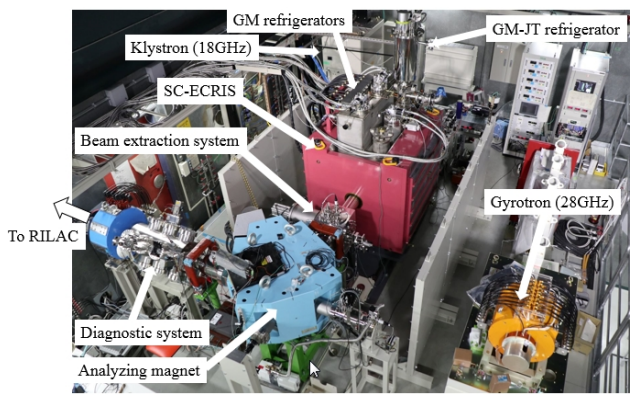


Figure 2: Newly installed SC-ECRIS for SRILAC. Currently, an 18-GHz RF source (Klystron) is only used for the beam commissioning [5].

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP30

Interlock

Generally, X-rays are generated during the operation of the ion source and the high voltage is utilized for extraction of heavy ions generated. Therefore, this interlock has to prevent workers from exposure to X-ray and/or accidental electrification.

Interlocks are roughly divided into two types. One is a human protection system, which has a mechanism to turn off the RF power and power supplies for high-voltage beam extraction when a person enters the ion source room. This interlock system utilizes the opening of doors and entrance information as the triggers. The other interlock is a machine protection system. The system monitors the cooling water, vacuum condition, etc., and automatically stops the ion source and the devices constituting it, safely in case of abnormality.

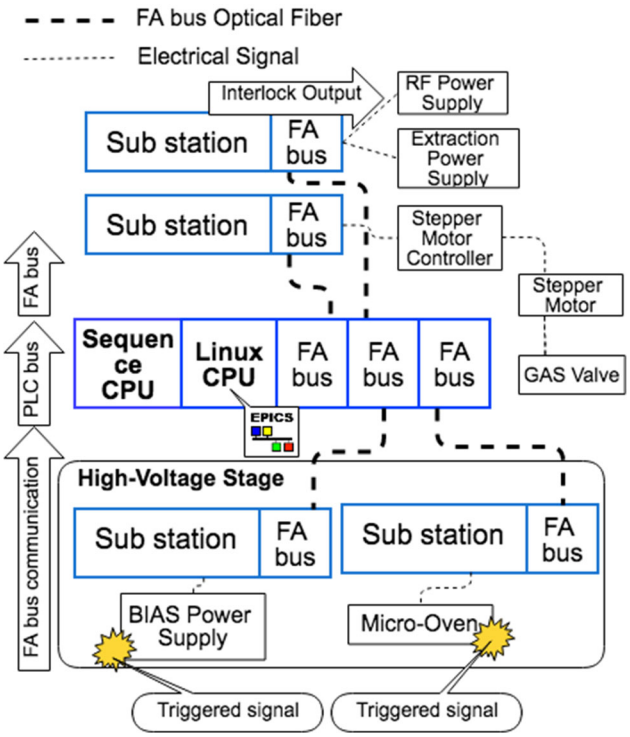


Figure 3: System chart of new SC-ECRIS control system for SRILAC. Interlock signal is delivered by bus communications.

I/O Sharing Method

The feature of the Yokogawa FA-M3V series is that, as a behavior when there are two CPUs, reading can be done from both CPUs unconditionally, but direct instructions to the output module adopt exclusive control. Therefore, the basic write (output control) is performed only from the sequence PLC CPU, when performing output control from the EPICS IOC on the Linux PLC CPU; it will be realized via internal registers of the sequence PLC CPU, for example, I00001.

I/O Refresh Time for FA Bus using 0ptical Fiber

According to Reference [12], the I/O refresh time for each access type (read/write) via the optical FA bus module is estimated to be the product of the access time and "Number of modules converted to a 16-point basis". The access time is the sum of "Time dependent on access type" and "Time dependent on transmission distance". Based on this estimation, the theoretical transfer I/O refresh time is 16.8 μs in this system. This value is sufficiently higher than the TCP/IP-based network speed.

System Availability

Based on the status information of the doors and entrance, we are able to provide a human safety mechanism as an interlock system of RF source and extraction voltage of SC-ECRIS. Accordingly, it is possible to realize the interlock signal for the machine protection system via FA bus communication without going through the TCP/IP network, even though the control of power supply for the oven system is mounted on the high-voltage stage. As the system behavior, the oven power supply is turned off by triggered by the degree of vacuum and cooling water temperature.

CONCLUSION

We solved the disadvantage of the insufficient reliability of the interlocks in the RIKEN 28-GHz SC-ECRIS control system and constructed a control system of the new SC-ECRIS for SRILAC. This new SC-ECRIS control system was successfully used in the test operation of the new SC-ECRIS performed in August 2018 without any serious problem. Because of the mounting of two CPUs in one base unit, it is possible to exchange the trigger signal for interlocking with the sequence PLC CPU from the Linux PLC CPU via the FA bus on the PLC base module. Therefore, it is possible to share the interlock signal of the high-voltage stage with the Linux PLC CPU and the sequence PLC CPU without going through the TCP/IP network, and it improves the system reliability of the interlock feature successfully without lowering the conventional system usability.

REFERENCES

[1] O. Kamigaito *et al.,* "Present Status and Future Plan of RIKEN RI Beam Factory", in *Proc. IPAC'16*, Busan, Korea, May 2016, pp. 1281-1283.

[2] M. Komiyama *et al.,* "Status of the RIKEN RI Beam Factory Control System", in *Proc. ICALEPCS'13,* San Francisco, CA, USA, Oct. 2013, pp. 348-351.

[3] H. Okuno *et al.,* "Operational Experiment and Upgrade Plans of the RIBF Accelerator Complex", in *Proc. Cyclotrons'16,* Zurich, Switzerland, Sep. 2016, pp. 1-6.

[4] N. Sakamoto *et al.*, "Construction Status of the Superconducting Linac at RIKEN RIBF", in *Proc. LINAC'18*, Beijing, China, Sep. 2018, paper WE2A03.

[5] K. Ozeki *et al.*, "SUPPLY OF METALLIC BEAMS FROM RIKEN 18-GHz ECRIS USING LOW-TEMPERATURE OVEN", in *Proc. HIAT2015,* Yokohama, Japan, Sep. 2015, pp. 244-246.

[6] T. Nagatomo *et al.*, "New 28-GHz Superconducting ECR Ion Source for Synthesizing New Super Heavy Elements of

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP30

Z > 118", presented at ECRIS'18, Catania, Italy, Sep. 2018, paper TUA3, unpublished.

[7] Y. Higurashi *et al.*, "Recent Development of RIKEN 28 GHz SC-ECRIS", in *Proc. ECRIS'16*, Busan, Korea, Aug.-Sep. 2016. pp. 10-13.

[8] M. Komiyama *et al.*, "Upgrading the Control System of RIKEN RI Beam Factory for New Injector", in *Proc. ICALEPCS'09*, Kobe, Japan, Oct. 2009, pp. 275-277.

[9] A. Uchiyama *et al.*, "Construction of Client System for 28GHz SC-ECRIS", RIKEN Accelerator Progress Report vol.43 (2010), p. 133-134.

[10] J. Ohnishi *et al.*, "Development of a high-temperature oven for the 28 GHz electron cyclotron resonance ion source", *Review of Scientific Instruments*, 85, 02A941 (2014).

[11] Y. Higurashi *et al.*, "Production of a highly charged uranium ion beam with RIKEN superconducting electron cyclotron resonance ion source", *Review of Scientific Instrument*s, 83, 02A333 (2012).

[12] Fiber-optic FA-bus Module User's Manual;
https://web-material3.yokogawa.com/IM34M06H45-01E.us.pdf

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP31

DESIGN OF PLC TEMPERATURE FLOW ACQUISITION SYSTEM BASED ON EPICS

H. Zheng, Y. X. Chen, Institute of Modern Physics, Chinese Academy of Sciences, 730000 Lanzhou, China
H. Cao, Contributor, Institute of Modern Physics, Chinese Academy of Sciences, 730000 Lanzhou, China

Abstract

In the design of the ADS injector II, the RFQ cavity holds a supreme status, and the RFQ temperature and flow information are the key parameters for the cavity frequency tuning. To ensure the long-term, stable and accurate acquisition of temperature flow data is the essential task of control. In this paper, the PLC temperature flow acquisition system which is based on EPICS design was described, and the EPICS driver of this PLC was developed independently. The driver uses TCP/IP connection to EPICS IOC, and the communication protocol uses the "data block overall transmission protocol", to ensure the stability of the device's data communications. After 3 months of long-term operation inspection, this acquisition system can ensure long-term and stable acquisition of real-time temperature and flow data of the equipment, and be able to send control information to related controlled equipment. In addition, redundant PLCs and redundant IOCs were adopted in this acquisition system to make the switch to alternate channels within milliseconds once a channel fails.

TECHNICAL SOLUTIONS AND IMPLEMENTATION

Hardware Components

The water temperature flow monitoring of the RFQ is implemented by two FC 460R controllers which can be backed up each other. The PROFINET protocol is used to connect four sets of PROFINET slaves. The slaves use analog modules to collect temperature and flow signals. The temperature signals are collected by a resistive temperature sensor. The acquisition of flow signal is collected by the rotameter and converted into a current value. The operation schematic diagram of RFQ the temperature flow acquisition system is shown in Fig. 1. The collected signal is transmitted by the analog module from the PROFINET bus to the FC 460R controller. The gathered data is transmitted periodically by the controller to the EPICS IOC, published and archived. If the temperature and flow values exceed the threshold, an alarm message will be transmitted by the controller, and the corresponding action will be executed.

In order to overcome the on-site interference signal, the isolation modules are adopted by the temperature acquisition. In addition, the temperature and flow data are processed by average and anti-shake modes to prevent large fluctuations in data.

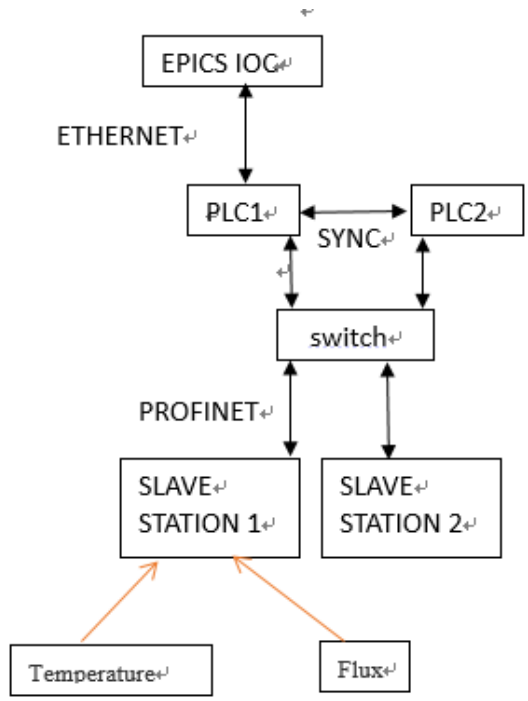


Figure 1: The operation schematic diagram of RFQ the temperature flow acquisition system.

All temperature and flow data are set with threshold comparison. It can be clearly seen through the OPI interface whether the threshold is exceeded. Moreover, the damaged signal of the sensor can be shielded quickly by the 'bypass' on the interface without affecting the beam debugging.

IOC Implementation

This driver is used to connect a Phoenix redundant PLC to the EPICS IOC via TCP/IP. A so-called "whole data-block transmission protocol" is used by the communication protocol. This driver is written by the Swiss PSI laboratory for the Siemens S7 PLC based on the basis of the EPICS driver to support redundant PLC pairs.

The data is periodically exchanged between the driver and the PLC through the network, and the size of the data block is fixed. All process variables are arranged in a data block in a certain way, with an offset in the data block as the flag. The process variables could be of all types supported by EPICS. The programmers of IOC and PLC should reach an agreement on the size of the data block and

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP31

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

the arrangement of the process variables in advance. Because after the operation, the size of the data block and the arrangement of the process variables cannot be dynamically adjusted.

On the other hand, the output records connected to the PLC are checked periodically by the driver whether they have been processed since the last cycle. If there is one and only one output record has been processed, the driver will send the data block containing all output variables to the PLC. When the IOC is started, the output data block is sent only after the "PINI" is run.

OPERATION INTERFACE

The interface of RFQ is mainly composed of two parts: RFQ cavity temperature and RFQ cavity flow, as shown in Fig. 2. The monitored temperature and flow of the RFQ cavity and water are clearly indicated in the interface. The interface state (interface configuration) is refreshed every second, and there is an indicator light indicating its fault status, the green light is normal while the red light is fault. There is a button on the right side of each indicator to shield the fault temperature or flow signal. The green light is on to enable the bypass function, otherwise is to disable the bypass function. The green small light on the right side of the temperature box will turn red once the temperature exceeds the threshold. If the time exceeds the threshold value for more than 30ms, the temperature display box will turn red, indicating that the temperature of the road actually exceeds the threshold. Then the RFQ power source will be cut off. The anti-shake processing of temperature monitoring is realized by the delay of 30ms.

TECHNICAL TEST, CONCLUSION

The redundant controllers and redundant IOCs are used to develop the RFQ water temperature flow monitoring system and therefore the high system reliability can be achieved. The temperature and flow signals are quickly transmitted by the high-speed PROFINET bus, thus ensuring the signal stability from hardware and software. In addition, the fault can be detected rapidly and the power source is cut off promptly to protect the RFQ cavity.

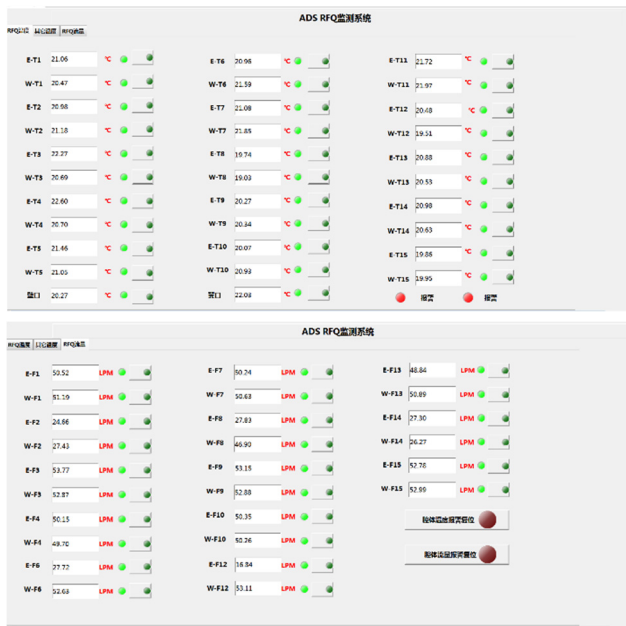


Figure 2: The interface of RFQ temperature and flow.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP32

THE DESIGN AND DEVELOPMENT OF AN AUTO-CONDITIONING SRF CAVITIES SOFTWARE

Hu Cao, Y. Chen,
Institute of Modern Physics, Chinese Academy of Sciences, 730000 Lanzhou, China

Abstract

As one of the major components of ADS Injector II, SRF (Superconducting Radio Frequency) cavities are used to transmit the intense-beam proton reliably, stably and efficiently. Before starting the process of transmitting particle beams, SRF cavities are routinely conditioned to achieve its optimized status in the deliverable energy. The whole conditioning process is involved in various types of hardware devices and is also a heavy task for engineers to manually operate these equipment. In this paper, the software ANSC is presented in details, which is used to automatically condition SRF cavities. At the present, ANSC is in the stage of testing. During the testing, ANSC indeed can achieve comparative results compared with manually operated conditioning.

INTRODUCTION

As China Accelerator Driven System (C-ADS) Injector II achieved the expected target of 25 MeV[1], how to make an improvement in efficiency is our current focus[2]. In every accelerator physics experiment[3], engineers of SRF are always required to condition SRF cavities by manual operation to ensure that these cavities can reach their optimal deliverable energy. The task is heavy for the engineers and manual operation is also low in efficiency. In order to get rid of the difficult situation, our engineers in the ADS control group are developing an Auto-Conditioning SRF Cavities program called ANSC. Currently, ANSC is in the stage of testing and obtained some positive feedback during the testing process. ANSC is based on the distributed control system EPICS, which has been widely used in accelerator field. Therefore, it is totally compatible with current control system in ADS and is also convenient to update the software in the future. During the development of the program, python and Epics Qt are core components in which Epics Qt is used to customize GUI and python is exploited to carry out the auto-conditioning process.

In the following sections, ANSC will be described in detail.

THE GUI OF ANSC

Because of flexibility and practicality of Qt,it is widely used in software development. At the same time, Epics Qt combines Qt's advantages with EPICS'. As a consequence, it is chosen as GUI design tool of ANSC. In reality, Epics Qt demonstrates its high performance in displaying PVs by Channel Access.

In ANSC, there are two main user interfaces: setting GUI and operation GUI. Setting GUI in Fig. 1 is responsible for setting some basic parameters; Operation GUI in Fig. 2 is used as setting auto-conditioning related parameters and an entry for executing auto-conditioning process.

In the Setting GUI, the key status information is set such as vacuum fluctuation, vacuum protection status and MPS (Machine Protection System), which is key for engineers to execute the auto-conditioning process. Operation GUI is used before the beginning of the process. Its main function is to set important parameters and thresholds such as vacuum thresholds, auto-conditioning up steps and down steps, then start the auto-conditioning process. In addition, scanning frequency and phase can be executed.

Figure 1: Setting GUI of ANSC.

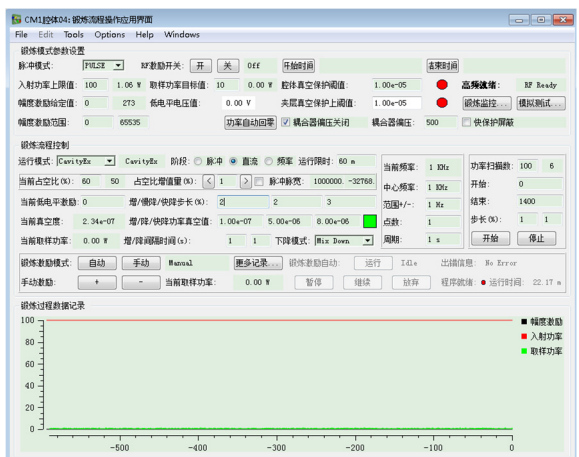


Figure 2: Operation GUI of ANSC.

REALIZATION OF AUTO-CONDITIONING PROCESS

The function of the GUIs is totally carried out by Python. Taking the complexity of auto-conditioning SRF cavities and restriction requirement in time into account, the program consists of four threads, including auto-loading thread, protection thread, exception dealing thread and main thread. The auto-loading thread takes charge of

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP32

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

increasing power; the protection thread is used to monitor status of devices and takes proper protection actions, and the exception dealing thread deals with other accidents such as disconnection of PVs.

Figure 3: The display of the result of ANSC.

In Fig. 3, the phenomenon of carrying out the auto-conditioning process is displayed. Figure 4 is the user interface to display historical information in execution of ANSC.

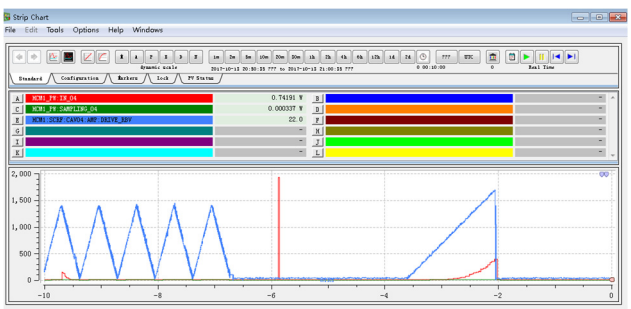


Figure 4: The display of historical records.

ANALYSIS OF PERFORMANCE

After the comparison of new user interface and old user interface in Fig. 5., it is definitely assured that the user interface of ANSC is more easy for users to use and is more beautiful than the old user interface.

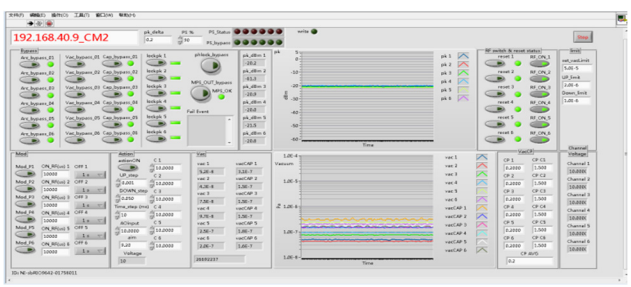


Figure 5: The user interface of old conditioning program.

During the testing, ANSC can meet basic requirement of auto-conditioning SRF cavities. It is used to increase power by automatically operating PVs and also monitors all kinds of alarms in the proceeding. To a great extent, ANSC can reduce the burden of engineers taking charge of SRF and also improve the efficiency in conditioning SRF cavities. Therefore, it demonstrates that auto-condition program is certainly alleviate the burden of engineers.

On the other hand, conditioning SRF cavities is complex and is extremely restricted in speed in some aspect. For example, when ARCA happens in the process, it is dealt instantly within a few μs, which is beyond the capability of software. Thus, it is necessary to come up with a new solution to complement EPICS's shortage in microsecond time. In addition, how to ensure multi threads harmoniously is another practical problem.

CONCLUSION

Until now, the testing task of ANSC is mostly completed. there are some serious prolems needed to be tackled; otherwise, the reliability of ANSC would be threatened greatly. For example, when ARCA is detected, ANSC is asked to cut down power supply within a few us which is beyond python because python is incapable of dealing with the special situation in such short time. As for the emergencies described above, some methods are being tried.

How to build a well model in theory is our final goal so that ANSC can totally replace humans and set them free from boring and repeated task. Meanwhile, increasing the robustness and reliability of ANSC is also critical for our future work. According to these difficulties, full automatic process is a long way.

ACKNOWLEGEMENT

During carrying out the work referred in the paper, we get many help form co-workers in ADS Linac control group. The authors wish to thank the numerous persons who participated in activities on ADS control group.

REFERENCES

[1] M. Stirbe *et al.*, “RF Conditioning and Testing of Fundamental Power Couplers for SNS Superconducting Cavity Production”, in *Proc. PAC’05*, Knoxville, Tennessee, USA, May 2005, paper TPPT083, p. 4132, doi: 10.1109/PAC.2005.1591741

[2] S. Liu *et al.*, “OPTIMIZATION OF THE SUPERCONDUCTION SECTION OF INJECTOR II FOR C-ADS”, in *Proc. HB’12,* Beijing, China, Sep. 2012, paper MOP232, p. 122.

[3] Y. Guo *et al.*, “Design of Software Platform for Injector II Control System in ADS”, *Atomic Energy Science and Technology,* vol. 48, pp. 704-707, Oct. 2014, doi: 10.7538/yzk.2014.48.S0.0704

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP33

INTRODUCTION OF CiADS CONTROL SYSTEM

Youxin Chen, Hu Cao, Zheng Hai, Haitao Liu, Jing Wang, Qiangjun Wu
Institute of Modern Physics, Chinese Academy of Sciences (IMP/CAS), Lanzhou, P. R. China

Abstract

CiADS is a science researching facility, which destination is about energy Providence. The control system of CiADS will have more than hundred types of device, and include more than thousand equipment and sensors. Based on the background of researching and energy project, the control system should overcome two challenges. First is that building a open architecture to face the flexibility of changed requirement. the second is that the flexibility should as less as possible influence the checking result of nuclear law and standard by authority. To meet the requirement, the control system will be divided into 3 levels. Level 2 will provide the OPI, data analysis interface and simulation to all users. Level 1 provides implement of control and security logic. Meantime it will provide an engine and interface for collection and package of some reconstructed data. Level 0 will implement the local control and provide all data and information to other levels. The paper mainly introduces the architecture and some works to build the control system to make it to overcome the two challenges.

BACKGROUND

CiADS (China Initiative Accelerator Driven System) is a large scientific project. It consists of three parts: linear accelerator, spallation target and nuclear reactor. The facility is a complex system, that consists tens of thousands equipment. There are millions of status information, thousands of control points and automated or semi-automated control processes in the control system. As planned. The control system was completed and put into operation in 7 years.

THE ANALYSIS OF REQUIREMENT

As a normal control system, the requirement of CiADS can be divided into two levels. The basic requirements: real-time acquisition of all state information of facility and can remote control various devices. On this basis requirement, it is necessary to provide enough safety and reliability. In addition, the facility will work at complex and various operation modes, the control system needs to adapt and support it. Finally, the control system also needs to provide data and data analysis support to help the operators to achieve the research aims. The following is explained separately.

Basic Functional Requirements

In order to support the operation of the device, the control system needs to implement remote control and reading status information of all devices., which is the most basic functional requirement of the control system. In such a complex system, there are hundreds of different types of devices, which have different control parameters and different operational functions. To reduce the difficulty of system integration, the unified hardware and software interface should be adapted.

In addition, the timing system must be provided for synchronized device operations; the MPS system for protecting the accelerator and the security protection system for nuclear safety. These together constitute the most basic service of the control system.

The timing system. 1. Provide the Trigger signal to synchronize the output or action of the related equipment. 2. Provide time service to the entire facility to achieve strict synchronization of system time to the relevant device, so that the data or status collected by the relevant device can be analyzed and compared on the same time axis. The CiADS control system will use the White Rabbit technology to implement the timing system.

The beam of proton accelerator will be reached mA, and the energy is hundreds of MeV. In this case, a small loss of the beam will cause serious damage to the accelerator, so a machine protection is necessary. The Machine Protection System (MPS) provides the safety to accelerator, while not affecting the availability. How to balance the safety and availability is a challenging task.

As a nuclear-related facility, the control system also needs to equip a high-reliable safety protection system. This protection system is designed on the analysis of all possible nuclear safety accidents which mainly based on hardware system implementation is based on the facility [1]. The device also requires approval of the nuclear safety regulatory agency.

Operation Modes

During the installation, commissioning (step-by-step debugging) and operation, the entire device will face different operating modes [2]. These different modes will involve the parameters of the device, the protection threshold, and the configuration of the protection system and the operating authority. In order to meet the modes of the facility, the databases, device drivers, and applications

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP33

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

should be used. And then, the design should abstract the equipment and mode and define a sufficiently complete and flexible interface. So that maybe can make it possible.

Archive and Data Analysis Support

As a research facility, in order to achieve the research goal, the collection and storage of the operating and status information is the basis task. For this reason, the control system needs to provide data storage and retrieval functions for this purpose. There are a variety of real-time requirements for each type, which challenges the design of the data archiving engine. On this basis, it is also necessary to provide the necessary platform and technical support for data analysis.

THE ARCHITECTURE OF THE CONTROL SYSTEM

In order to meet the requirements of control system, we need to do related design with software and hardware system. In the design, the architecture (included function block software interface; hardware interface and hierarchical design) must be carefully considered, so that to reduce the impact of changes in demand, because such researching facility is doomed to have many change.

The Design of the Hardware System

The control system interface to most devices is defined as the network interface and the fiber-optic network will be used for the system-level interconnection. In addition, the WR technical solution will be adopted for the timing system, and the Timing system service will be provided to the device through the embedded interface card. In addition, the control system has a unified hardware platform, which is based on PLC, it is used to get the basic temperature and flow valve; control switch; motor control and other functions. The framework of hardware is shown in Figure 1.

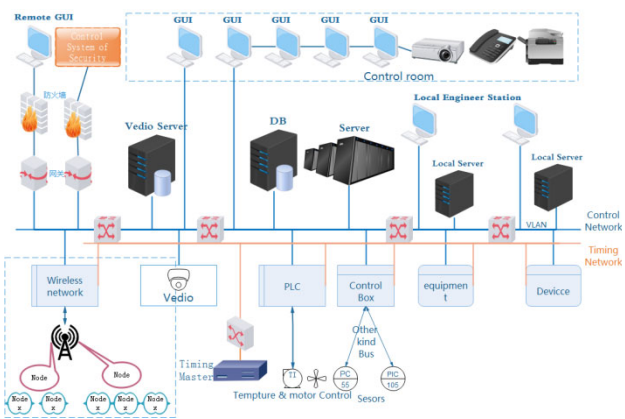


Figure 1: Control system hardware scope of view.

The Framework Design of the Software System

Based on such a set of hardware systems, the software of the control system will use the EPICS system. The control system will be divided into three layers. The view of the architecture is show in Figure 2. The bottom layer is the device driver layer, which is used to operate and read status information of the device, based on EPICS SoftIOC or embedded IOC. In addition, for the convenience of system integration, some software will also support device-level configuration and online software upgrade.

The second layer is the middle layer, which implements all automated or semi-automated control processes; real-time data acquisition and archiving; operation mode change, and modification; security management; alarm systems; physical process simulation and debugging software Support and so on. This layer is the core functional layer of the control system. To this end, it is necessary to standardize the modules and interface as much as possible in order to cope with the changes of subsequent requirements.

The top layer is the interface for all types of operators and management, and in most cases will be implemented by a variety of graphical interfaces. This includes support for data browsing and remote data support for nuclear regulators. This layer is the interface of the control system to the user. For this reason, the control system needs to design a set of GUI writing specifications to control the writing of the interface to realize a unified interface of style and operation mode.

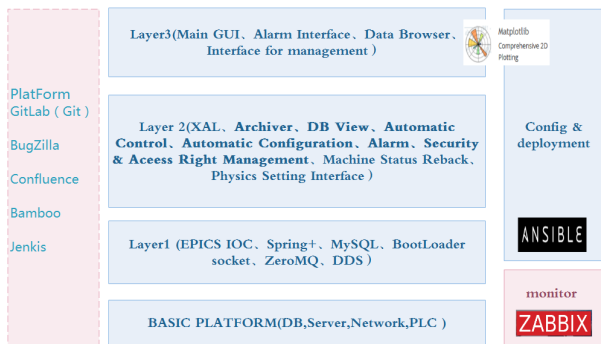


Figure 2: Software architecture of control system.

PROBLEMS AND CHALLENGES

Which Technology is Adopted to Realize the Modification of Equipment Configuration?

In order to help modify the operating mode and facilitate system integration, we need to implement dynamic modification of device parameters and configuration. The current EPICS system can only support a single data type in the V3 version, while the dynamic configuration and

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP33

modification requires the configuration table of the structure type. To do this, the V7 version of the EPICS system or publish-subscribe technology like ZeroMQ may be used to implement such a function. This is a problem that needs to be carefully considered and tested in the next design.

Considerations for Test Environments, Test Packages, and Automated Deployment

As a huge control system, there is a huge number of device interfaces and device access, and a huge amount of code needs to be written or tested. If not fully tested, such a control system must not provide stable and reliable service to the relevant operators of the facility. Therefore, the control system must be tested rigorously and effectively, and the test here will contain the following main contents:

Testing the device interface to check whether the device interface program reads and manipulates the state according to a certain mechanism.

Testing the access of device to check whether every device is to be correctly controlled and read.

Testing of automated or semi-automated operational processes to determine whether the processes themselves operate in accordance with a defined process, and under abnormal circumstances, It will not do any operation that is not allowed.

Performance test to check whether all software systems or hardware systems with performance requirements meet the corresponding performance requirements.

Testing of the GUI, including tests of display, input and output, especially for some important operations and displays, to ensure that the program can run stably, the response of the program must be predictable and deterministic.

Such a series of tests must be done automatically, and each deployed program must be tested, and its own test set must be constantly refined and evolved.

SUMMARY

CiADS control system is a very large and complex system, the current design work is still only in the framework design stage, but has been faced with enormous challenges. Therefore, the design and construction of the whole control system need a lot of serious and meticulous work to achieve a relatively stable, reliable and friendly control system.

REFERENCES

[1] A. Monera Martinez, R. Andersson, *"Overview and Design Status of the Fast Beam Interlock System at ESS", in Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, paper MOPGF138, pp.409-412

[2] H. Danared, "Design of the ESS Accelerator", in *Proc. IPAC'12*, New Orleans, USA, May 2012, paper THPPP071, pp.3904-3906

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP34

THE CONTROL SYSTEM BASED ON EPICS FOR THE EXPERIMENTAL TARGET PROTOTYPE

Q. Zhao†, C. Liu[1], J. R. Sun, C. W. Qiang, L. Li, W. F. Yang, F. Wang, M. B. Lv, L. L. Pang, Y. B. Zhu, M. H. Cui, S. X. Zhao, Y. B. Shen, X. Y. Zhang
[1]Institute of Modern Physics, Chinese Academy of Sciences, 730000 Lanzhou, China
[1]also at Northwest Normal University, 730070 Lanzhou, China

Abstract

Building a high power spallation target is one of the critical issues in Accelerator Driven System (ADS). The control system which was built based on the real-time distributed control software of experimental physics and industrial control system (EPICS) was attempted for the experimental target prototype. There are several sub-systems in the target system, e.g. the elevating system, the vacuum system, the heat-exchanging system. As an IOC, each subsystem was finally integrated into the control system of the target by different drivers and methods because different hardware devices were used for each sub-system. The "SLS s7 driver" which was developed based on the Ethernet Interface was used for the communication between the Siemens PLCs and the Human Machine Interface (HMI). The interfaces between Labview and EPICS were used for the National Instruments (NI) DAQs systems. In addition, the driver developed by ourselves was used for devices with serial ports, e.g. RS-232 or RS-485/422. The control system was finally proved stable and could basically meet the elementary requirements of the spallation target system.

INTRODUCTION

In an Accelerator Driven System (ADS), a heavy metal spallation target locates at the centre of a sub-critical core. A beam of high intensity protons emitted from an accelerator bombard on the target, generating neutrons to drive the sub-critical reactor. Building a high power spallation target is one of the critical issues in an ADS [1]. In China initiative Accelerator Driven System (CiADS), a new concept for a high-power spallation target: the gravity-driven dense granular target (DGT) was proposed.

An experimental target prototype has been constructed and tested for some important issues. The layout of the prototype is shown as Fig.1. The helium loop which is an auxiliary system is separate and not shown in Fig.1. Three loops: Loop 1, Loop2 and Loop 3, as shown in Fig.1, were designed for different functions and experiments. The target design is one of the challenging and key technologies. So, the Loop 1 was designed mainly for testing the performance of the spallation target. At the same time, the feasibility of the mass flow-meters and sieve sorter was also tested. The Loop 2 was designed for testing the feasibility of the heat-exchanger. A proton beam was not considered at the first stage of the target design, so, a middle frequency heating device was used in the Loop 2 to heat the grains before they flow into the heat-exchanger. The Loop 3 acts only as a recovery system. A four-way valve was fixed at the entries of the three loops to control which loop to be used for experiment. Many important issues would be tested and studied as the experimental target prototype running, e.g. flowing character of the grains in the target container, heat conductivity of the grains, the feasibility of the method which is used for measuring the rate of the flow.

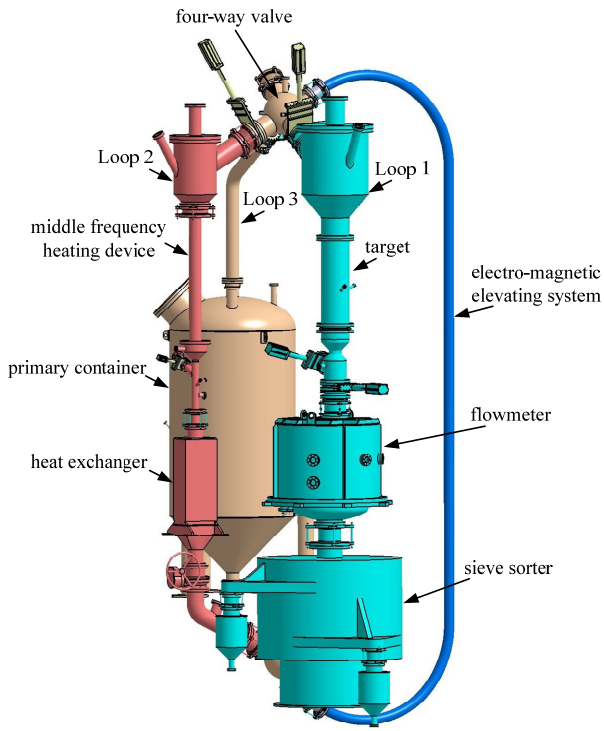


Figure 1: Layout of the prototype.

The architecture of the control system is also one of the important issues to be tested and studied. The control system of the accelerator has been designed and constructed based on EPICS [2, 3]. It will be easy to couple together if the control system of the target system is also constructed based on EPICS. However, considering the software toolkits of EPICS is open-source, the stability of the control system constructed based on EPICS is the most worried and concerned problem due to the extreme high safety requirements of the target system. Constructing the control system based on EPICS for the target system aimed to initially testify the stability of it.

* Work Supported by the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDA03030000) and the National Natural Science Foundation of China (Grant No. 11505258)
† email address: zhaoqiang46@impcas.ac.cn

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP34

THE ARCHITECTURE OF EPICS

EPICS is a set of Open Source software tools, libraries and applications developed collaboratively and used worldwide to create distributed soft real-time control systems for scientific instruments such as a particle accelerators, telescopes and other large scientific experiments [4]. It consists of a set of software components and tools that application developers can use to create control systems. The basic components are operator interface (OPI), input/output controller (IOC) and local area network (LAN) [5]. The architecture of EPICS is shown as Fig. 2.

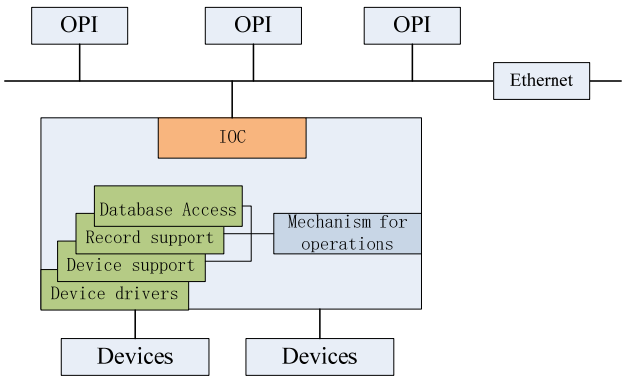


Figure 2: Layout of EPICS.

The OPIs working as clients read (or receive) and write (or send) data with IOCs (working as servers) through the Ethernet. The platforms OPIs running on usually are Windows, Linux and VxWorks. An IOC works as a server which responses the request sent by an OPI. It is the heart of the EPICS and has a complex mechanism to control the data flow.

THE CONTROL SYSTEM FOR THE PROTOTYPE

The Architecture of the Control System for the Prototype

There are various devices used in the control system of the target system for various physical demands. Some physical quantities are slow-response, e.g. temperature. However, some others are fast-response, e.g. neutron flux. Controllers have been very advanced now and usually have at least one Ethernet interface. In consideration of the feasibility and simplicity of integrating the devices with EPICS, the Siemens PLCs were finally used for most of the measuring and controlling of the physical quantities. And the NI products were used for those physical quantities that demand high precision and high sampling rate even though some of them could not be measured because proton beam was not considered during the design of the experimental target prototype.

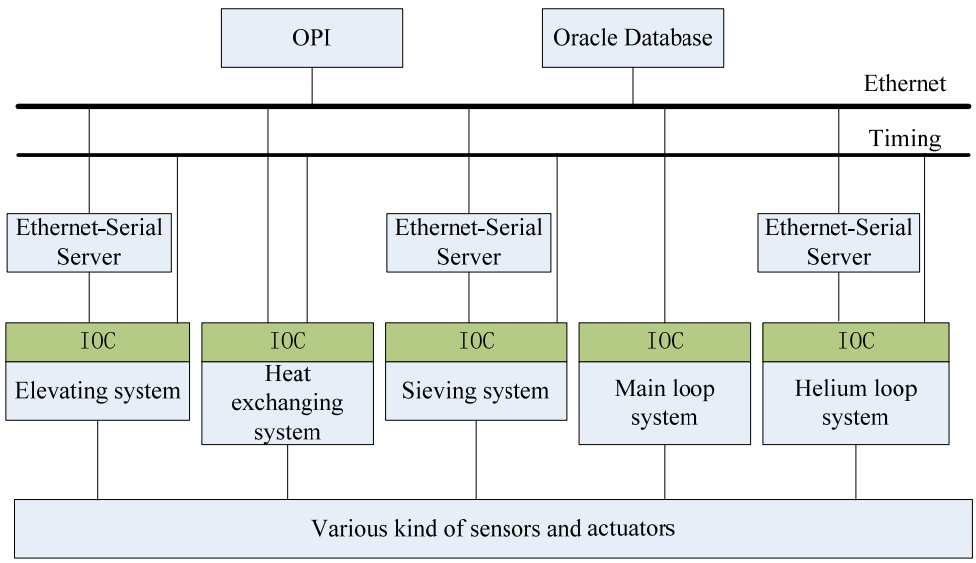


Figure 3: Architecture of the control system.

The architecture of the target control system is shown as Fig.3. As Fig.1 shows, the target system contains several sub-systems and each one has their own control system. So, they all were considered as an IOC when integrated with EPICS. The OPI was developed by Labview rather than CSS because the Labview can meet the requirements of the design now. All the data was recorded in an oracle database, so, an operator or scientist could look up and analysis the data any time. The security and efficiency of the Ethernet which was built as a ring were not taken into consideration only for simplicity now and will be studied in the next step. In addition, a timing system was also designed and constructed for each sub-system, assuring that the data would be sampled by each controller at the same time and be better analysed in the future. All the sensors and actuators were directly connected to the data acquisition cards or modules of the control system.

The Integration of the Siemens PLCs with EPICS

Siemens PLCs were used as the controllers of the subsystems of Elevating system, Sieving system and Helium-loop system. The “SLS s7 driver” was used as the driver

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-WEP34

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

for the corresponding IOCs after some minor revisions for our own requirements [6]. The revisions involve the driver support, device support, record support and the script files under the principles of EPICS. Ethernet-Serial Server was used because the Siemens PLCs only configured a serial (RS485) module rather than an Ethernet communication module.

Some actuators used for the protection of the target system should be controlled by the well programmed algorithms that had been designed and downloaded in the PLCs initially because it is difficult for the serial or Ethernet interfaces to meet the requirements of some fast protection system. The severity of the potential accidents during the operation of the target system will be evaluated and some protections triggered by hardware directly will be also studied and tested in the next step for the high requirements of some critical accidents.

The Integration of the NI DAQs with EPICS

The NI DAQs were used as the controllers of the Heat exchanging system and Main-loop system. The programming language is Labview which is systems engineering software for applications that require test, measurement, and control with rapid access to hardware and data insights. It offers a graphical programming approach for visualizing every aspect of the application, including hardware configuration, measurement data, and debugging [7]. The Labview reads and writes data by bounding the Network published shared variables with the PVs (Process Variable) [8].

Figure 4: A photo of the target system.

In this work, the variables which were related to the other sub-systems were communicated by building a CA Client and CA Server. As mentioned above, the type of all the variables are Network Published Shared variables. A simple test was also performed for the performance of the communication between the sub-systems.

The PID algorithm was used to control the valve of the cooling water and achieve the control of the temperature at the outlet of the heat-exchanger. The hysteresis only using a PID controller is obvious. So, another PD controller with the temperature at the inlet of the heat-exchanger as the input signal was added to avoiding the hysteresis. Finally, it noticeably improved the algorithm and worked well during the experiment. The other physical quantities which were also needed to be controlled automatically and accurately were not finished in this work now due to some objective reasons and will be studied and developed in the next step. The Fig. 4 is a photo of the target system.

CONCLUSION

The experimental target prototype consists of several sub-systems which perform different tasks as the target system running. The development process of the control system for the experimental target prototype is as follows. Firstly, the framework of the control system was constructed and tested based on EPICS which is a distributed control system. Each sub-system was considered as a separate IOC. Different drivers and methods were taken for integrating them with EPICS. Then, the function of data acquisition and control for each sub-systems was finished and the variables needed by other sub-systems were published on the Ethernet as PVs. At the same time, all the variables were recorded into a database ensuring the subsequent analysis. Finally, the algorithm designed for a specific physical quantity was used for the control system. It was finally proved stable and could basically meet the elementary requirements of the spallation target system in ADS.

ACKNOWLEDGEMENTS

This work was finished based on the concerted effort of all the staff of this project. We thank the support of all the staff. This work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDA03030000) and the National Natural Science Foundation of China (Grant No. 11505258).

REFERENCES

[1] Q. Zhao, Z.Y. He, L. Yang, X.Y. Zhang *et al*., "Monitoring method for neutron flux for a spallation target in an accelerator driven sub-critical system", *Chin. Phys. C,* vol. 40, no.7, p. 076203, 2016, doi:10.1088/1674-1137/40/7/076203.

[2] YU Chun-lei, GUO Yu-hui, HE Yuan *et al.*, "Design of Control System for High Intensi ty Proton RFQ Based on Redundancy Technology", Atomic Energy Science and Technology, vol. 48(4), pp.740-745. 2014.

[3] GUO Yu-hui, ZHENG Ya-wei, LONG Yin-dong *et al.*, "Design of Vacuum Control System for Injector II in ADS", Atomic Energy Science and Technology, vol. 46, pp. 539-543, 2012.

[4] https://epics.anl.gov/

[5] EPICS Application Developer's Guide. 2015.

[6] http://epics.web.psi.ch/software/s7plc/

[7] http://www.ni.com/en-us/shop/labview.html

[8] http://www.ni.com/white-paper/14144/en/

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA1

QUEST FOR THE NEW STANDARD PSI IOC PLATFORM

D. Anicic, Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

Abstract

With its four accelerator facilities the Paul Scherrer Institut (PSI) has already several decades of control system Input Output Computer (IOC) experience. The technology is moving forward fast. The older hardware is becoming obsolete: it is slow, consumes too much power, does not match new computing, networking and bus technologies, and replacements can no longer be purchased as models have been discontinued. All this forces us to opt for a new "standard" IOC platform with increasing regularity. What used to be twenty years, became ten, and is now tending towards five years. Here we present past and possible future IOC platforms which we are investigating. Feedback from the conference would be highly appreciated.

PSI ACCELERATOR FACILITIES

The Paul Scherrer Institut (PSI) operates four accelerator facilities. The oldest of these, the High Intensity Proton Accelerator (HIPA) has already been in operation for more than four decades. It has undergone several beam intensity, hardware and software upgrade and rejuvenation processes. The second oldest, Swiss Light Source (SLS) has been operated for almost two decades and is going to be upgraded to SLS2 soon. Third, the Proton Scanning (PROSCAN) medical cancer-treatment facility, with its superconducting, compact, medical cyclotron (COMET) has been successfully treating patients for a decade, and has recently been extended with the third Gantry area. And the newest, Swiss Free Electron Laser (SwissFEL) is just coming in operation.

Although that was not the case in the past, nowadays all facilities are based on the same or similar hardware. There is a common VME bus system in use, with mostly MVME5100 VxWorks and IFC1210 RTLinux [1] Input Output Computers (IOCs). The common control system is EPICS.

MOTIVATION

Our existing processing platforms are getting gradually old and we also have to respond to availability and performance issues. For some of the used components manufacturers have already issued the end of life notices. Some CPU applications are already at the limits of their resource usage, like processing power and memory and bus throughput and the FPGA applications are exceeding available size.

All, especially older, facilities are constantly going through hardware rejuvenation, due to the lack of the spare parts, being upgraded to match higher processing and precision demands, or are simply been extended or adapted to the new user demands.

Probably the biggest demand is driven by the SLS2 upgrade. This should start in year 2020, and the New Processing Platform (NPP) should provide a solution until then.

The NPP project was therefore initiated in December 2017, and the working group began investigating use cases and possible implementations.

GOAL

The design and implementation of the NPP is seen as the necessity, and it has to provide the platform that can commonly be used in all facilities and should equally serve as base in all application areas. Because it is hardly possible and meaningful to use single identical platform for all use cases, it is desirable to provide for scalable solution. Direct consequence of the common platform is optimal usage of available resources. Sharing the manpower, know-how, development tools and expertise between different groups can greatly speed up our work and lower the costs.

USED PLATFORMS HISTORY

HIPA has initially been based on Digital Equipment's PDP-11 and later on hprt743 single board VME computers with HP-RT OS, but they have been replaced, since around year 2000, with MVME5100 on LynxOS. Custom home-made control system was used.

PROSCAN has always been running on MVME5100, initially on LynxOS, with custom home-made control system, same as HIPA.

HIPA and PROSCAN control systems have then been replaced with VxWorks running EPICS [2] as control system, in scope of control system standardisation project at PSI.

SLS is running EPICS since beginning, initially on MVME2300 VME computers. Later MVME5100 were added, and for a few demanding applications MVME6100, too.

For SwissFEL, new platform has been chosen. In cooperation with Swiss company IOxOS [3], the IFC1210 was introduced. Decision was to keep the VME bus, and new was that two FMC cards could be plugged in, and user code could be implemented in on-board FPGA. For many applications FPGA app using one or two FMC modules was enough, and VME bus was only used to supply electrical power.

At the same time it was decided to go for PREEMPT_RT patched Linux (RTLinux) as the operating system, and EPICS is still our control system.

Now the time has come for a new platform.

REQUIREMENTS

As with any other technology, the time comes to upgrade processing platform to the higher level. Several

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA1

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

groups in PSI have been expressing wishes for faster, bigger, more up to date processing capabilities. That's how NPP came to life.

Different groups have already looked for what could come next. The preliminary task, to get the requirements showed that we would have to cover a wide range, from simple and cheap to advanced and expensive. The involved groups, Low Level Radio Frequency (LLRF), PROSCAN Patient Safety System (PaSS), HIPA diagnostics, Digital Beam Position Monitors (DBPM), and Digital Power Supply, together with general purpose controls system group with EPICS IOC produced the following boundary conditions:

- Low latency loops (ADC -> DAC below 1µs)
- General purpose I/O (20 or more)
- Supports VME form factor
- Has dedicated Programmable Logic (PL) memory
- FPGA has at least 200K Logic Elements
- At least 8 gigabit links
- Two Gigabit Ethernet links
- Long term availability
- Costs and effort savings
  - Can reuse existing FMC cards
  - Can reuse VME transition boards
  - Can reuse developed software and firmware

The CPU processing power should well exceed existing IFC1210 performance.

Although the VME was specified as boundary condition, there are many applications which can do without it. It is not to be strictly taken into account.

What was already visible from discussions is that almost everybody was thinking of Xilinx Zynq UltraScale+ MPSoC [4] chips as the base, and one way or the other reach the goal to satisfy own requirements.

Anyway, some other possibilities had to be investigated, too.

POSSIBLE WAYS TO GO

One very obvious and less-effort possibility is to use what is already available. Here we think of IOxOS IFC1410/IFC1420 and IFC1211 boards.

The more preferred would be the use of Silicon on Module (SoM). This solution provides for better scalability and diversity of bus or non-bus carrier boards or embedded systems.

Similar to SoM would be a possibility to design own boards based on UltraScale+ MPSoC. This would be the most expensive and the most resource exhaustive solution. But we would not have to start from scratch, we could base it on Digital Beam Position Monitor (DBPM3) for which the development has already started. This solution does not provide either VME or MTCA.4 [5] bus, at least in this moment.

Already Available Platforms

PSI has started collaboration with European Spallation Source (ESS) in Sweden as In-Kind contributor. ESS decision was to use MTCA.4 as crate standard. IOxOS has developed IFC1410/IFC1420 boards for them. PSI role is to deliver these boards, together with software and firmware packages to ESS. So we could use already working solution. It incorporates T2081 processor and Kintex UltraScale FPGA and provides, as IFC1210, two FMC slots. Initial considerations came to conclusion that switching from VME to MTCA.4 would be radical change. Also, full operating MTCA.4 crate is much more expensive than VME crate. Additionally it seems that MTCA.4 has quite small user and provider community, and most of all there is not much know-how and expertise in PSI. Change would also have consequence on existing front-end electronics, which would have to be redesigned and developed. So, the tendency was to stay with VME, and keep an eye on trends and possible growing availability of MTCA.4 processing and I/O boards.

IOxOS has also developed the IFC1211 board, which is the same as IFC1410, but in VME form factor. That could be very simple replacement for existing IFC1210, but offering only slightly better performance and bigger FPGA resources. This idea was discarded immediately, because many user groups would like to have more than that.

The main penalty by using IOxOS boards is that system firmware is combined with user firmware in the same FPGA. This greatly reduces the number of FPGA Logical elements available for user applications.

On the other hand, it is not Zynq UltraScale+ based, what many users like to have.

SoM Concept

This concept integrates MPSoC, memory and communication interfaces on a pre-engineered printed circuit board (PCB). It represents the complete computer with FPGA, lacking only outside world connectivity. It can be simply plugged into any (compatible) carrier board to give complete solution. It can also be used directly embedded into custom system.

For this we have investigated the Swiss company Enclustra [6] Mercury+ XU1 module mounted on Mercury+ PE1 base board.

The Zynq UltraScale+ seems to be quite powerful and available in different speed and size grades. We would probably go for a quad core Cortex-A53 APU (Application Processing Unit), with dual core Cortex-R53 RPU (Real-time Processing Unit) and 350K Logic elements FPGA.

The main advantage of Zynq UltraScale+ solution is that APU, RPU and FPGA have access to the same memory over AXI bus. Besides low latencies, and therefore higher access speeds (compared to PCIe bus on IFC boards), there is no need to copy data between processing units.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA1

The drawback of this solution is that it would be quite challenging to implement single slot VME or MTCA.4 boards, with proper cooling. Namely, the MPSoC would most probably need own heat dispenser, increasing the board thickness to double-slot.

SoC Solution

We have used the original Xilinx ZCU102 reference board implementation for testing. It was mainly used as test project for evaluating the Zynq UltraScale+ usability and performance.

There are some groups in PSI which are tending towards using SoC on their own custom designed boards. DBPM3 is one of them. Its main purpose was to implement beam position monitors for coming SLS2. It is still in design phase, and that gives us opportunity to rethink and change design to better fit the needs of other groups and applications, too.

Because SoC is going to be directly soldered to the carrier board, there should be no problem of having it as single slot implementation, with proper cooling, making it better choice than using SoM.

The drawback is that it would probably cost more, and the current design does not provide neither VME nor MTCA.4 bus connectivity and form factor.

Decision

The decision is still not made. It is hard to separate from VME, and give up on expertise and re-use of existing VME resources and also know that costs would be higher. On the other hand, we should not give up on possibility to stride the new ways, if indeed the time is ripe.

The final decision is to be made soon.

SOFTWARE

Regardless of the path we take, any new processing platform would also need an operating and control system.

It is quite clear and never taken in consideration to use anything else then EPICS as control system. RTLinux would most probably be used as operating system.

To get the boards booted one also needs the u-boot, board matching device tree, kernel and Linux root file system.

U-boot and Device Tree

U-boot is boot loader usually used on platforms other than Intel x86. It is mostly provided by the manufacturer, because it has to be adapted to match the board. For the same reason the device tree also comes from manufacturer.

For the IFC1410/IFC1420 boards, IOxOS provides the u-boot and device trees. Although it does the job as expected, from our experience it never does exactly what we want. So we have reconfigured and modified the u-boot to match our needs.

For Xilinx reference board we have used the Xilinx petalinux as base. It provides the u-boot and device tree, which we have also reconfigured and modified for our needs.

Enclustra provides the build-root based system. It comes with u-boot and device tree for their boards. Same as for the other boards we have reconfigured and modified the u-boot.

We have not done anything yet for the DBPM3 board, because there is still no prototype available. As being our own custom design, we will have to provide the u-boot and device tree ourselves. This is going to be a challenge, because we have not done such board bring-ups from scratch before. The main idea is to make it as much as possible the same as either Xilinx reference board or like Enclustra boards.

Kernel

The Linux kernel is free and open source code. It supports a lot of architectures and microprocessors. We have expected that there will be not much problems with it. That unfortunately did not happen to be the case.

The problem is that not all drivers are available in the original, mainline, “vanilla” kernel source code. The manufacturers are promising and working on driver integration into mainline kernels, but this happens quite slowly. For that reason we have to stick, at least for beginning, to the manufacturer provided sources.

The second reason is that we would like to use the RT patch, which is not supported by manufacturers. So, if there is no RT patch available for a particular kernel provided by manufacturer, we have to either integrate additional manufacturer drivers into some other kernel, or re-work the RT patches for the manufacturer provided kernel.

For IOxOS boards we have already had kernel v4.1.8-rt8 used for IFC1210 boards. The only additional drivers, the “Tosca” suite, were developed for us by an external company in cooperation with IOxOS. We have ported Tosca drivers to work with IFC1211, and IFC1410/IFC1420 boards, too.

Zynq UltraScale+ boards from Xilinx and Enclustra provide basically the same kernel v4.9.0. We have patched them with PREEMPT_RT 4.9.0-rt1. This seems to work, but we will probably go for newer v4.9 version, with newer RT patch. The biggest problem with this two kernels is that they are not quite the same. Although, we assume, the Enclustra has taken Xilinx kernel sources, they have modified them to work with own boards.

Our idea was to support only one kernel for all Zynq UltraScale+ boards. So we have taken the same kernel sources from the Yocto project with additional Xilinx meta-layer. Unfortunately, it turned out that this kernel, again, differs from both Xilinx and Enclustra kernels. The problem is mainly related to Ethernet driver, but there are also differences in other drivers (clock, flash subsystem and some others). Although we managed to get both platforms boot the same kernel, there are still some limitations. We still have to work on better solution. This step will be done as soon as we start working on kernel for our custom Zynq UltraScale+ board.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA1

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Root File System

From different sources we have received information that using manufacturer root file system is never the best idea. Manufacturers usually provide just basic Linux operating system, just to get the boards booting, and have fast hands-on on using it. Although such basic system can also be reconfigured and rebuilt to support additional features, sooner or later there will be something missing. They almost never provide all possible Linux services, libraries or tools. It would be much better to build own Yocto or build-root Linux system.

For that reason we have chosen to use Yocto as our Linux root file system for Zynq UltraScale+ platforms. We use only Yocto generated root file system, and simply remove any kernel, u-boot and device tree files, which we build afterwards separately.

For the IFC1410/IFC1420 boards we are using original Freescale provided QorIQ-SDK-V2.0, which is anyhow Yocto based, and provides everything we have needed. But we also use only root files system, and we build u-boot, device tree and kernel separately.

Just to mention that our IOCs are always diskless machines, booted over network, with read-only root-over-nfs mounted root file system. Although Yocto has an option to produce the read-only system, from our experience it never does it properly. So, our root file systems are after installing always adapted manually to really work as read-only root-over-nfs.

EPICS

Most IOCs in all four facilities are running EPICS version 3.14.12. The new facilities, such as the upcoming SLS2 and possibly also new projects in other facilities, should introduce newer version of EPICS, most probably EPICS 7. We already have initial installations of EPICS versions 3.15.5 and 3.16.1 as fallback option for EPICS version 7.0.1 (former EPICS4), which is also available. The latest versions are compiling and are still being tested. Not all drivers have been installed yet. Of course, nothing was really tested in operation. The first tests will be done soon in SwissFEL, because there are some users willing to make the step forward.

All these EPICS versions and drivers are available for all used and possible upcoming platforms: MVME5100, IFC1210, IFC1211, IFC1410/IFC1420 and also for any future Zynq UltraScale+ based boards.

The only exception is that PV access (for EPICS 3.16.1 and 7.0.1) is not available for older VxWorks based platforms, but they will anyhow not be used in future.

PERFORMANCE MEASUREMENTS

Performance is not the main factor on choosing our new platform, but it is still significant one. So we have performed different “benchmarks”, just to compare what we have and what we can get. Here we present some benchmarking results as comparison of IFC1210 (dual-core), IFC1211 (octa-core, same as IFC1410/IFC1420) and Zynq Ultrascale (quad-core) boards. All of them are running on 1.2 GHz clock, but with different RT kernels. The performed tests actually measured single core performance. Also the new boards could also run on higher frequencies (IFC boards on 1.8 GHz, UltraScale+ on 1.5 GHz). Also to mention is that UltraScale+ boards also come with the additional single- or dual-core Real-Time CPU (RPU), which could be used to increase the RT performance.

All the tests have always been executed on idle machine, not to influence each other. The tests also do not, as any other benchmark, give really real overall application performance. That’s why we do not present here exact numbers. Approximate numbers will be given here as triplet, for {IFC1210, IFC1211, UltraScale+}.

Cyclictest

Cyclictest [7] is also influenced by used kernel. We have observed mostly identical times of in average 10 microsecond latencies on all three platforms. They slightly deviate in maximum latency, {95, 155, 55}.

iPerf

iPerf [8] measures Ethernet throughput. All three platforms have gigabit Ethernet, and all achieve 930 – 960 Mbit/s as server, and 900 – 925 Mbit/s as client. The CPU usage varies from 60 to 100 percent of one core.

Linpack

Linpack [9] demonstrates the floating point performance for basic mathematical operations. Approximate values are: {220, 520, 530} MFLOPS.

memcpy

Simple, C library ‘memcpy’ function test. It demonstrates not only CPU, but also RAM bus speed. The values are: {200, 500, 2000} MB/s.

Math Operations

Table 1 compares the speed (in microseconds per operation) of various single floating point operations for three different platforms.

Table 1: Single Floating Point Operational Speeds

Platform: / Operation	IFC1210 (µs/op)	IFC1211 (µs/op)	UltraScale+ (µs/op)
sqrt	0.335	0.215	0.050
pow	0.780	0.655	0.360
exp	0.345	0.325	0.150
log	0.765	0.325	0.150
atan/asin/acos	0.350	0.290	0.160
sin/cos	0.280	0.290	0.135

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA1

CONCLUSIONS

Although the future hardware platform have not been chosen yet, it is quite clear that it will be Zynq UltraScale+ based. This decision and start of developments will start very soon.

We are ready from the software point of view. Of course there is still much work to do, but we are confident that software will not be the show stopper.

ACKNOWLEGDMENTS

This paper has been based on work of several groups in PSI. The author's contribution was in bringing up the RTLinux with root file system and drivers, and providing for some performance tests.

The author wishes to thank all groups taking part in specifying, analyzing and testing the different aspects of NPP, especially the NPP representatives, namely B. Kalantari (project leader), W. Koprek, K. Bitterli, E. Johansen, K. Ambrosch, D. Zimoch, R. Kalt, B. Ronner, B. Keil, G. Marinkovic, J. Raabe, M. Brückner, M.Eichin, and G. Theidel.

REFERENCES

[1] RTLinux, https://rt.wiki.kernel.org/.

[2] EPICS, http://epics.org/.

[3] IOxOS, https://www.ioxos.ch/.

[4] Xilinx Zynq UltraScale+ MPSoC, https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html

[5] MTCA.4, https://www.picmg.org/openstandards/microtca/.

[6] Enclustra, https://www.enclustra.com

[7] Cyclictest, https://wiki.linuxfoundation.org/realtime/documentation/howto/tools/cyclictest/start

[8] iPerf, https://iperf.fr/.

[9] Linpack, http://www.netlib.org/linpack/.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA2

DEVELOPMENT OF MicroTCA-BASED LOW-LEVEL RADIO FREQUENCY CONTROL SYSTEMS FOR CERL AND STF

F. Qiu, T. Matsumoto, S. Michizono, T. Miura
Accelerator Laboratory, KEK, Tsukuba, 305-0801 Ibaraki, Japan

Abstract

Low-level radio frequency (LLRF) control systems based on µTCA standard have been developed for facilities such as compact energy recovery linac (cERL) and superconducting test facility (STF) at the High Energy Accelerator Research Organization (KEK), Japan. Three different types of boards were developed according to their different applications. Experimental physics and industrial control system (EPICS) was selected as the data communication system for all of these µTCA boards. The LLRF systems showed good performance during the beam commissioning. This paper presents the current status of the µTCA-based LLRF systems in the cERL and STF.

INTRODUCTION

The compact energy recovery linac (cERL) and the superconducting test facility (STF) are test facilities for the 3-GeV ERL and international linear collider (ILC), respectively, constructed at the High Energy Accelerator Research Organization (KEK), Japan [1, 2]. The cERL is a 1.3-GHz superconducting radio frequency (SCRF) machine that is operated in the continuous-wave mode. The STF (and ILC) is also a 1.3-GHz SCRF facility but it is operated in the pulse mode. In order to fulfil the desired beam quality requirements, the radio frequency (RF) field in the RF cavity of each accelerator needs to be controlled precisely. For cERL, the amplitude fluctuation of the RF field should be maintained at less than 0.1% (rms) and the phase fluctuation at 0.1° (rms). The STF (or the ILC) requires RF stabilities of 0.07% (RMS) and 0.35° (RMS) with regard to the amplitude and phase, respectively. Field-programmable gate array (FPGA)-based digital low level radio frequency (LLRF) feedback (FB) systems have been applied to stabilize the RF field [3]. The digital platform for the LLRF systems was realized based on the micro telecom computing architecture (µTCA) standard. We have developed three types of µTCA board according to their applications.

In this paper, we first introduce the digital LLRF system, and then we present the various µTCA boards applied in the cERL and STF. Finally, we present the performance of the µTCA-based digital LLRF systems during beam commissioning.

LLRF SYSTEM

Figure 1 shows the block diagram of a typical LLRF system [3]. The RF pick-up signals from all the cavities are down-converted to intermediate frequency (IF) signals. The IF signals are sampled in the next stage and fed to a µTCA FPGA to execute the DSP algorithms. The baseband in the phase and quadrature components (I/Q) are extracted from the IF signal by digital IQ detection algorithm. The I/Q signals are fed to a rotation matrix to compensate the loop phase. The vector-sum signal is then obtained by calculating the superposition of all the cavity pick-up signals. After being filtered by digital low-pass filters, the I/Q components are compared with the set-values and the I/Q errors are calculated. Then, the I/Q errors are regulated by a proportional and integral (PI) FB controller. The regulated I/Q signals are added to the feedforward (FF) models. The combined signals are fed to the I/Q modulator by a digital to analog converter (DAC) to modulate the RF signal from the master oscillator. Finally, the LLRF FB loop is closed by means of a high-power source, which drives the cavities.

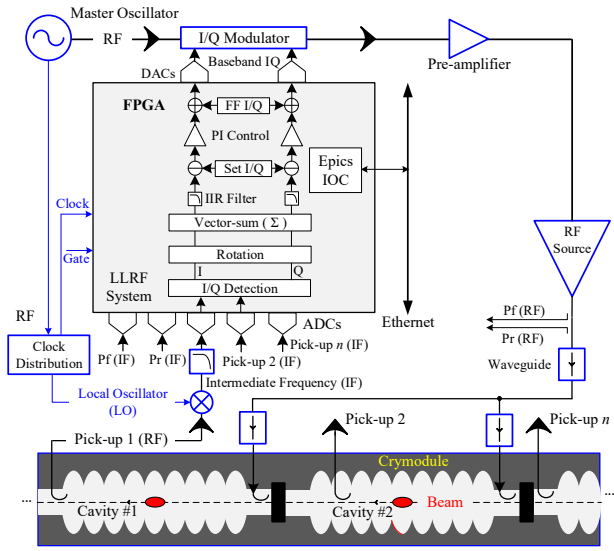


Figure 1: Block diagram of the LLRF system.

µTCA FPGA BOARD

The µTCA FPGA board is the key component of the LLRF system. As shown in Fig. 2, the board integrates ADCs, DACs, an FPGA embedded with power PC (PPC), a digital I/O, and an external trigger. The cavity pick-up signal, forward signal (Pf), reflected signal (Pr), and the reference signal are sampled by the ADCs and fed to the FPGA. The DSP algorithms for the LLRF FB control (see Fig. 1) are implemented in the FPGA. A Linux operation system is installed in the PPC (or ARM). Experimental physics and industrial control system (EPICS) is adopted as the communication protocol. The detailed information about this µTCA board can be found in [4].

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA2

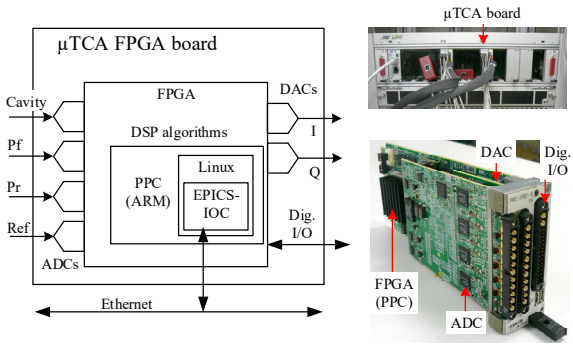


Figure 2: µTCA digital board (type I).

Three types of µTCA boards were developed for the cERL and STF [5]. The µTCA.0 standard type I board and µTCA.4 standard type II boards are mainly used for the LLRF control systems in the cERL and STF, respectively. The main difference between the type I and type II boards is the number of ADCs. We adopted individual cavity control in the cERL (except for the second and third cavities in the injector); therefore, four ADC channels were adequate for our LLRF system. In the STF, we adopted vector-sum control (the vector-sum voltages of the eight cavities were controlled by one controller, see Fig. 1), and accordingly more ADCs were required in the board. The type III board was mainly used for the RF monitor (thus, the DAC is not necessary). The RF signals were directly detected on the board by direct-sampling algorithm without down-conversion (see Fig. 7). The board was equipped with two fast ADCs with sampling frequency up to 400 MSPS [2]. The detailed specifications of these three types of µTCA boards are listed in Table 1 [5].

Table 1: Specifications of the µTCA Boards

TYPE	TYPE I	TYPE II	TYPE III
Facilities	cERL	STF-II	ERL & STF
Function	LLRF	LLRF	Monitor
Standard	µTCA.0	µTCA.4	µTCA.0
ADC	4×16 bits (LTC2208, 130 MSPS)	14×16 bits (AD9650, 105 MSPS)	2×14 bits (ADS5474, 400 MSPS)
FPGA	Virtex-5 FX	Virtex-5 FX	Zynq-700
DAC	4×16 bits (AD9783, 500 MSPS)	2×16 bits (AD9783, 500 MSPS)	N/A
CPU	PPC 440	ARM	PPC 440
OS	Wind River Linux	Xilinx Linux	Wind River Linux

PERFORMANCE

To evaluate the performance of the µTCA LLRF systems of the cERL and STF, the RF field fluctuations were measured by the I/Q detector on the FPGA at the first stage. The achieved field stabilities should be further confirmed by measuring the beam energy fluctuations..

cERL

In the cERL, three two-cell SC cavities were installed in the injector, and two nine-cell SC cavities were installed in the main linac [3]. The beam was accelerated on the crest (i.e., the beam phase φ_b was 0). The measured RF field for the amplitude (left) and phase (right) of the injector and the main linac are illustrated in Fig. 3. Table 2 shows the measured RF stabilities in terms of the different cavities. It should be mentioned that 100-kHz bandwidth digital filters were applied to remove the high-frequency clock jitters.

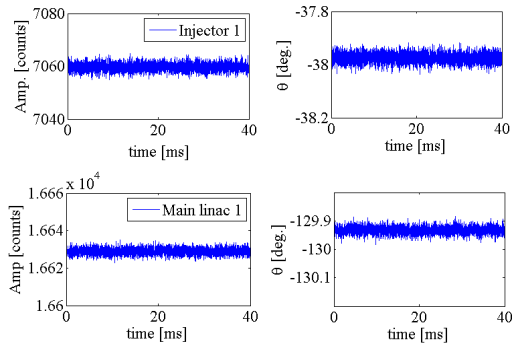


Figure 3: Field stabilities of SC cavities in the injector (top) and main linac (bottom) measured by the µTCA.0 board (type I).

Table 2: RF Stabilities of cERL

Cavity	φ_b	ΔA/A [rms]	Δθ [rms]
Injector 1	0°	0.02%	0.02°
Injectors 2&3	0°	0.02%	0.015°
Main linac 1	0°	0.01%	0.01°
Main linac 2	0°	0.01%	0.01°

To measure the stability of the beam energy, a screen monitor was installed downstream of the bending magnet with a 2.2-m dispersion and a 62.6-µm/pixel resolution. The beam momentum jitter was then calculated by extracting the information of the beam projection on the screen monitor. As shown in Fig. 4, the calibrated beam stability during the measured period of 20 min was approximately 0.0065%.

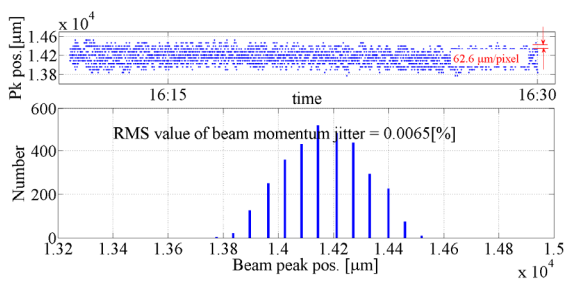


Figure 4: Beam momentum jitter (20 min).

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA2

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

STF

The injector of the STF consists of a photo-cathode RF-gun and two SC nine-cell cavities in a capture cryomodule. In the main linac, 12 SC nine-cell cavities were installed within two cryomodules driven by a 10-MW multi-beam klystron; however, four cavities suffered from degradation because of heavy field emission. Therefore, only eight cavities were operated in the current status [2]. The beam commissioning of the STF will be performed in the year 2019.

The cavities in the STF (and ILC) were operated in the pulse mode with 1.65-ms pulse duration and 5-Hz repetition rate. Figure 5 illustrates the cavity signals measured by the μTCA.4 board (type II) on the first cryomodule of the main linac. Our target was to maintain a stable vector-sum cavity voltage during the flat-top period. Figure 6 shows the vector-sum of the eight cavities. Stability of 0.006% (rms) and 0.024° were achieved in the amplitude and phase, respectively. A digital filter with 250-kHz bandwidth was applied to reject the 8π/9 mode in this measurement [2].

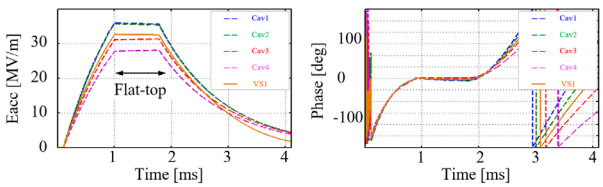


Figure 5: Cavity pick-up signals in the first cryomodule measured by μTCA.4 board (type II).

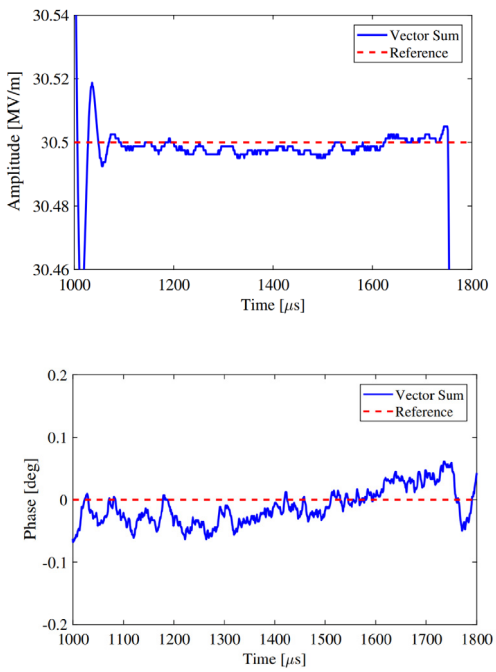


Figure 6: Amplitude and phase of vector-sum for eight cavities during the flat-top period.

An RF monitoring system based on direct sampling technique was installed on the μTCA.0 board (type III), as shown in Fig. 7. The main objective of this system was to evaluate the performance of the fast ADC. The cavity pick-up signal and forward signal was sampled by the fast ADC directly in the absence of down-convertors. By applying the direct sampling algorithm, the amplitude and phase of the RF signals were detected successfully by the type III board. Because the frequency of the ADC input signal was up to 1.3 GHz, the signal-to-noise ratio was higher than in traditional LLRF systems employing down-convertors. The stabilities of the amplitude and phase were approximately 0.1% and 0.1°, respectively. In the STF (and cERL), the system was also used to monitor the long-term drifts of the master oscillator and local oscillator because it is not affected by the characteristics of the down-converters [2].

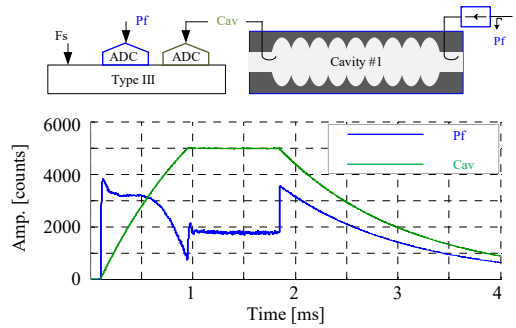


Figure 7: RF monitor system based on the μTCA.0 board (type III).

CONCLUSION

Three types of μTCA boards were developed and applied in the LLRF systems of the cERL and STF. The numbers and types of the digital chips (ADC, DAC, and FPGA) were selected according to the applications of the boards. The EPICS protocol was embedded in the boards for data communication. The performance of the LLRF systems fulfilled the requirements of cERL and STF (ILC).

REFERENCES

[1] M. Akemoto *et al.*, "Construction and commissioning of compact energy-recovery linac at KEK", *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 877, pp. 197-219, Jan. 2018.

[2] S. B. Wibowo *et al.*, "Digital low level rf control system for the International Linear Collider", *Phys. Rev. Accel. Beams*, vol. 21, p. 082004, 2018.

[3] F. Qiu *et al.*, "Application of disturbance observer-based control in low-level radio-frequency system in a compact recovery linac at KEK", *Phys. Rev. ST Accel. Beams,* vol. 18, p. 092801, 2015.

[4] K. Hayashi, K. Akai, H. Ishii, "MicroTCA inter-board data communications applied to BPM and LLRF systems (in Japanese)", in *Proc. 10th Annual Meeting of Particle Accelerator Society of Japan*, Nagoya, Japan, Aug. 2013, Paper SUP096, P.1164.

[5] T. Kobayashi *et al.*, "Applications of MicroTCA in SuperKEKB and KEK Facilities", in *Oral presentation. 5th MicroTCA Workshop*, Hamburg, Germany, Dec. 2016. https://www.caenels.com/wpcontent/uploads/2016/12/PosterProgram.pdf

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA3

A FEEDBACK/FEEDFORWARD SYSTEM AT THE TPS AND ITS COMPONENT PERFORMANCE

C. H. Huang†, P. C. Chiu, K. H. Hu, Y. S. Cheng, C. Y. Wu, K. T. Hsu
NSRRC, Hsinchu 30076, Taiwan

Abstract

For a low-emittance photon light source like the Taiwan Photon Source (TPS), beam stability is a very important property for high-quality photon beams. It is, however, hard to completely remove beam disturbing effects. Therefore, a feedback/feedforward system becomes an effective tool to suppress beam motion. In this report, we discuss the performance of such a system implemented at the TPS. The component performance of the feedback system has been tested to understand its bandwidth limitations.

INTRODUCTION

The Taiwan Photon Source (TPS) is a third-generation light source at the NSRRC with 24 double-bend achromatic cells and a beam emittance of 1.6 nm-rad [1]. To achieve a high-quality beam, the beam motion must be controlled to within 10 % of the beam size or less. The vertical beam size in the center of the straight sections is around 5 μm making it necessary to control the beam motion to better than 0.5 μm.

There are many sources which may adversely affect beam stability[2]. Much effort have been spent to remove or eliminate such perturbations [3]. However, it is hard to remove all kinds of adverse effects. Therefore, a feedback/feedforward system has been developed to counteract beam motions. At first, the components of the feedback/feedforward system are introduced in this paper. The performance and computation model are described in the second part. Third, the performance of the feedback components are discussed and the bandwidth limitations of the fast orbit feedback system is determined.

FAST ORBIT FEEDBACK SYSTEM

There are 24 cells in the storage ring of the TPS, each including seven beam position monitors (BPMs), seven slow correctors and four fast correctors in each cell [4] as shown in Fig 1. The kick angle for the fast horizontal/vertical corrector is 4.8/12 μrad/A, respectively. There are mainly two types of BPMs installed in the storage ring. The k_x/k_y of standard BPMs, which are installed in the arc sections are 13.8/12.73 mm whereas the k_x/k_y of the primary BPMs, installed in the straight sections, are 6.58/8.89 mm. The eight (4 horizontal + 4 vertical) fast corrector power supplies in each cell are controlled by one corrector power supply controller (CPSC).

Libera brilliance+ is used as the BPM electronics. Each BPM platform deals with signal processing of four BPMs resulting in two platforms per cell. The electronics includes custom-written applications with VirtexTM 5 and Virtex 6 to be used for orbit feedback computations. One platform is responsible for the four horizontal correctors per cell and the second platform for the four vertical correctors. The correction rate is 10 kHz and the platform infrastructure is shown in Fig. 2. The bandwidth of the FOFB, shown in Fig. 3, is around 300 Hz in both the horizontal and vertical plane [5].

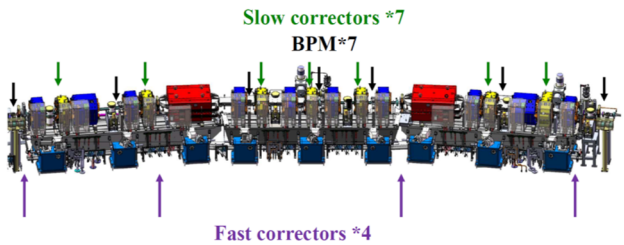


Figure 1: The location of beam position monitors (BPMs), slow and fast correctors in each cell.

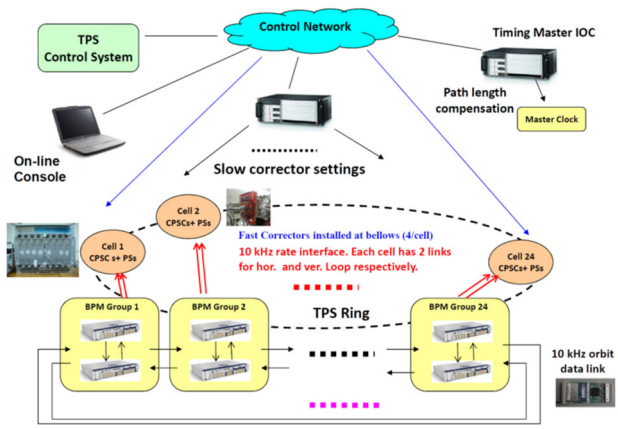


Figure 2: The infrastructure of the fast orbit feedback system.

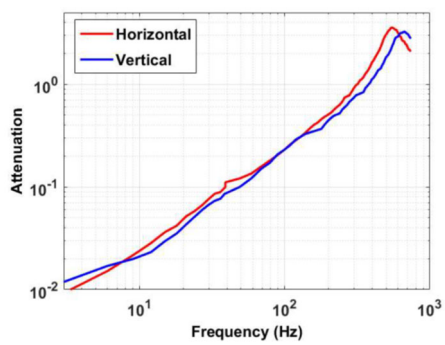


Figure 3: Frequency response of the fast orbit feedback system.

† huang.james@nsrrc.org.tw

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA3

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

RADIO FREQUENCY ADJUSTEMENT

As the FOFB is applied to correct beam motions or orbit drifts, we observe that the output current of the second and third fast corrector in each cell varies significantly compared to the other two due to the correction of path length changes. Although the FOFB system can compensate for path length changes during a short time period, the orbit drifts still increase by tens of microns per day of operation. Therefore, it is necessary to adjust the radio frequency to compensate for path length changes.

The radio frequency (RF) adjustment, as shown in Fig. 4, uses all fast horizontal corrector currents, $\Delta \boldsymbol{I}$, as obtained by the FOFB, and converts them to orbit drifts by the response matrix($\boldsymbol{R}$) at one Hz. With the dispersion matrix $\boldsymbol{D}$, the RF frequency change $\Delta \boldsymbol{F}$ [6] is given by

$$\Delta \boldsymbol{F} = \boldsymbol{D}^{-1}\boldsymbol{R}\Delta \boldsymbol{I}. \quad (1)$$

The RF frequency adjustments shows two peaks per day, as shown in Fig. 5, due to the earth tide. The radio frequency trends higher day by day due the decrease of ambient air temperature.

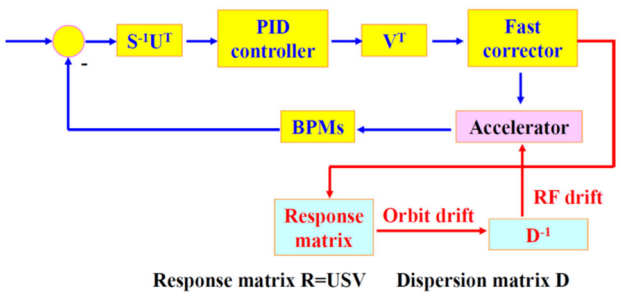


Figure 4: Scheme of the radio frequency (RF) adjustment.

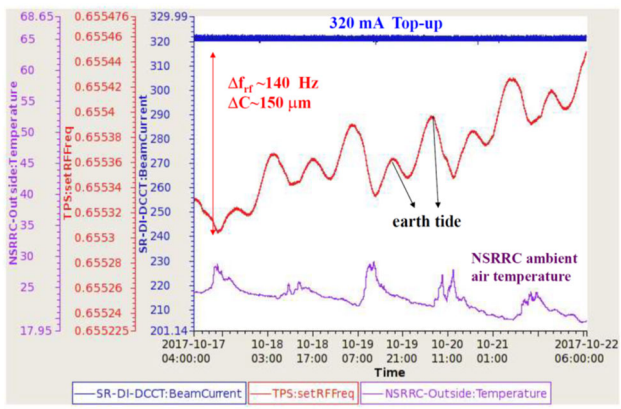


Figure 5: Radio frequency and ambient air temperature as a function of time.

PERFORMANCE OF THE COMPONENTS OF FEEDBACK SYSTEM

The 60 Hz beam motion in the vertical direction is more significant (~ 5 μm) than at other frequencies due to the RF transmitter and cooling fans for BPMs and nearby bellows [3], as shown in Fig. 6. In order to eliminate such 60 Hz beam motion, a feedforward correction is implemented because the 60 Hz perturbations are quite stable. At first, the beam motion, $\boldsymbol{b}(t)$, for all BPMs is recorded for one second simultaneously. The beam motion at 60 Hz in each BPM can be obtained by a Fourier transform of the beam position $\boldsymbol{B}(60) = \int_0^T \boldsymbol{b}(t)e^{-i2\pi*60*t}dt$ and the result is generally a complex number. Applying the inverse response matrix $\boldsymbol{R}^{-1}$ to the 60 Hz beam motion, $\boldsymbol{B}(60)$, we get the corrector strength to eliminate the 60 Hz beam motion from $\boldsymbol{C}(60) = -\boldsymbol{R}^{-1}\boldsymbol{B}(60)$. Because of unavoidable errors in the orbit and response matrix measurements singularities can appear resulting in unreasonably large and competing corrector strengths caused mostly by noise rather than error sources. We use therefore singular value decomposition (SVD) to rewrite the response matrix as $\boldsymbol{R} = \boldsymbol{USV}$ [7] and properly choose a singularity rejection parameter. The corrector strengths are then $\boldsymbol{C} = -\boldsymbol{V}^T\boldsymbol{S}^{inv}\boldsymbol{U}^T\boldsymbol{B}$, where the corrector strength would be a complex number as well and written as $C_j exp(i\phi_j)$. Here C_j is the amplitude of the j^{th} power supply and ϕ_j is the related phase [8].

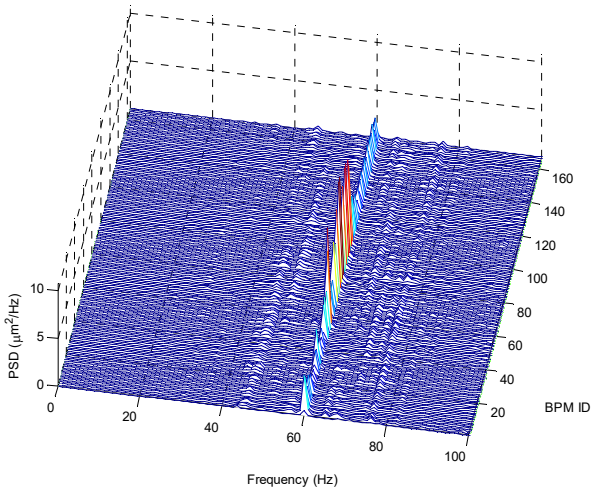


Figure 6: Power spectral density of the vertical beam motion.

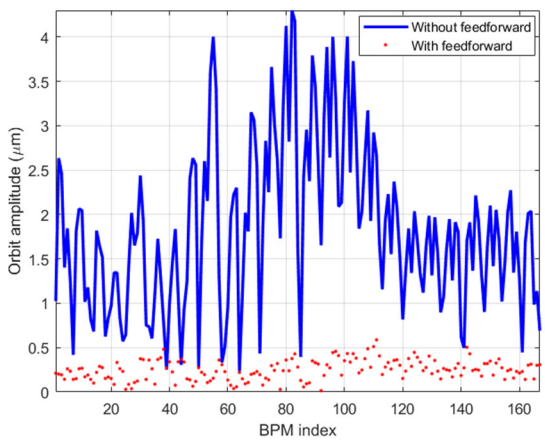


Figure 7: 60 Hz beam motion with and without feedforward correction.

With the correcting power supply generating a waveform, i.e. $C_j \cos(2\pi*60*t+\phi_j)$, the 60 Hz beam motion is corrected. Note, that the 60 Hz beam motion comes from the mains powering the electronics. The mains frequency is not exactly 60 Hz and changes depending on the total load. Therefore, the waveform should be locked to the

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA3

phase of the mains. Figure 7 shows that the 60 Hz beam motion is reduced to 1/10 with feedforward correction compared to no feedforward correction.

COMPONENT PERFORMANCE OF THE FOFB SYSTEM

In the second section we discussed that the bandwidth of the FOFB is roughly 300 Hz in both planes. In order to understand the limitation of this feedback system, the performance of each component, i.e. power supply, power supply controller, magnet, vacuum chamber and BPMs must be evaluated. A sweeping sine wave is generated by a waveform generator and sent into the controller of the power supply, as shown in Fig. 8, to measure the frequency response of the fast corrector and vacuum chamber. As the output current and the magnetic field is sent into the same analog-to-digital convertor (ADC), the magnetic field response and phase lag with respect to the output current of the power supply can be obtained.

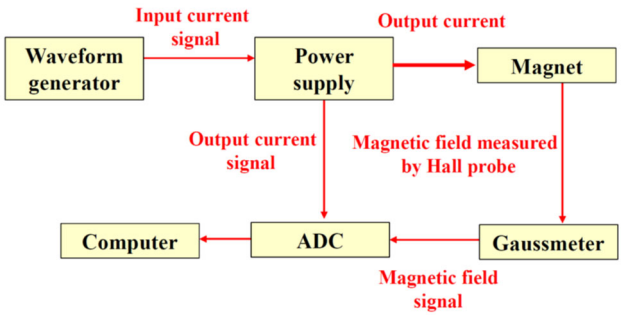


Figure 8: Block diagram for the measurement of dynamic magnetic field response.

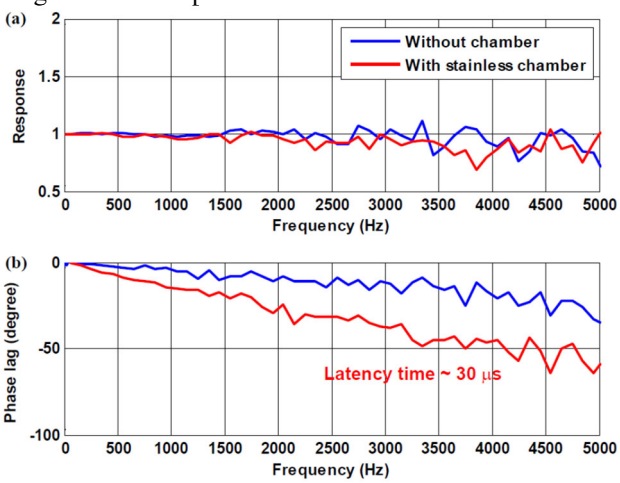


Figure 9: Frequency response (a) and field phase lag (b) of the fast correctors with and without vacuum chamber relative to the power supply output current.

From the frequency response of the fast correctors in Fig. 9, we find that the amplitude attenuation is quite low below 5 kHz. A phase lag can be observed and from $\phi=2\pi ft$, where ϕ, f and t are phase lag, frequency, and time lag, respectively, we estimate the latency time to be ~ 30 μsec for the magnetic field measured in the stainless vacuum chamber..

Homemade fast corrector power supplies [9] are used in in the FOFB. Proportional integral (PI) controls are used in these power supplies. As the K_p parameter is increased, the bandwidth of the power is higher and the phase lag decreases, as shown in Fig. 10. However, at high K_p-values, such as 1.5, a resonance can be observed above 5 kHz in a constant current output, which leaves an ample operational range to adjust the power supply for a high bandwidth, low phase lag and high stability.

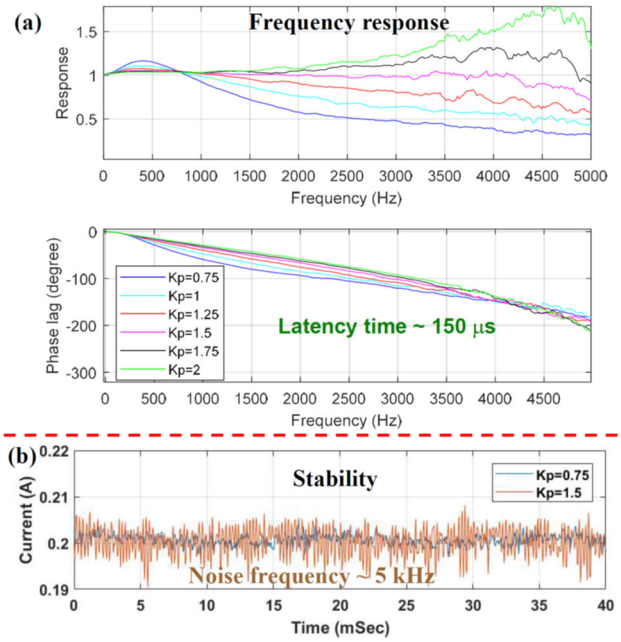


Figure 10: Performance (a) and stability (b) of the fast corrector power supplies for different K_p.

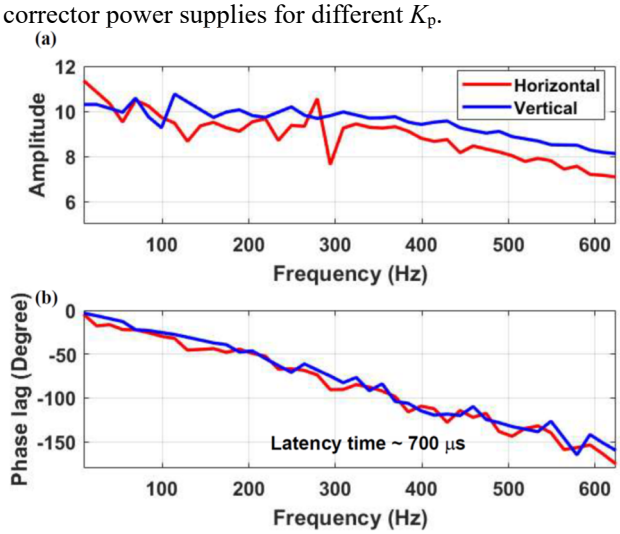


Figure 11: Frequency response and phase lag of the beam with respect to the current setting of the power supply.

The bandwidth and processing latency of the BPM electronics is approximately 2 kHz and 270 μsec and the latency of the CPSC is around 100 μsec. As to the overall contribution to the beam, we observe that the amplitude attenuation is low for frequncies from 0 Hz to 600 Hz. The phase lag, though, is quite large and approximately 150° at 600 Hz. Therefore, the reduction of the phase lag is important for a large bandwidth of the FOFB.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA3

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

CONCLUSION

To eliminate beam motion, a fast orbit feedback (FOFB) was implemented after initial beam commissioning. The bandwidth is about 300 Hz in both planes. To minimize orbit distortions, caused by a path length change due to the earth tide or ambient air temperature, the radio frequency is adjusted as follows. Because a vertical 60 Hz beam motion is dominant after applying the FOFB, a feedforward correction was proposed and put in operation this year.

From the analysis of the FOFB system components, the amplitude attenuation of the magnetic field detected by the beam is negligible as the frequency of the alternating current given by the power supply is increased. However, the phase lag should be reduced to increase the bandwidth of the FOFB.

REFERENCES

[1] C.C. Kuo, *et al.,* "Commissioning of the Taiwan Photon Source", in *Proc. IPAC'15*, Richmond, VA. USA, May 2015, paper TUXC3, pp. 1314–1318.

[2] R.O. Hettel, "Beam stability at light sources", *Rev. Sci. Instrum*. 73 (2002) 1396-1401.

[3] C. H. Huang, *et al.,* "Study of 60 Hz beam orbit fluctuations in the Taiwan Photon Source", in *Proc. IPAC'*17, Copenhagen, Denmark, May 2017, paper TUPAB107, pp. 1566–1569.

[4] P. C. Chiu, *et al.,* "Fast orbit scheme and implementation for TPS", in *Proc. IPAC'*13, Shanghai, China, May 2013, paper TUOCB202, pp. 1146–1148.

[5] P. C. Chiu, *et al.,* "Preliminary beam test for TPS fast orbit feedback system", in *Proc. IPAC'*16, Busan, Korea, paper WEPOW04, pp. 2930–2932.

[6] P. C. Chiu, *et al.,* "Orbit correction with path length compensation based on RF frequency adjustments in TPS", in *Proc. IPAC'*17, Copenhagen, Denmark, May 2017, paper TUPAB103, pp. 1553–1555.

[7] Y. Chuang, *et al.,* "Closed orbit correction using singular value decomposition of the response matrix", in *Proc. PAC'93*, Washington, USA, May 1993, pp. 2263–2265.

[8] C. H. Huang, *et al*., "Methods to detect error sources and their application at the TPS", in *Proc. IPAC'*18, Copenhagen, Denmark, May2017, paper WEPAL057, pp. 2305–2308.

[9] K. B. Liu, "Rigorous mathematical modelling for a fast corrector power supply in TPS", Jinst. 12 (2017) T04004.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA4

DEVELOPMENT OF A NETWORK-BASED TIMING AND TAG INFORMATION DISTRIBUTION SYSTEM FOR SYNCHROTRON RADIATION EXPERIMENTS AT SPring-8

T. Masuda†, Japan Synchrotron Radiation Research Institute (JASRI), Sayo, Hyogo, Japan

Abstract

Time-resolved measurements in synchrotron radiation experiments require an RF clock synchronized with a storage ring accelerator and a fundamental revolution frequency (zero address) signal. For using these signals around the experimental station, long RF cables from the accelerator timing station, divider modules, and delay modules must be deployed. These installations are costly and require significant effort to adjust the timing by experts. To lower these costs and efforts, the revolution frequency, which is ~209 kHz at the SPring-8 storage ring, and tag information distribution system have been studied based on a high-precision time synchronization technology over a network. In this study, the White Rabbit technology is adopted. The proof of concept consists of a master PC, a slave PC, and two WR switches. The master PC detects the zero-address signal and distributes time stamps with tag information to the slave PC. Then the slave PC generates ~209 kHz signals synchronized with the target bunch by adding the offset time calculated by software. The output signals from the slave PC achieved a measured one-sigma jitter of less than 100 ps.

INTRODUCTION

High-precision timing signals synchronized with a storage ring accelerator are indispensable for time-resolved measurements in synchrotron radiation experiments utilizing short pulse characteristics of the synchrotron radiation. These timing signals have been introduced into the required beamline by deploying long distance RF cables from the accelerator timing station. In this method, beamlines that can utilize the precise timing signals are quite restricted, as their expansion is difficult. Furthermore, introducing these RF signals demands high cost and effort since various types of electronics have to be deployed and adjusted to divide the frequency or to tune the phase.

In order to optimize experimental results, it is quite common to pursue a more appropriate timing distribution system with higher expandability. In other words, distribution of the timing signal should be processed as universal digital information closely coupled with a control system, instead of effecting direct distribution of the analog RF signal. For any beamline that requires precise timing signals, experimental users should be able to employ and manipulate these signals easily by means of GUI programs in a control system.

I have studied a timing distribution system based on a high-precision time synchronization technology over a standard network. Since the system delivers "digital information" related to timing signals over the network, the system can be handled and expanded without difficulty. The timing signal generated based on this information can be precisely and easily adjusted by the software. In addition, other useful information can be attached easily such as a shot number.

As a high-precision time synchronization technology, I have adopted White Rabbit (WR) [1], which is promoted as an international collaborative project. It extends the IEEE 1588 standard and achieves highly precise synchronization with sub-nanosecond accuracy by using clock synchronization at the hardware layer of Synchronous Ethernet and phase detection based on digital dual-mixer time difference. The WR technology is opened to the public at the Open Hardware Repository (OHR) [2], and any information required for development is freely available in accordance with open licenses such as the CERN Open Hardware License [3].

To conform the digital-based timing distribution system, I have built the proof of concept (PoC) system with the minimum configuration. In the case of SPring-8, the aim of the PoC system is to realize generation of a 208.8 kHz revolution frequency signal synchronized with the target bunch of the storage ring, which was accelerated by 508.58 MHz RF, and deliver tag information, such as a shot number.

BUILDING THE POC SYSTEM

The PoC system has been constructed as the first step for verification of the new timing and tag information distribution system, as shown in Fig. 1. The system consisted of a master PC, a slave PC, two WR switches (WR-A and WR-B), a GPS receiver, and single mode fibers (SMFs).

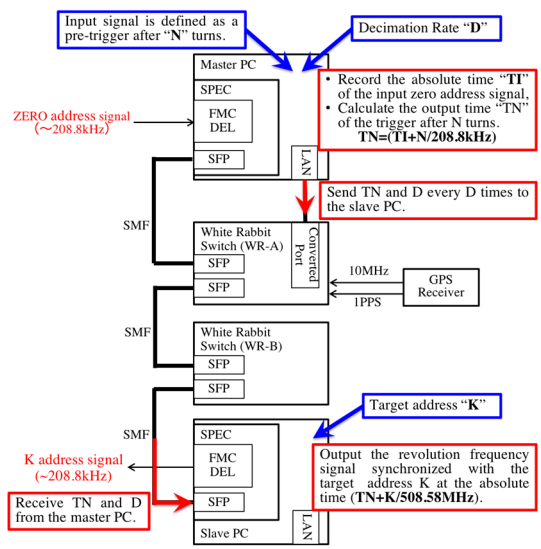


Figure 1: A block diagram of the PoC system.

† masuda@spring8.or.jp

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Both the master and slave PCs are equipped with FMC DEL 1 ns 4cha [4] (Fig. 2), which is a fine delay FPGA Mezzanine Card (FMC), combined with a PCI Express FMC carrier card called SPEC [5] (Fig. 3) equipped with Xilinx Spartan-6 [6]. FMC DEL 1ns 4cha has one trigger input channel and four output channels with a user-programmable fine-delay function. It is operated by three modes: pulse delay, pulse generation and time-to-digital converter modes.

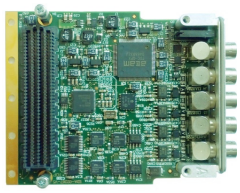


Figure 2: Picture of an FMC DEL 1ns 4cha SPEC card.

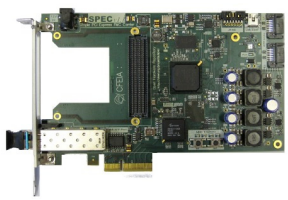


Figure 3: Picture of a SPEC board.

By installing WR-core firmware into the FPGA of the SPEC, both the master and slave PCs served as WR nodes. After a 10 MHz clock and a 1 PPS signal input from the GPS receiver to the WR-A switch, the master and slave PCs, as well as the WR switches were synchronized with the GPS via the SMF. Ubuntu [7] 12.0.4 LTS was adopted as the OS for both PCs.

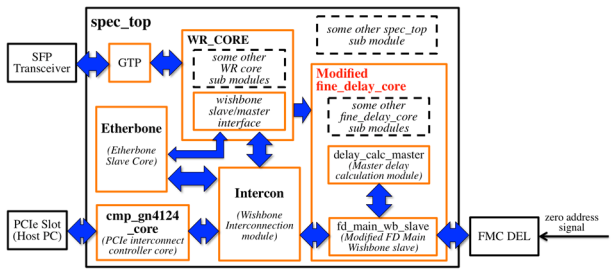
The master PC received the 208.8 kHz zero-address signal, i.e., the fundamental revolution frequency signal. This signal was defined as the pre-trigger after N turns at the slave side, where N was the programmable parameter at the master side. The master recorded the absolute time T_I of the received zero-address signal and calculated the output time T_N of the trigger after N turns.

$$T_N = T_I + \frac{N}{208.8\ kHz} \quad (1)$$

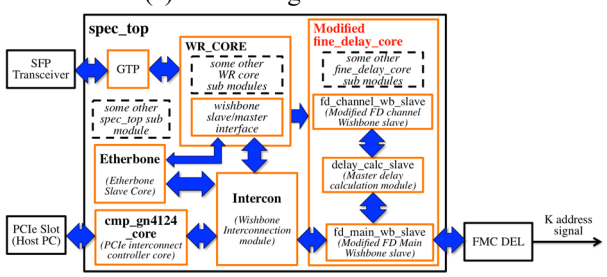
The master PC transmitted T_N and D for every D value to the slave PC, where D denotes the decimation rate of the transmission, and is assigned by the software. The slave PC received the values of both T_N and D, and output the revolution frequency signal synchronized with target bunch address K at the absolute time T_K defined as follows.

$$T_K = T_N + \frac{K}{508.58\ MHz} \quad (2)$$

The slave PC interpolated the output decimation by means of the pulse-train generation function of FMC DEL 1ns 4cha and realized the continuous output of the 208.8 kHz revolution frequency signal. Figure 4 shows the timing chart of the consecutive processes on the master and slave PCs. Figure 5 illustrates the block diagrams of the developed FPGA logics to achieve the functions shown in Fig. 4. These functions were mostly embedded into *modified_fine_delay_cores* in both the master and slave PCs, which were newly developed to fulfill the original functions as mentioned above.



(a) FPGA logic for a master PC.



(b) FPGA logic for a slave PC.

Figure 5: Block diagrams of FPGA logics.

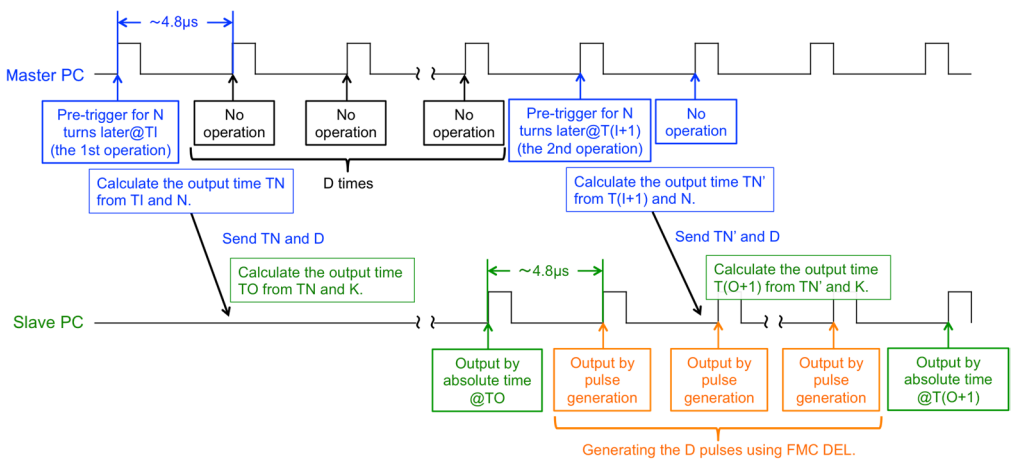


Figure 4: Timing chart of consecutive processes on the master and the slave PCs.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA4

When development of the PoC system was initiated, T_N and D could not be delivered directly from an SFP module of the SPEC board due to the limitation of FPGA logic on the master side. Therefore, the master sent this information from the LAN port on the PC via software. Here, only the case of $D > N$ was considered in order to simplify the logic construction.

Since utility software was prepared to change the output frequency, delay, pulse width, etc., in addition to the required parameters for the FPGA logic operation, it was relatively easy to adjust the output signal in the PoC system.

EVALUATION OF THE POC SYSTEM

Evaluation measurements of the PoC system were carried out on the test bench. Figure 6 shows a photo of the measurement environment of the system. For simplicity, a synthesizer substituted the GPS receiver because the system did not need to be synchronized with the coordinated universal time (UTC) in the evaluation phase. A 208.8 kHz signal simulating the zero-address signal was generated by dividing 508.58 MHz from the synthesizer by the harmonic number 2436. A 10 MHz clock from the synthesizer was input to the WR-A switch.

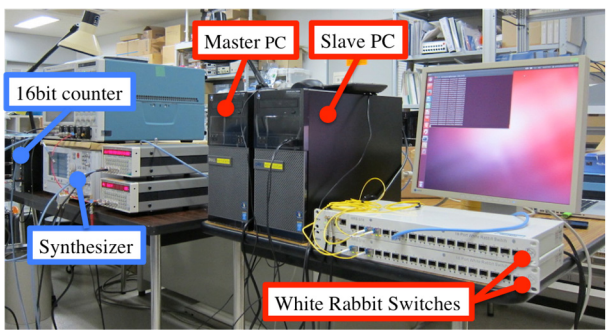


Figure 6: Measurement environment of the PoC system.

Jitter Measurement of the Output Signal

After fine-tuning of the output frequency in efforts to minimize the jitter, I measured jitter of the 208.8 kHz signals from the slave PC in the case of $D = 1000$ and $N = 900$. Figure 7 shows the measurements results. The data indicate presence of unnatural structures in the output jitter, and large jitter beyond the ±2 ns was observed. Standard deviation σ of the entire jitter was about 250 ps, while it was only about 170 ps for the center structure alone.

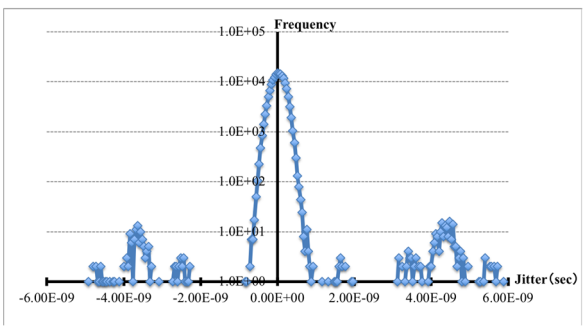


Figure 7: Result of the jitter measurement.

Measurement of the Decimation-Rate Dependence of Jitter

Table 1 shows results of the measured jitter with varying decimation rate D. As expected, the results show that jitter decreases with decrease in D. Therefore, D should be reduced as much as possible to lower the jitter. However, adopting smaller D results in a network traffic increase. The range $D = 600 – 1000$ is considered practical because the system hung up when a smaller D was specified. These values were larger than those I initially assumed.

Table 1: Decimation Rate D Dependence on the Jitters

Parameters	σ (Whole)	σ (Center)	Number of Samples
D = 1000, N = 900	254 (ps)	171 (ps)	163,233
D = 600, N = 500	221 (ps)	139 (ps)	68,321
D = 300, N = 200	201 (ps)	131 (ps)	71,249

INVESTIGATION AND COUNTERMEASURE OF THE PROBLEMS

As the evaluation results mentioned in the previous section, the PoC system had following problems:

1. Output signal from the slave was sometimes missing.
2. Large jitter beyond ± 2 ns was sometimes observed.
3. There was the limitation of $D > N$ in the operation parameters.

In particular, 1 and 2 would pose serious problems when the system is set into actual operation. Therefore, I investigated the cause of these two problems with the goal of solving them.

Lack of Output Signals

As the result of detailed examinations, I found that the lack of the output signal that sometimes occurred the consequence of an execution delay of software which was running on the master PC, and was responsible for transmitting the output time T_N and the decimation rate D.

In order to solve this problem, the FPGA logic on the master side should perform data transmission as originally planned. Fortunately a sample version of the Etherbone [8] master core, which was a missing piece at the construction phase, was released by OHR at that timing, and I planned to improve the PoC system by employing it.

Large Jitter Beyond ±2 ns

In order to investigate the cause of the large jitter beyond ± 2 ns, the jitter was measured again using a new FMC DEL 1ns 4cha card. The results of this measurement are shown in Table 2 and Fig. 8. The large off-center jitters as shown in Fig. 7 were absent.

After numerous measurements carried out by using the new FMC DEL 1ns 4cha card for an extended period of time, the big jitter reappeared. At some time, I noticed that the FMC DEL card was heating up, and the core temperature of the FPGA had surpassed 70 °C. This phenomenon was documented in articles related to the heat problem of

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

the FMC DEL 1ns 4cha on the OHR website. Upon removal of a side panel of the PC and cooling of the FMC with the use of an electric fan from the outside, the large jitters disappeared. Subsequently, I decided to attach a small fan on the SPEC board in order to cool the FMC DEL 1ns 4cha continuously according to the design guide provided on the OHR website.

Table 2: Result of the Jitter Measurements with a New FMC DEL 1 ns 4cha

Conditions	Ave. (ps)	Min. (ps)	Max. (ps)	σ (ps)	Num. of Samples
D = 300, N = 200	20.4	-522	595	127.7	121.2k

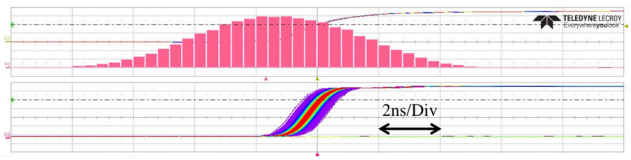


Figure 8: Hardcopy of an oscilloscope display at the jitter measurement with a new FMC DEL 1ns 4cha.

IMPROVEMENT OF THE POC SYSTEM

In order to solve the problems described in the previous section, the PoC system has been improved as follows.

Realization of Data Transmission by FPGA

In order to prevent the lack of the output signal from the slave, the Etherbone master core has been integrated into the FPGA logic for the master as illustrated in Fig. 9. The red rectangle illustrates the modified part from the previous version in Fig. 5(a). At start, I thought that this function could be realized by implementing only the Etherbone master core, but in fact an Etherbone slave core was required as well. By installing the Etherbone master core, the master was able to transmit data directly from the SFP module on the SPEC board without software assist. Figure 10 illustrates the block diagram of the improved PoC system.

Implementation of FPGA Logic in the Case of $D < N$

As mentioned in the previous section, the $D < N$ case functionality was originally not implemented in the initial PoC system for the sake of logic simplicity.

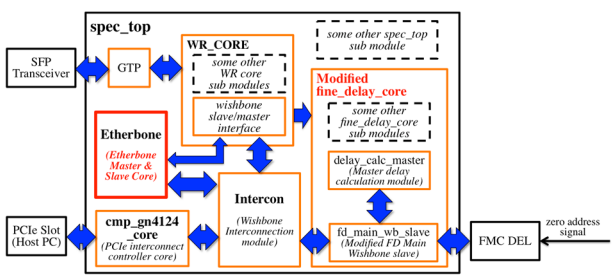


Figure 9: Block diagram of the improved FPGA logic for the master PC.

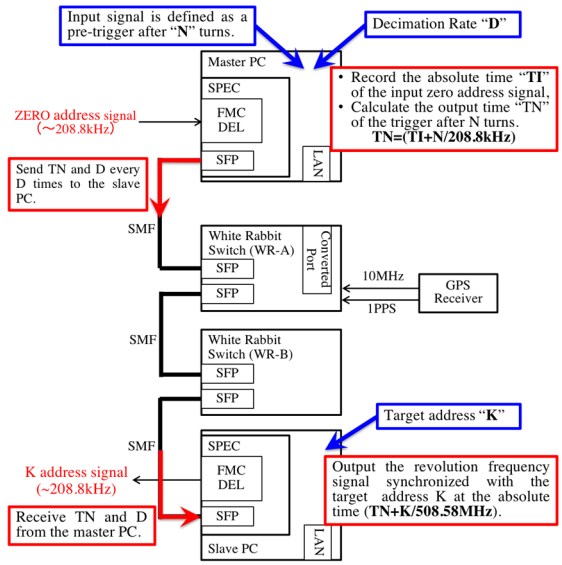


Figure 10: Block diagram of the improved PoC system.

Implementation was difficult because the slave received the data transmitted from the master before the first signal output.

In the improved PoC, the logic for this case has been successfully integrated into the system. This functionality makes it possible for the PoC system to specify smaller D values, and smaller jitter is expected.

Installation of a Cooling Fan on SPEC

In order to solve the heat problem of FMC DEL 1ns 4cha, which exacerbates the jitter, a cooling fan was installed on the SPEC board according to the OHR design guidelines.

EVALUATION OF THE IMPROVED POC SYSTEM

Evaluative measurements of the improved PoC system were performed with the same setup as shown in Fig. 6. The system has serious difficulties in stopping output from the slave PC within a few milliseconds, except in the case where target bunch address $K = 0$ and fine delay time $F = 0$ in the operation parameters of the slave.

Jitter Measurement of the Output Signal

Evaluation of the improved PoC system is limited due to the stability problem described above. Nevertheless, jitter of the 208.8 kHz output signal from the slave was measured while varying the parameter D on the master. The operation parameters of the slave were fixed to $K = 0$, $F = 0$ and fine tuning of output frequency = -31.2 ps. Table 3 shows results of the jitter measurements. Figure 11 shows the hardcopy of an oscilloscope display in the case where $D = 50$ and $N = 1000$.

A standard deviation of measured jitter below 100 ps is achieved thanks to the smaller D enabled in the improved system. This result indicates that performance of the jitter is improved by basing the output on absolute time as much as possible instead of the complementary pulse-train generation function of the FMC DEL 1 ns 4cha.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA4

Table 3: Results of Jitter Measurement with the Improved PoC System

Conditions	Ave. (ps)	Min. (ps)	Max. (ps)	σ (ps)	Num. of Samples
D = 1000, N = 1000	4.89	4.24	5.53	155.2	108.2k
D = 50, N = 1000	5.06	4.55	5.62	97.33	458.3k

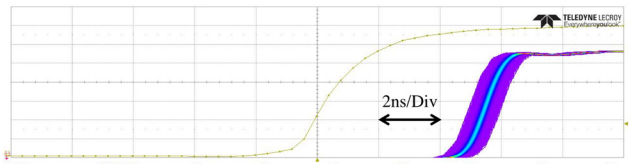


Figure 11: Hardcopy of an oscilloscope display at the jitter measurement with the improved PoC system.

Stability of the Improved PoC System

As mentioned in the beginning of this section, the improved PoC system has a severe stability problem. For example, in the case where $K = 0$ and $F = 0$, which is a unique condition that enables the PoC system to generate output from the slave PC, there was an occasion where the system stopped immediately, while on another occasion the system ran continuously for over 18 hours. Additionally, there was an occasion where the slave PC hung up during the test.

Furthermore, the PoC system has not become functional after a series of measurements with any parameter choice. I suspected malfunctioning of hardware, including SPEC board, FMC DEL 1ns 4cha card and the PC, and replaced all of the parts with new ones. However, the improved PoC system did not perform. The stability issue persists.

FUTURE PLAN

I have recently succeeded in acquiring a new fund to study the 508.58 MHz clock distribution over the WR network. In the time-resolved measurement of the synchrotron radiation experiment, there are many opportunities to utilize the 508.58 MHz clock rather than the 208.8 kHz zero-address signal. Since the possibility of the practical use of RF clock distribution over WR technology is already shown in the proof of concept system referenced in [9], I aspire to apply it to the 508.58 MHz clock of the SPring-8 storage ring. Figure 12 shows an overview of the system that I plan to build.

In this study, distribution of the 208.8 kHz revolution frequency signal is also projected for integration into the future system. Since I regard that the 208.8 kHz signal distribution will be realized based on the PoC system, the PoC system will correspondingly be debugged to fix the stability problem. However, considering that debugging of the FPGA logic is quite difficult, I also will have to take into account other approaches for implementation to the system. One of these approaches would be to count the RF clock up to the harmonic number 2436 locally at the slave side and to reset the local counter simultaneously at the absolute time transmitted from the master.

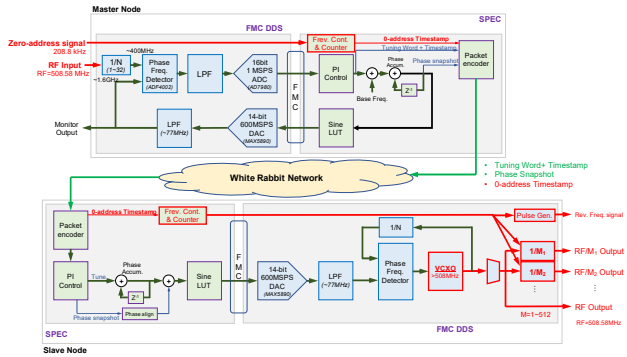


Figure 12: Overview of a planned system for distribution of the 508.58 MHz clock as well as the 208.8 kHz revolution frequency signal over WR network.

SUMMARY

I have developed the PoC system, which distributes the 208.8 kHz revolution frequency signal synchronized with the dedicated electron bunch of the SPring-8 storage ring, with the aim of building a trigger and tag information distribution system based on a high precision time synchronization technology.

The initial PoC system has achieved an output jitter of 127 ps (1σ) at the decimation rate $D = 300$. In the realistic parameter range of $D = 600 – 1000$, the resulting jitter was 140 – 170 ps (1σ). Since the master has transmitted information such as absolute time by software, output signals from the slave PC were sometimes dropped.

The improved PoC system has successfully realized automatic transmission of the information by means of hardware, i.e., by integrating an Etherbone master core in the FPGA logic in the master. This improvement has made it possible to operate the system at smaller decimation rates, by which consequently the system accomplished a jitter below 100 ps (97 ps (1σ)). However, the improved PoC system has a severe stability problem that remains unresolved.

At present, I have commenced the study of a 508.58 MHz clock distribution over the WR network. In this new study, I plan to integrate the distribution of 208.8 kHz revolution frequency based on the PoC system as well.

REFERENCES

[1] https://www.ohwr.org/projects/white-rabbit

[2] https://www.ohwr.org

[3] https://www.ohwr.org/projects/cernohl

[4] https://www.ohwr.org/projects/fmc-delay-1ns-8cha

[5] https://www.ohwr.org/projects/spec

[6] https://www.xilinx.com/products/silicon-devices/fpga/spartan-6.html

[7] https://www.ubuntu.com

[8] https://www.ohwr.org/projects/etherbone-core

[9] T. Włostowski *et al.*, "Trigger and RF Distribution Using White Rabbit", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, pp. 619-623. doi:10.18429/JACoW-ICALEPCS2015-WEC3001

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA5

RETHINKING PLCs: INDUSTRIAL ETHERNET FOR LARGE-SCALE REAL-TIME DISTRIBUTED CONTROL APPLICATIONS

B. Ploetzeneder, O. Janda, A. Krucenko, J. Trdlicka, P. Pivonka, P. Bastl
ELI Beamlines/Institute of Physics of the ASCR, Prague, Czech Republic

Abstract

Many research facilities rely on PLCs to automate large slow systems like vacuum or HVAC, where price, availability and reliability matter. The dominant architecture consists of local units of controllers/modules (programmed in IEC61131-3 languages), which operate mostly autonomously from a SCADA layer.

While some vendors provide low-level stacks to encourage growth of their ecosystems, PLC programming remains largely within a closed, proprietary world.

In this paper, we introduce a different way of thinking about PLC hardware.

Working with the open stacks intended for the design of new EtherCAT (Beckhoff)/Powerlink (B&R) modules, we built an abstract C++- API to control the existing ones. These industrial Ethernet busses can be propagated using standard network hardware, so any RT-Linux system can now control any PLC module from anywhere in our facility using high-level languages (C++, LabVIEW).

This way, PLC modules are seamlessly integrated into our distributed TANGO-based control system. PC-PLC interfaces are no longer needed; or in the case of traditionally implemented subsystems, trivial.

BACKGROUND

Programmable logic controllers (PLCs) are cheap, reliable, modular and fault-tolerant "hard" real time control solutions, even in challenging environments. They are easy to program and maintain, and allow online modification of hardware/software.

With a few notable exceptions [1], most facilities rely on a standard architecture shown in Figure 1: Local units (processors and modules) are distributed in the field, and operate almost autonomously from a SCADA layer, interacting using thin communication links (Profibus, Modbus, OPC, custom serial/Ethernet interfaces, ...).

This approach has a number of disadvantages:

- It requires control system engineers with skillsets that are not commonly taught together (IEC61131-3 languages for the PLCs vs. high-level languages for the SCADA systems).

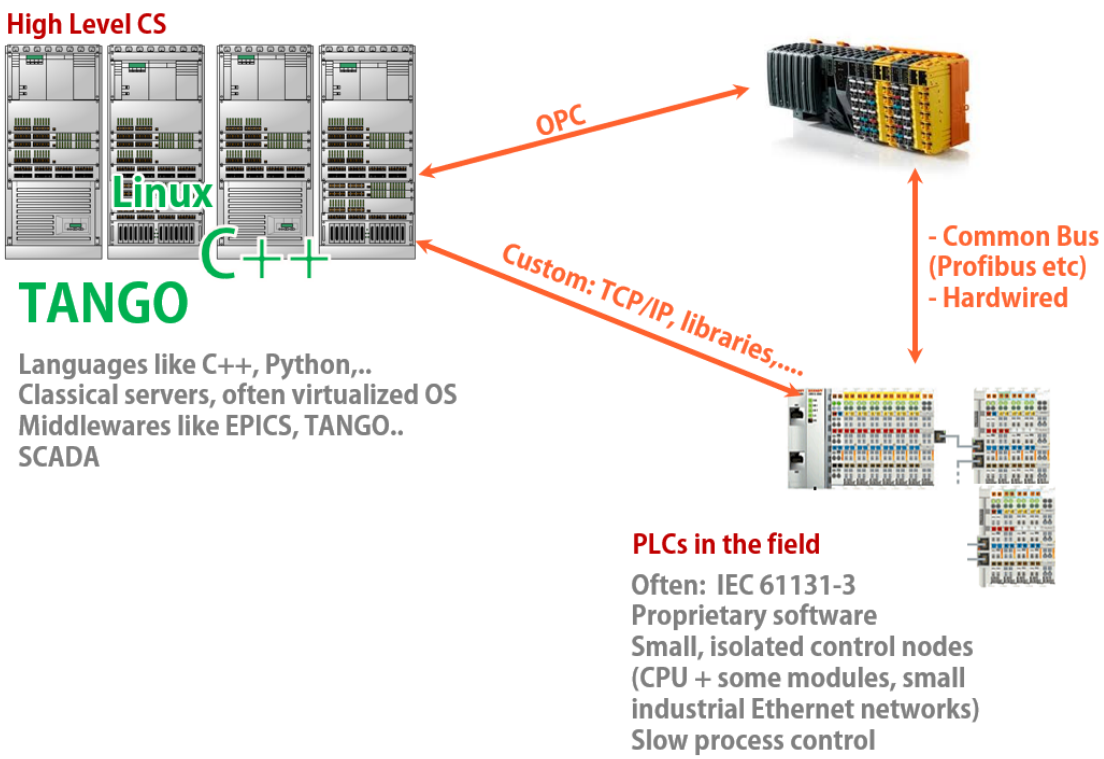


Figure 1: Commonly found PLC architectures.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA5

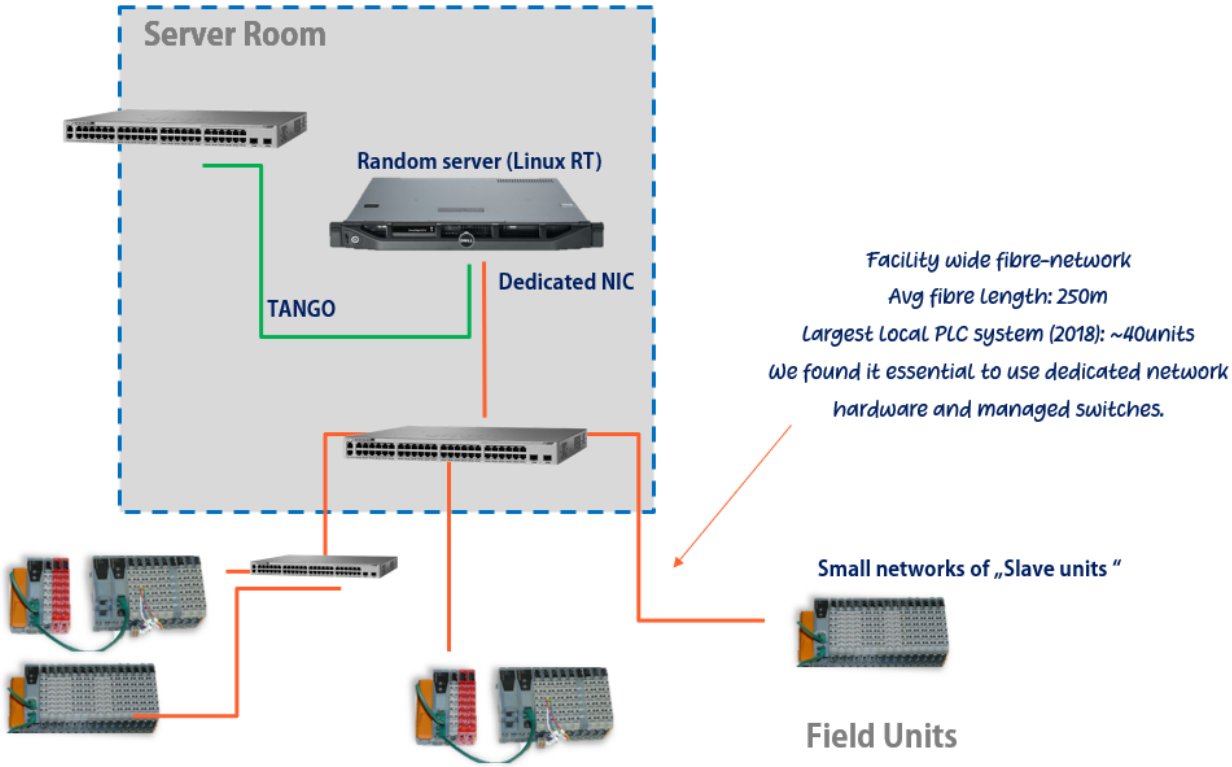


Figure 2: Alternative concept for propagating Industrial Ethernet systems.

- It requires the development and maintenance of software development infrastructure for both PLCs and SCADA systems.
- Interfaces are costly in terms of resources (require manpower, customized solutions).
- There is limited flexibility and the system is not truly distributed.

ALTERNATIVE CONCEPT

While PLC programming mostly remains within a closed, proprietary world, a few vendors provide low-level stacks to encourage growth of their ecosystems, most notably Beckhoff with EtherCAT [2] and B&R with PowerLink [3]. Both companies formed some form of consortium to make their standard freely available, and provide documentation and even software implementations for developers of new modules (slave units).

We tested and implemented prototypes for both technologies, and while we prefer some of Beckhoffs' configuration possibilities and the design of their IPCs, B&R has the more attractive selection of modules, the "reaction" technology and modules that can be directly connected to fibre-based networks.

The availability of those stacks including software implementations, however, also allows the replacement of the master units. This means, you can remove the CPUs which drive the modules using proprietary software, propagate the field busses via standard networks (copper, fibre) and drive them with custom software (see Fig. 2).

The hardware can be then exposed facility-wide via the middleware and allows truly distributed access.

In our experience, it is helpful to use dedicated network hardware and managed switches for this purpose. It is possible to operate smaller, local networks over VLANs using cheap switches, but for larger systems and longer distances, we could not maintain real-time communication. Note that fibre-length in the hundreds of meters can limit cycle time due to signal propagation times.

This architecture has multiple advantages:

- The PLC hardware is seamlessly integrated into the high-level controls architecture.
- CS Engineers can use high-level languages like C++ to drive the hardware.
- Interfaces to "classically programmed" PLCs (for example, from external contractors or for safety-critical systems) are trivial using communication modules like Profibus.
- The solution can significantly reduce cost-per-channel by removing expensive CPU units.
- Only the slave units are deployed in the field and exposed to EMP, radiation, … the driving hardware is safely located inside the server room. This reduces maintenance cost and increases availability.
- The approach is highly flexible; with our software, adding channels to the control system requires no programming, just configuration and a bus restart.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCA5

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

RESULTING SOFTWARE

To implement this system, we provide two APIs:

- **Fieldbus:** An abstraction of the stacks that drive the industrial Ethernet networks (Beckhoff and B&R), simplifying bus management to a few commands (start the bus, stop the bus, module discovery..).
- **CSIO:** This abstract class (with specific implementations like 'DIO', 'AIO', 'Stepper') is used to access the individual modules.

To use PLCs, a user has to go through the following steps:

1. Physical setup including network infrastructure
2. Configuration of the topology and cycles in Automation Studio/System Manager. This step can technically be also done by custom implemented software, but we currently don't find value in doing so.
3. Starting the bus, for example using a dedicated TANGO Server.
4. Discovering modules via the Fieldbus API. When calling this method, a collection of CSIO objects is returned. Our TANGO Server currently uses this to create dynamic attributes for access; though later on we wish to expose the module via individual servers.
5. Working with the modules. The methods of each CSIO implementation correspond to some register-read/write command. After each call, the bus has to be "synced", corresponding to the typical cycles in a PLCs. This is hidden by our TANGO server.

This set of APIs is also the base for a "native" LabVIEW integration [4], which allows seamless use of B&R and Beckhoff hardware within NIs' LabVIEW

OUTLOOK

The system shown is currently an operational protoype, running two beamlines and various small subsystems in ELI Beamlines. In the next months, we will

- Work on the upper abstraction layers and completely integrate the system into the local control infrastructure described in [5]. Specifically, we wish to hide the implementation specifics and use the same abstract API we use for any other motor/DIO/...: A user shouldn't know they are dealing with a fieldbus-based implementation. This also means exposing the modules as individual TANGO servers rather than as TANGO attributes.
- Quantify and optimize the performance and limitations of the drivers.

REFERENCES

[1] D. Maier-Manojlovic, "Real-Time EtherCAT Driver for EPICS and Embedded Linux at Paul Scherrer Institute (PSI)", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, pp. 153–156. doi:10.18429/JACoW-ICALEPCS2015-MOPGF027

[2] EtherCAT, https://www.ethercat.org/default.htm

[3] Powerlink, https://www.ethernet-powerlink.org

[4] Birgit Ploetzeneder, "Native LabVIEW pro průmyslovou automatize", *TA ČR GAMA Grant*, 01/17-12/2019.

[5] P. Bastl *et al.*, "Hardware Architecture of the ELI Beamlines Control and DAQ System", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 2000–2006. doi:10.18429/JACoW-ICALEPCS2017-FRAPL05

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB1

ACOP.NET: NOT JUST ANOTHER GUI BUILDER

P. Duval, M. Lomperski, J. Szczesny, H. Wu, DESY, Hamburg, Germany
T. Kosuge, KEK, Tsukuba, Japan
J. Bobnar, Cosylab, Ljubljana, Slovenia

Abstract

ACOP (Advanced Component Oriented Programming) tools have been useful in control system GUI application development for some time, originally as an ActiveX component [1] offering a transport layer and a multi-faceted chart and then later as a suite of components in the Java world [2]. We now present a set of ACOP components for development in .NET. And where the emphasis in the past has been primarily on rapid application development of rich clients, this new palette of components is designed both for fully featured rich-client development in any of the .NET supported languages (C#, C++, VB, F#) as well as for fully configurable clients (with design-time browsing), where no programming on the part of the developer is necessary, and of course for any combination between these extremes. This is an important point, which will become clear when we contrast application development with ACOP.NET with other control system GUI builders such as Control System Studio (CSS) [3] and Java DOOCS Data Display (jDDD) [4]. Although Visual Studio is the GUI builder of choice, we will present other available options, for example on Linux. Examples using transport plugs for TINE [5], STARS [6] and EPICS Channel Access [7] will be given.

INTRODUCTION

The control system for a particle accelerator or other large facility must meet stringent requirements for stable operation. And it must offer diagnostic tools for spotting, finding and fixing problems as well as online and offline analysis tools for examining machine data and improving operations. This much goes without saying. Yet, the control system is often judged by its operational interface, i.e. at the presentation level.

There are several strategies for providing the operators, engineers, and machine physicists with useful control system applications. These applications can be provided by a controls group. Or application development tools can be given to the end users so that they might generate the controls applications themselves. Or … both. And note that the three groups of end users mentioned above will each have a different perspective as to what a controls application should be able to do. They are also likely to have different skill sets concerning programming abilities and understanding the various aspects of the machine being controlled.

A very common strategy is to have a controls group provide certain core applications, but to allow the operators and engineers to create control panels using some framework, where no programming skills are required, and where, more often than not, programming is not even possible. This is for instance the case of CSS, Taurus [8], and jDDD, each of which provides the panel developer with a rich set of displayer widgets, which can be attached to a control system address, and combined with other graphical widgets providing some display logic. Here it is assumed that there is no need to program any additional display logic, although this would be nominally possible with Taurus in a Qt environment.

Machine physicists, on the other hand, frequently need higher level control in order to 'test things' and improve the overall performance of the machine. That is, they need to be able to program at the client side. A common strategy here is to provide control system components in a mathematically based programming environment such as MatLab or Python (and NumPy).

This two-pronged approach, a panel builder for basic display and control applications, and MatLab or Python support for high level controls, is common to many accelerator facilities.

Yet another strategy is to support rich client development using rapid application development (RAD) tools. This approach has met with great success at HERA and PETRA-3 where RAD tools in either Visual Basic and ACOP ActiveX or Java and ACOP beans were used by both the controls group and machine physicists to develop applications [9].

We shall now describe ACOP.NET which provides a single application development paradigm, offering both a non-programming panel building environment as well as a fully-programmable environment (and of course any combination between these two extremes). In fact, a panel application without a single line of user code can be extended by supplying additional logic at any later date.

ACOP AND .NET

ACOP was originally designed as a RAD tool in the ActiveX world [1] and was predominately used in Visual Basic applications. It featured a very powerful chart, with a number of control system oriented features and a transport layer. It was later ported to java and expanded to include a variety of displayer beans suitable for rich client development in java [2].

Now, in general, rich client development in java requires more extensive programming skills than programming in Visual Basic. Thus, it is very tempting to offer a panel building framework with smart widgets which can be configured at design time and remove the programing aspect entirely from the application developer. This is in fact the approach of both CSS and jDDD, where an application might exist as an XML file

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB1

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

which is interpreted and rendered at runtime. Taurus likewise makes use of an XML configuration file which is rendered at runtime, but here the platform is Python rather than java.

The latest variant of ACOP, which we now present, also offers a panel building paradigm with smart widgets configured at design time. The developer need not write a single line of code, but does need to choose a .NET programing language. The finished ACOP application will exist as a single executable file. The development project consists of a .NET *solution* and code modules (maintained by the Visual Studio designer) instead of XML files. The executable is every bit as portable as a java application and can be run under mono [10] on non-windows systems.

As noted earlier, not all control system application needs can be met by pure configuration (no matter how many '*CALC*' records or properties are invented on the server side). A team responsible for high level controls will typically require an interface to Python or MatLab in order to be able to 1) program and 2) to make use of the available mathematical packages.

And here the beauty of ACOP.NET comes to shine. Any ACOP.NET application can instantly become a rich client (and easily leverage third party mathematics and other packages). The developer will be able to choose among such languages as Visual Basic (VB), C#, C++/CLI, and F#, tailoring an application to his own programing preferences. Thus, some thought should go into the question as to which language to choose for a particular controls project, even if it might be destined to remain a simple configured panel. And, as an aside to Python aficionados, note that Python and .NET can be combined via Python for .NET [11].

.NET, Visual Studio, and Mono

Microsoft .NET enjoys a wide acceptance and client base in the industrial and business world, if not in the control system community. Applications written in .NET on a Windows platform can generally be run on a Linux or Mac platform as is via *mono* (which is now technically sponsored by Microsoft). Although ACOP.NET GUI applications can in principle be written using mono developer on a Linux platform, it is strongly recommended to make use of Visual Studio, which features an integrated designer capable of writing and isolating the GUI component relevant code. As the *community* version of Visual Studio can be downloaded and installed free of charge this should not present any cost or licensing burdens on the developer.

The current version of ACOP.NET makes use of WinForms, which are mature, wide-spread, and encompass an extensive variety of common components.

Real vs. Graphical Programming

As is the case with any panel builder the developer can configure a useful, working application by simply setting design time properties in the GUI builder (here Visual Studio). ACOP.NET offers design-time browsing of the control system, so that the developer is not required to know a priori the end-point addresses he wishes to attach to the ACOP.NET displayers.

Often, an application can only go *so far* by merely attaching addresses to displayers before some simple display logic is desired. This can be something as mundane as changing display characteristics based on a condition, or applying some trivial manipulation of the associated data before it is displayed.

If there is no ability to actually code something, then the GUI builder might solve logic problems by offering additional logical widgets (e.g. jDDD), which is tantamount to offering a form of graphical programming. Problems of data manipulation (because the data acquired from a control system address is not in the form needed for display) are then passed on to the front end, where new properties (or CALC records) are invented to make the displayer's life easier (although not necessarily the front-end developer's).

None of this is necessary with ACOP.NET, as simple logical decisions can be coded simply in the developer's favorite language, just as minor data manipulation prior to display can be coded easily. And when more extensive experimentation with front end data is required there is effectively no alternative to programming.

The ACOP.NET control widgets can be used as is in any application and these include shape widgets which might be useful in synoptic displays. These can be used side by side with the standard .NET or other 3rd-party components.

Runtime Features minus Popup Pollution

The ACOP controls do not themselves launch independent, non-modal windows. Thus there is no danger of inadvertently filling the desktop with disjointed display windows (Popup Pollution). Launching a separate daughter window, if that is in fact desired, requires a tiny bit of code, where the click event of the launching component is used to show the additional ACOP Form.

On the other hand many controls such as the chart do offer context menus whereby various properties can be edited, or settings applied at runtime.

Should an ACOP control be bound to a control system address which is not responsive, an initial modal window is displayed alerting the user to this fact. Likewise, some components will alert the user to a failure via a modal window, for instance, when an access lock cannot be obtained, or a setting cannot be applied.

Data acquisition errors are otherwise displayed, where possible, within the ACOP displayer component in the given *error color* and *blinking* state. In addition for many components a specific *error value* can be supplied, which will supersede the last displayed value upon any data acquisition error.

Some ACOP components are in fact designed to interact with other ACOP components. This is particularly true of the ACOP Chooser, which allows a user to make an end-point address change (such as

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB1

changing the targeted device or context) at run-time and apply the change to those ACOP components designated at design time. The ACOP Status Bar can likewise set the enabled state of a list of designated components from *true* (expert mode) to *false* (read-only mode). Finally those ACOP components which can apply settings can also set selected properties of specific components in lieu of control system end points. For example an ACOP Wheel Switch could be used to set the Y-Axis Maximum of a chart.

If no explicit tool tip text is provided for an ACOP Component, it will offer the targeted end point address. This is not true of the ACOP Chart however (except for the chart frame). The application designer can instead specify an auto-link-update mouse move tool tip where the displayed data vs. horizontal coordinates or simply the chart coordinates can be displayed.

ACOP Components

As noted earlier the preferred framework for building ACOP applications is Microsoft Visual Studio. When making use of one of the ACOP project templates, the ACOP components should already be included in the toolbox, as shown below in Fig. 1. New Acop Forms can be added via the project's *Add New Item* context menu.

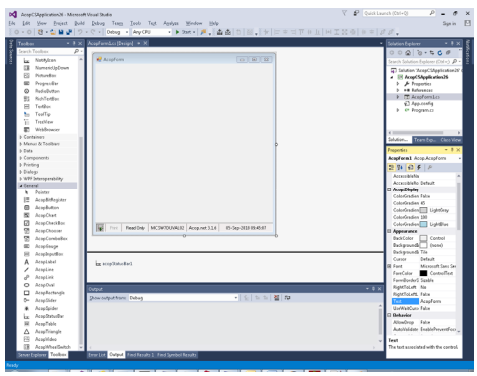


Figure 1: Starting a new project: an empty ACOP application.

As can be seen in the figure above, the basic ACOP components consist of 20 GUI components plus a non-GUI ACOP Link component which can be used in situations where explicit programming is necessary. Of these 20 components, there are two specialty components, the ACOP Spider and the ACOP Status Bar. The ACOP Spider is useful as a debugging portal, where specific link information is made available. And the ACOP Spider is itself contained within the ACOP Status Bar, which serves as a menu footer when placed on an ACOP Form (and is included by default on the primary ACOP Form when making use of an application template).

In addition, there are several shape components, which can be used in synoptic display. See Fig. 2 below. Synoptic displays often make use of clusters of components which are repeated on a panel's façade. The application designer can make his life easier by either selecting the entire cluster and copy-and-paste or by including the cluster in his own .NET component (Project Solution category = *Class Library*), which can then be used and re-used in any ACOP application.

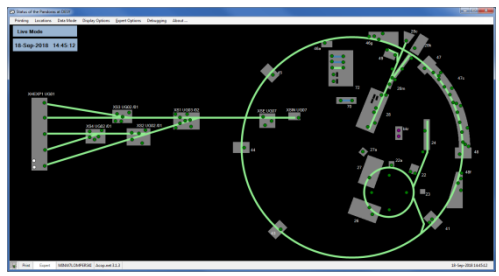


Figure 2. A synoptic display example.

Control System Plugins

ACOP components offer a control-system independent transport interface to obtain and display data from the attached control system end points. A supported control system protocol needs to provide a .NET plugin library which implements the ACOP Transport interface. Over the years there have been many, many transport layers which purport to provide a common interface to various known control systems, and we will not name them here. In addition to this requisite feature, an ACOP transport plugin should also provide browsing instructions and transport information and statistics, useful in debugging an application.

Although some ACOP components might make use of the browsing interface at runtime (such as the ACOP Chooser), this feature is mostly called upon at design time, obviating the need for the developer to be aware of and key in control system end-point addresses.

The transport information interface is used to provide a snapshot of the application's connectivity via the ACOP Spider, which is embedded in the ACOP Status Bar but can also be placed independently on any ACOP form. A green spider indicates no connectivity issues and a mouse click on the spider will launch one of the few non-modal windows automatically available to an ACOP application, where the application's connection tables can be examined in more detail. An example of connection information window is shown in Fig. 3 below.

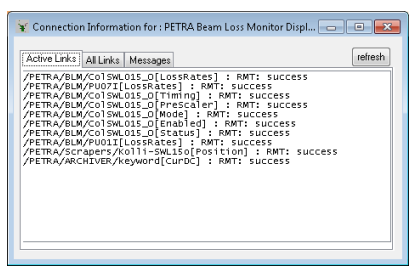


Figure 3: An example of the connection information made available by the ACOP Spider.

The end point address paradigm used in the ACOP transport API is one of a 4-tiered hierarchy, following that used by the TINE control system as well as DOOCS [12] and TANGO [13]. The top level is the Link Context,

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB1

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

followed by the Link Server, the Link Device (which together form the link end-point address), and the Link Property (which specifies the method or attribute to be accessed at the end point). As this does not directly fit some control system protocols, an ACOP plugin can turn off browsing and make use a simple Link Address or provide a mapping into the ACOP hierarchy. For example, Channel Access [7] does not provide any systematic hierarchy other than a flat namespace appended by a known set of meta-property decorations. STARS has a more-or-less open hierarchy. In such cases one can simply supply a Link Address without browsing or make use of extra naming services specific to the control system site (e.g. where control system addresses are maintained by LDAP or other file system services), in which case the ACOP plugin used for a particular control system protocol would be site specific.

To this end, ACOP.NET is somewhat TINE centric, as it makes allowances for features available in TINE (such as multi-channel arrays, property-oriented server browsing, structured data, etc.) which are not available in other control systems. This is hardly a short-coming and is in analogy with CSS being EPICS centric, jDDD being DOOCS centric, and TAURUS being TANGO centric.

In any event, an ACOP transport plugin must be written as a .NET DLL which utilizes the *AcopPlugin* interface. This is shown in Fig. 4.

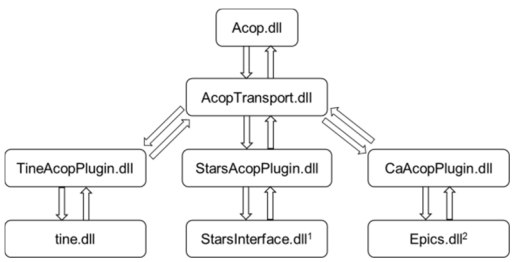


Figure 4: Acop Transport plugins.

An application can make use of multiple plugins if desired. Plugin information, such as the default protocol for an ACOP widget or design-time browsing instructions, can be supplied in the *AcopTransport.dll.config* file, deployed during setup and installation.

APPLICATION BUILDING

As mentioned earlier, the most sensible method of building an ACOP application is by making use of some flavor of Microsoft Visual Studio. In addition it is recommended to make use of one of the ACOP application templates for C#, VB, or VC++ development. The choice of language might seem irrelevant if one is not intending to program. If however, the application is likely to be long-lived, a modicum of thought concerning the programming skills of those who might want to add a finishing-touch at some later stage is nonetheless encouraged.

Installing ACOP.NET

ACOP.NET is most easily installed on a Windows machine by making use of the *Setup* utility found on the ACOP.NET Web page [14]. A pre-requisite is an existing installation of Visual Studio. Depending on the ACOP plugins selected, the corresponding control system should also be installed on the local host.

Following the installation, a new Visual Studio project will offer an AcopCSApplication in the C# Templates category and an AcopVBApplication in Visual Basic Templates. Selecting one of these will produce an empty application panel resembling that shown above in Fig. 1.

Configuring Simple Panels

Starting with the blank slate shown in Fig. 1 above, one then places the GUI components one wants on the empty panel, as with any modern GUI builder. The ACOP components have their own properties and events, some of which are specific to display and GUI interactions, as well as others pertinent to data acquisition. The latter fall into the category *Acop.Transport*, which is shown in the designer property grid in Fig. 5 below.

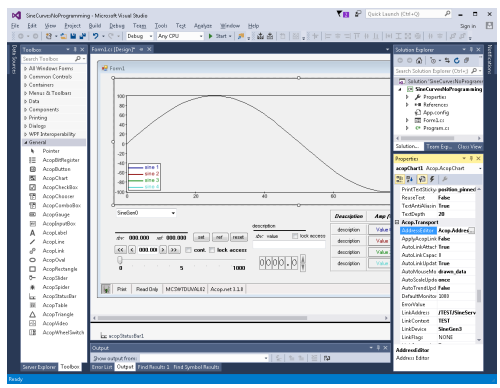


Figure 5: An ACOP application in design mode, highlighting the Acop. Transport category.

It is here that design time browsing can greatly assist in finding a control system end point address and its corresponding property display characteristics, although these values can also be keyed in by hand in the designer.

Further help for the application programmer is provided through various modal designer editors which allow multiple settings of ACOP properties at the same time. For instance the Address Editor, shown below in Fig. 6, is available to all ACOP GUI components and allows browsing and selection of multiple link properties in a logical and intuitive manner, in place of setting these properties one by one in the property grid.

Otherwise each ACOP component will offer specific display properties in the designer property grid.

If synoptic displays are desired, the ACOP shape components can be utilized for basic shape design. These are likewise smart components, which can display control system data as well as adjust display settings such as fore and back colors and the blinking state based on

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB1

comparing incoming data to given thresholds. More complicated shapes can easily be built from the basic shapes.

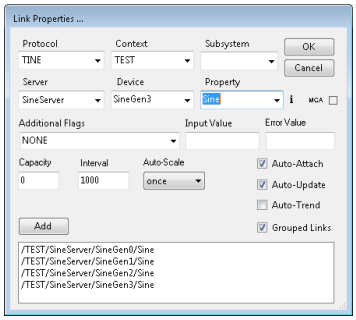


Figure 6: The design-time ACOP Address Editor.

Adding Code

One can generate a powerful and fully operative control system application via the simple panel configuration method described above. And it is easy enough to take the application to the next level by adding a smattering of code, which just might help render the application a bit more intuitive and functional to the end user.

An event delegate can be supplied for any of the common GUI component events (such as mouse click, or mouse move, etc.) as well as ACOP transport events (such as Link Update or Link Error). There are likewise numerous ACOP API calls to obtain the control system data so that the application might make use of it.

For instance if the value at the cursor position over an ACOP Chart display is needed in an ACOP Table cell, this could be achieved as follows: 1) Get the drawn data by supplying a link update delegate; 2) Catch the mouse move over the chart with a mouse move event delegate, which also provides the drawn data array index; 3) Add this element to in the desired ACOP Table cell.

Runtime Options

Various ACOP GUI properties and actions can be also be accessed at runtime, via associated context menus. And as noted earlier, the ACOP Status Bar provides the ACOP Spider and its runtime debugging capabilities. It also provides runtime printing options (including print to logbook) as well as application mode settings (read-only vs. expert).

CONCLUSIONS

There are of course numerous packages which can be used as control system GUI builders. CSS and jDDD are java-based panel builders, which do not offer the ability to program (although they are both able to launch independent applications or scripts). Taurus is python based and allows simple panel building (with some hidden python programming) as well as additional programming (in python) at an expert level through the Qt designer. In addition, PyQt, MatLab and LabView all offer GUI building with extensive programing capabilities (as long as there is an interface to the control system), but do not offer simple designer panels.

ACOP.NET on the other hand provides a GUI building package which can cover the entire spectrum from simple designer panel to complex application with as much computational programing as is desired and offers design-time browsing of the control system. When one does need to or wish to add code, this is again aided by the Visual Studio designer providing event delegates and intelli-sense information, and can be done in one of several languages suited to the developer's needs. The available languages range from Visual Basic (for those whose programing skills are not particularly pronounced) to C# (for the java aficionados) to F# (for those who prefer functional programing), along with C++/CLI for C++ purists. Additional packages such as Math.NET and Python.NET can provide the user with a rich development environment indeed.

ACKNOWLEDGEMENTS

We would like to thank Graham Cox, STFC Daresbury Laboratory, UK, for his assistance in providing the EPICS Channel Access .NET library [15] for discussions concerning its integration into ACOP.NET.

REFERENCES

[1] I. Deloose, P. Duval, H. Wu, "The Use of ACOP Tools in Writing Control System Software", *ICALEPS'97,* 1997.

[2] J. Bobnar, *et al.*, "The ACOP family of beans: the framework independent approach", *ICALEPCS'07,* 2007.

[3] CSS website, http://www.csstudio.org

[4] jDDD website, http://jddd.desy.de

[5] TINE website, http://tine.desy.de

[6] STARS website, http://stars.kek.jp

[7] EPICS website, http://www.aps.anl.gov/epics

[8] Taurus website, https://taurus-scada.org

[9] P. Duval and H. Wu, "Using ACOP in HERA Control Applications", *PCaPAC'00,* 2000.

[10] MONO website, https://www.mono-project.com

[11] Python for .NET website, http://pythonnet.github.io

[12] DOOCS website, http://doocs.desy.de

[13] TANGO website, http://www.tango-controls.org

[14] ACOP.NET Website, http://acop.desy.de

[15] G. Cox, "A .NET Interface for Channel Access", *PCaPAC'08*, 2008.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB2

DEVELOPMENT OF ACOP .NET STARS TRANSPORT LAYER

T. Kosuge†, H. Nitani, Y. Nagatani, H. Ishii
High Energy Accelerator Research Organization, 3050801 Tsukuba, Japan
J. Szczesny, P. Duval, Deutsches Elektronen-Synchrotron, 22607 Hamburg, Germany

Abstract

Simple transmission and retrieval system (STARS) is an extremely simple and flexible software for small scale control systems with TCP/IP sockets. It is used by systems such as the beamline control system, and other systems at the KEK Photon Factory (KEK-PF). STARS works on various operating systems; therefore, STARS client developers can choose their preferred programming language. .NET is commonly used to develop graphical user interface (GUI) applications for beamline control at the KEK-PF.

Advanced component oriented programming (ACOP), which was developed by DESY, is very useful for GUI development, and a .NET version of ACOP was recently developed. ACOP communicates with various systems through a transport layer. We are currently developing ACOP .NET STARS transport layer. So far, we have succeeded in adding very primitive functionality.

OVERVIEW OF STARS

Simple transmission and retrieval system (STARS) is a software for small-scale control systems [1,2]. STARS consists of a STARS server and STARS clients. Each client is connected to the STARS server via a TCP/IP socket and handles text-based messages. The current version of the STARS server is written in Perl; therefore, STARS users can choose their preferred operating system and programming language for STARS client development.

Node Name and Hierarchical Structure

Every STARS client has its own unique node name that is used to identify the destination of the STARS text- Figure

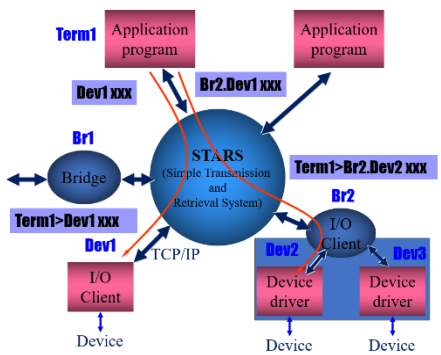


Figure 1: Message transportation on STARS.*

based message. Figure 1 shows examples of STARS message transportation. If a STARS client that has "Term1" as a node name sends a text-based message of the form "Dev1 xxx", the message will be delivered to the client "Dev1". Hierarchical structure node names that are separated with a period "." are available, and the server uses the first part of the node name as the destination.

Command, Reply and Event

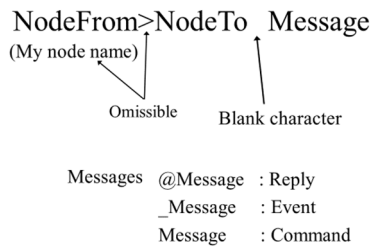


Figure 2: STARS message structure.*

Figure 2 shows the structure of a STARS message. The first part before ">" shows the origin of the message, and the next part before the first whitespace shows the destination of the message. The string which follows the first whitespace character can be either a command, a reply, or an event. A reply string starts with "@", an event string starts with "_", and a command string does not contain a preceding symbol. The strings can also contain values.

Event Delivery Function

STARS has an event delivery function. An event delivery request is registered by sending a "flgon" command to the server, which also has a node name "System". Figure 3 shows an example the event delivery function. "Tm1" and "Tm2" request an event from "Dev1" by sending "System flgon Dev1" to the server. After the registration, if "Dev1" sends an event to the server, the message is delivered to "Tm1" and "Tm2".

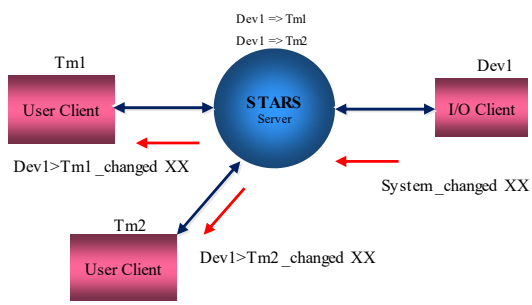


Figure 3: Event delivery function of STARS.*

† takashi.kosuge@kek.jp
* Source: http://stars.kek.jp/

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB2

Certification Procedure

STARS has a simple certification procedure.

- Host name certification
- Node name and keyword certification
- Node name and host name certification (optional)

The server checks the client's host name first. If the client passes host name certification, the server then sends a random number to the client. The client must then send its node name and keyword corresponding to the number. After keyword validation, the server checks node name and host name, if necessary.

STARS Interface Library

STARS is a very simple system, and the development of a STARS client is easy if the programmer can handle TCP/IP socket and text. However, there may be some difficulty for beginners. STARS provides interface libraries and tools that help in the development of a STARS client as follows:

- Perl interface library
- Python interface library
- C interface library
- Java interface library
- .NET interface library
- Perl STARS client development wizard
- Windows form C# template for Visual Studio
- Windows form VB .NET template for Visual Studio

Using the interface library, programmers can develop a STARS client while avoiding TCP/IP complexity.

STARS CLIENTS

STARS users can add new functions to their control system by developing STARS clients. STARS clients can be roughly classified into two types as IO clients and User clients. An IO client behaves like a device driver and controls hardware. User interface clients, including graphical user interface (GUI), are called User clients. A general purpose system can be developed if common commands are defined and a standard control system with STARS for KEK-PF beamline is introduced. Figure 4 shows a common beamline control system at the KEK-PF. The system provides the same command interface (getting values or setting values etc. of the hard ware). The beamline users can choose their preferred software such as LabVIEW, SPEC, common GUI program provided by the KEK-PF, or their own developed program.

Debugger

A client with the node name "Debugger" has a special purpose. When connected to the STARS server, the client can receive a copy of messages sent to all clients from the STARS server. "Debugger" works effectively during troubleshooting.

ACOP .NET

Advanced component oriented programming (ACOP) helps in the development of GUI programs [3]. The .NET version of ACOP was newly developed by DESY and is still under development.

ACOP tools appear in the Microsoft Visual Studio toolbox if ACOP .NET is installed (Figure 5). Powerful GUI tools (e.g., AcopChart and AcopLabel) can be used to choose and deploy the designer screen. ACOP supports TINE [4, 5] protocol, and programmers can develop the GUI without having to write the source code.

ACOP .NET STARS TRANSPORT LAYER

We started the development of ACOP .NET STARS transport layer to support STARS protocol. The primitive part of the transport layer was developed. ACOP .NET transport layer is written in C# and we added code for STARS.

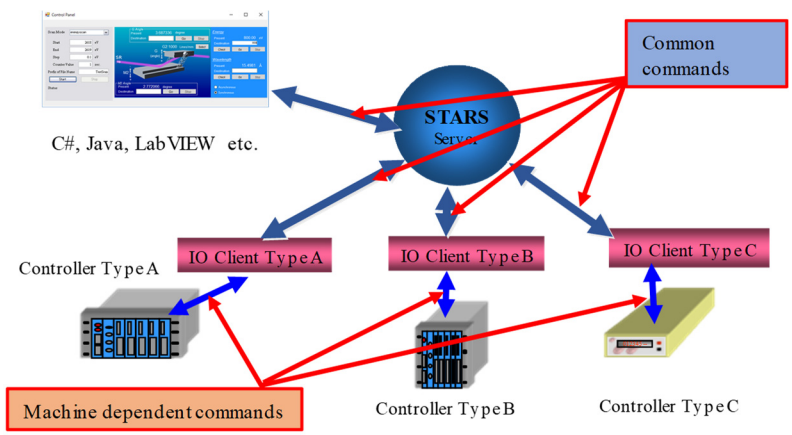


Figure 4: Standard control system for KEK-PF beamline that provides common interface commands.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB2

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

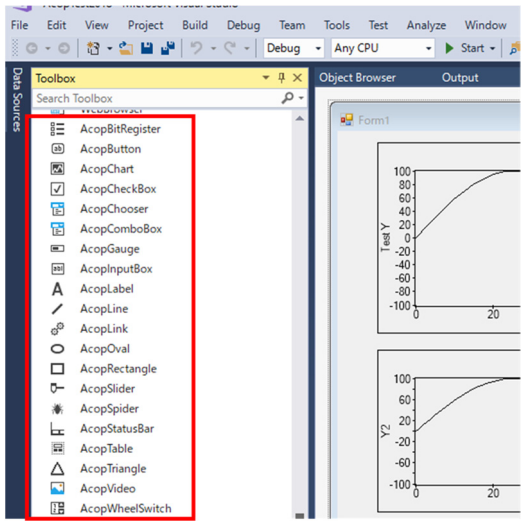


Figure 5: ACOP tools in Toolbox of Microsoft Visual Studio.

STARS .NET Interface Library

ACOP .NET STARS transport layer communicates with the STARS server through STARS .NET interface library (Fig. 6), which processes the connection and the keyword certification procedure during connection.

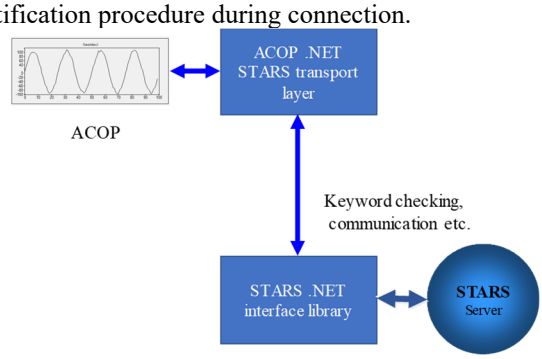


Figure 6: ACOP .NET STARS transport layer and STARS .NET interface library.

Mapping of Name Space

Mapping of name space between ACOP and STARS is one of the primary issues. The procedure to allocate name space for device names, properties, and commands was discussed, and the first version of the mapping table was generated (Table 1).

Table 1: Name Space Mapping Table

ACOP Properties	STARS Name Space
Context	Node name of GUI
Subsystem	Not allocated
Server	Host name : Port number
Device	Node name of IO Client
Property	[Command /] Reply / Event

Context of ACOP is mapped to its own GUI node name of STARS. Server is mapped to host name and port number of STARS server, which are separated with the ":" character. Device is mapped to node name of STARS IO client that is being monitored by this GUI client. Property is mapped to the part of STARS command, reply, or event. Multiple messages can be defined using the "/" character as a separator. This first version of mapping will be improved after practical examination.

Attach and STARS Event

If the attach method in ACOP .NET STARS transport layer is called, the event delivery function of STARS is enabled by sending the "flgon" command to the STARS server, and the information of the request such as host name, node name, etc. is registered in the address list. The attach method does not send the "flgon" command again if the same information is in the list.

EXAMPLE OF PROGRAMMING

STARS Windows form template for Visual Studio (C# or Visual Basic) helps in the development of the STARS GUI Client. Once the new Visual Studio project is created with the template, primitive source code that handles the method of the STARS .NET interface library is created automatically. However, this means that the programmer must carefully examine the source code and handle functions for communication with the STARS server. ACOP .NET with STARS transport layer allows for source code with less programing on the part of the programmers. Figure 7 shows an example of ACOP chart properties in the property window of Visual Studio. The host name of STARS server, node name etc. can be set with ACOP by editing these transport properties. The programmer requests the procedure that chooses an ACOP component from the tool box, deploys the component on the form, and sets parameters.

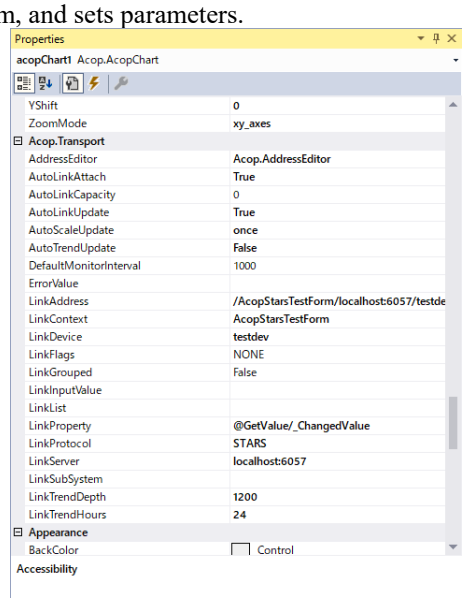


Figure 7: Property window of ACOP chart.

The ACOP address editor is a useful tool for setting ACOP transport properties (Figure 8). The editor window appears by clicking the button in the address editor field. "LinkAddress" property is generated automatically by entering "Context," "Server," "Device," and "Property" in

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB2

the text fields of the address editor. If the "Auto-Attach" check box is marked, the "AutoLinkAttach" property becomes "True." The component is attached automatically when the program is started, and parameters are set if the "OK" button is clicked.

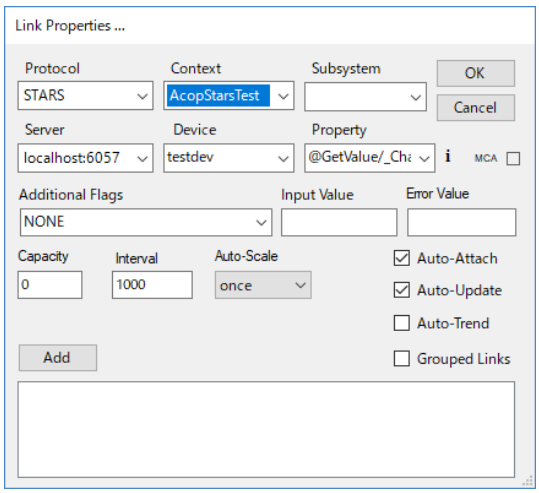


Figure 8: ACOP address editor.

Test with IO Client Simulator

We prepared an IO client simulator that returned values of the sine curve, added small noise values, and created a GUI client having an ACOP chart, and an ACOP button. Figure 9 shows a screenshot of the telnet terminal connected to the STARS server with the node name "Debugger." The "GetValue" command from the GUI named "AcopStarsTest" to IO client simulator named "testdev," and the reply message from "testdev" is observed on the screen.

STARS handles text-based messages only, and the IO client simulator returns values in text. Type-conversion (strings to integer, float etc.) is achieved by ACOP .NET STARS transport layer automatically. Figure 10 is a screenshot of a GUI client. We verified that the ACOP chart shows graph of values from the IO client simulator correctly.

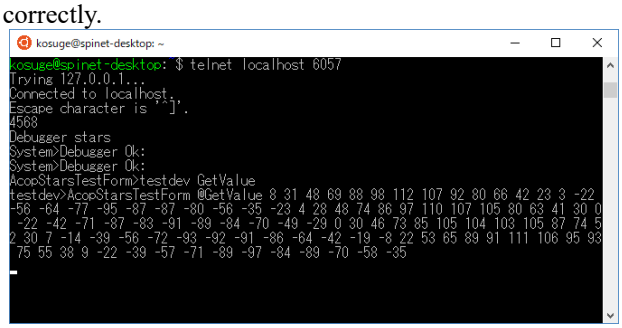


Figure 9: Screenshot of debugger.

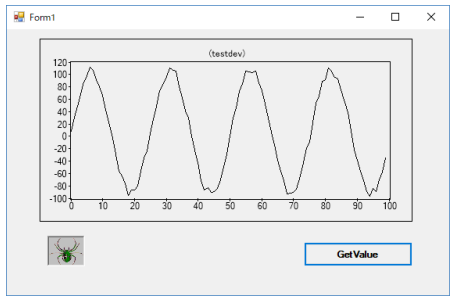


Figure 10: Screen shot of STARS GUI client with ACOP.

ACOP Spider

ACOP has a debugging tool called ACOP spider. The debugging function is enabled by deploying "AcopSpider" on the form. If the "AcopSpider" icon is clicked when the program is running, the debugging window appears on the screen and shows the connection information. The debugging window has "Active Links," "All Links," and "Messages" tabs. Figure 11 shows the debugging window with the "Messages" tab active. We verified that the ACOP spider works satisfactorily on debugging as well.

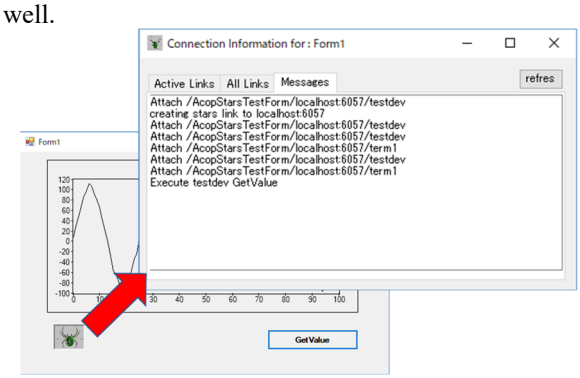


Figure 11: Debugging with ACOP spider.

CONCLUSION

We have succeeded in developing a first version of ACOP .NET STARS transport layer and obtained sufficient results upon making a GUI STARS client with ACOP. The name space mapping between ACOP and STARS still needs to be refined. Checking all the ACOP components and replacing the name space mapping table (if necessary) will be the next steps in the development.

REFERENCES

[1] STARS, http://stars.kek.jp

[2] T. Kosuge and Y. Nagatani, "STARS: Current Development Status," in *Proc. PCaPAC'14*, Karlsruhe, Germany, Oct. 2014, paper WPO019, p. 75.

[3] I. Krznar *et al.*, "New ACOP Beans and TINE General Purpose Diagnostic Applications," in *Proc. ICALEPCS'09*, Kobe, Japan, Oct. 2009, paper TUP034, p 161.

[4] TINE, http://tine.desy.de

[5] P. Duval and S. Herb, "The TINE Control System Protocol: How to Achieve High Scalability and Performance," in *Proc. PCaPAC'10*, Saskatoon, Canada, Oct. 2010, paper WECOAA02, p. 19.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB3

IMPROVING Web2cHMI GESTURE RECOGNITION USING MACHINE LEARNING

R. Bacher, Deutsches Elektronen-Synchrotron DESY, 22607 Hamburg, Germany

Abstract

Web2cHMI is multi-modal human-machine interface which seamlessly incorporates actions based on various interface modalities in a single API, including finger, hand and head gestures as well as spoken commands. The set of native gestures provided by off-the-shelf 2D- or 3D-interface devices such as the Myo gesture control armband can be enriched or extended by additional custom gestures. This paper discusses a particular method and its implementation in recognizing different finger, hand and head movements using supervised machine learning algorithms including a non-linear regression for feature extraction of the movement and a k-nearest neighbour method for movement classification using memorized training data. The method is capable of distinguishing between fast and slow, short and long, up and down, or right and left linear as well as clockwise and counter clockwise circular movements, which can then be associated with specific user interactions.

INTRODUCTION

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.

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. Therefore, any new interaction capability such as gesture or spoken command recognition will only be accepted by the users if they provide comparable or even better handiness, ease-of-use and reliability to click-like interactions.

This paper discusses a particular method and its implementation aiming at improving the quality and reliability in recognizing gestures based on different finger, hand and head movements using supervised machine learning algorithms. This includes a non-linear regression for extracting the features of the movement and a k-nearest neighbour method for movement classification using memorized training data.

Web2cTOOLKIT

The Web2cToolkit [1] is a collection of Web services, i.e. servlet applications and the corresponding Web browser applications, including

- Web2cViewer: Interactive synoptic live display to visualize and control accelerator or beam line equipment,
- Web2cViewerWizard: Full-featured graphical editor to generate and configure synoptic live displays,
- Web2cArchiveWizard: Web form to request data from a control system archive storage and to display the retrieved data as a chart or table,
- Web2cArchiveWizardWizard: Full-featured graphical editor to generate and configure archive viewer displays,
- Web2cToGo: Interactive display especially designed for mobile devices embedding instances of all kinds of Web2cToolkit Web client applications,
- Web2cGateway: Application programmer interface (HTTP-gateway) to all implemented control system interfaces,
- Web2cManager: Administrators interface to configure and manage the Web2cToolkit.

The Web2cToolkit provides interfaces to major accelerator and beam line control systems including TINE [2], DOOCS [3], EPICS [4], TANGO [5] and STARS [6].

Web2cHMI

The Web2cHMI [1] is a platform-neutral Web-based human-machine-interface implementation for accelerator operation and maintenance applications in the context of the Web2cToolkit Web service collection. It supports various modalities which can be used simultaneously including

- 1D/2D flat gestures including single-finger actions (mouse) and single- or multi-finger gestures (touch-sensitive display)
- 2D/3D spatial gestures including hand-gestures (LEAP Motion controller [7]), hand- or arm-gestures (Myo gesture control armband [8]) and 3-axis (yaw, pitch roll) head movements (smart glasses including Epson Moverio BT-200 [9] and Vuzix M100 [10])
- English spoken commands (Sphinx speech recognition [11]).

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB3

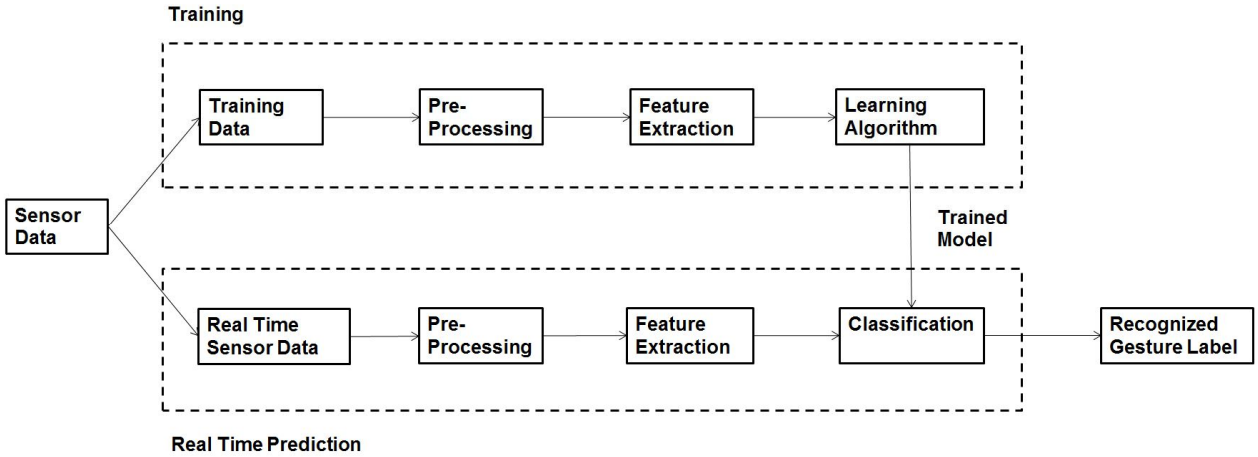


Figure 1: Gesture recognition workflow using supervised machine learning (classification).

Web2cHMI recognizes various primitive, i.e. native or input device-specific gestures including

- Mouse: *Click, Move*
- Touch-sensitive display: *Tap, Move / Swipe, Pinch* (two fingers)
- LEAP Motion controller: *Key-Tap, Swipe, Open-Hand, Closed-Hand, Circle*
- Myo gesture control armband: *Double-Tap, Wave-Out / Wave-In, Fingers-Spread, Fist*
- Smart glass: *Move-Fast / Move-Slow, Roll*

In addition, enriched gestures formed by primitive gestures followed by moves or rotations etc. are supported.

The Sphinx speech recognition system knows a Web2cHMI-specific vocabulary listed in a dictionary such as "*Browse Up*" or "*Lot More*".

GESTURE RECOGNITION WORKFLOW USING MACHINE LEARNING

Web2cHMI gesture recognition using supervised machine learning consists of two distinct phases [Fig. 1], training phase and real time prediction phase. The input for both phases is continuously recorded position data of the user's input device in both Cartesian (X, Y) and polar (R, PHI) coordinates resulting from finger, hand, arm or head movements. The continuous stream of sensor data is pre-processed to reduce the noise floor of the measurement or to remove obvious outliers. Next, the cleansed data are fed into a mathematical algorithm to extract a set of parameters (features) representative for a particular gesture.

During the training phase the extracted feature values are used by a supervised learning algorithm to train a model resulting in sets or clusters of learned data representing valid gesture types. The Web2cHMI learning algorithms provides four distinct learned data sets representing *Long-Slow*, *Long-Fast*, *Short-Slow* or *Short-Fast* movements. During real time prediction phase the extracted feature values are classified by comparing with the memorized learned data sets. The result of this classification analysis is the proper gesture label.

It is important that both training and real time prediction phase are using the same pre-processing and feature extraction algorithms.

The Web2cHMI gesture recognition algorithm is entirely coded in JavaScript capable of being processed locally by a standard Web-browser and does not rely on any server-side logic.

Feature Extraction

The feature extraction algorithm implements a non-linear regression (Nelder-Mead method) to fit consecutive 333 ms long time sequences of sensor data to a predefined mathematical model [Eq. 1]:

$$\begin{aligned} &\mathbf{t} \le \mathbf{t_{start}}: && \mathbf{f(t) = s_{start}} \\ &\mathbf{t_{start} < t < t_{end}}: && \mathbf{f(t) = s_{start} + \left(\left(\frac{s_{end} - s_{start}}{t_{end} - t_{start}} \right) * (t - t_{start}) \right)} \quad (1) \\ &\mathbf{t \ge t_{end}}: && \mathbf{f(t) = s_{end}} \end{aligned}$$

Here, the parameters t_{start} and t_{end} mark the start and end time of the recognized gesture within the analysed user's finger, arm or head movement, respectively. Similar, s_{start} and s_{end} are indicating the start and end position of the gesture.

The regression reduces the dimension of the gesture recognition task by two orders of magnitude and is performed for each Cartesian coordinate orientation (linear horizontal movement, linear vertical movement) separately. Fitting the polar angle PHI allows the identification of circular movements. If the regression algorithm converges and the parameters are confined within reasonable limits, the duration ($t_{end} - t_{start}$) and the length ($s_{end} - s_{start}$) of the gesture are calculated.

Figure 2 is exemplary and shows a recognized linear diagonal gesture (horizontal (red): rightward movement, vertical (blue): upward movement) and the corresponding values for t_{start}, t_{end}, s_{start}, s_{end} (horizontal (red) and vertical (blue) marker lines).

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB3

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

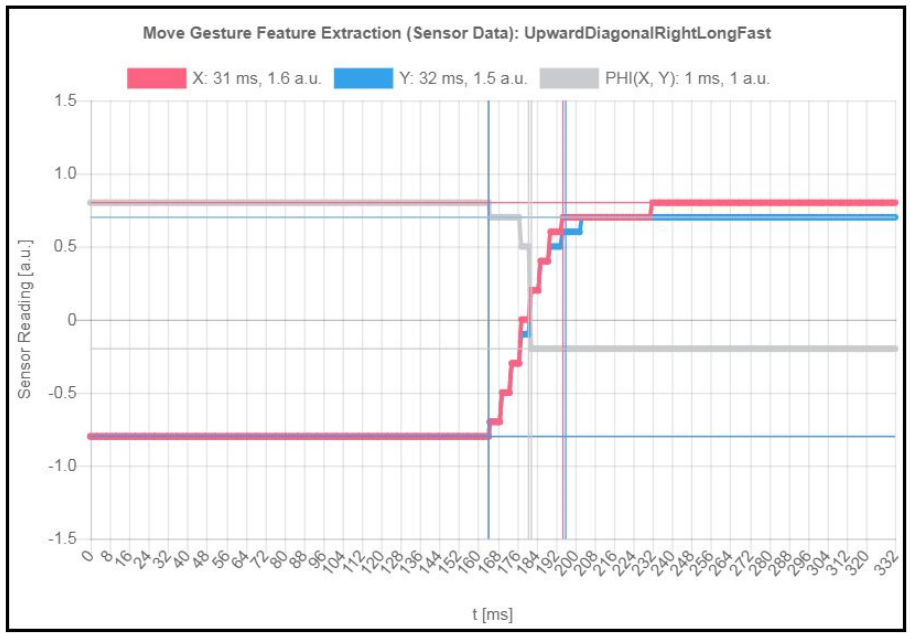


Figure 2: Regression identifying a linear diagonal gesture (horizontal (red): left to right, vertical (blue): bottom to top) and the corresponding values for t_{start}, t_{end}, s_{start}, s_{end} (horizontal (red) and vertical (blue) marker lines).

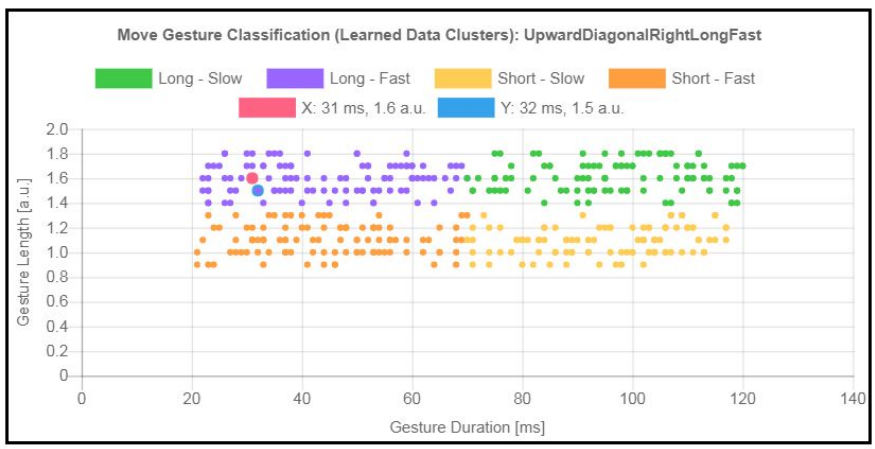


Figure 3: Predicted gesture duration / length (horizontal: red dot, vertical: blue dot) and learned data clusters. The recognized gesture is classified as a linear diagonal (left to right / bottom to top movement) *Long-Fast* gesture.

Classification

Finally, the classification algorithm delivers the orientation and the type of the recognized gesture.

Comparing s_{start} and s_{end} allows distinguishing between left / right linear, up / down linear and clockwise / counter clockwise circular movements.

The duration and length of the gesture are used as input for a k-nearest neighbour (k = 3) analysis. The algorithm calculates the Euclidean distance between the predicted data point (duration / length) and each memorized learned data point. It searches the k nearest neighbours to identify the learned data cluster the predicted data point is belonging to (Fig. 3).

REFERENCES

[1] Web2cToolkit; http://web2ctoolkit.desy.de
[2] TINE; http://tine.desy.de
[3] DOOCS; http://doocs.desy.de
[4] EPICS; http://www.aps.anl.gov/epics
[5] TANGO; http://www.tango-controls.org
[6] STARS; http://stars.kek.jp
[7] LEAP; https://www.LEAPmotion.com
[8] Myo; https://www.myo.com
[9] Sphinx-4; http://cmusphinx.sourceforge.net/sphinx
[10] Epson Moverio BT-200; https://www.vuzix.com/
[11] Vuzix M100; https://www.epson.com/

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB4

LEVERAGING INTERNET OF THINGS DEVELOPMENTS FOR RAPID PROTOTYPING OF SYNOPTIC DISPLAYS

L. T. Stant*, T. Cobb, Diamond Light Source, Oxfordshire, UK

Abstract

Recently the technology industry has been laying foundations for the eponymous Internet of Things (IoT): efficient publish-subscribe protocols; process control schemas for household items; and improved low-power radio communications. Accelerator controls and IoT have several aspects in common - small payloads, low latency, dashboard/synoptic data presentation format are some examples. The IoT now provides several open-source projects which can provide a partial implementation of one or more accelerator controls software features. Because development is typically a lower priority for accelerator controls groups, there is a valid case to try and utilise the free efforts of others for the benefit of accelerator controls. In this paper, the authors present examples of the use of IoT frameworks for synoptic display/GUI development. The work in this paper enables other developers to access this resource and experiment with their own systems.

INTRODUCTION

The IoT is a long-term upgrade of our global interconnectivity which aims to increase the efficiency with which we go about both our personal and professional lives. It will achieve this by adding remote interfaces to everyday items, deferring tedious interactions to increasingly intelligent computer algorithms. The deployment of this technology is slower than expected, especially as it requires standardisation and harmonisation which is less favourable when vendors are developing the technology.

However, we are starting to see the IoT enter our homes in the form of lamps and thermostats. In addition to commercial products, various suppliers along with the open-source community are now offering development platforms and software implementations to facilitate custom IoT solutions. Two of these solutions are of particular interest for controls applications at accelerators - "dashboards" and portable intelligent displays - which we have started to research for use with the EPICS deployment at Diamond Light Source (DLS).

DASHBOARDS

The IoT uses software dashboards as a synoptic user interface (UI) to allow a broad overview of sensory data acquired from distributed devices. An example application could be location trackers attached to livestock, with the dashboard showing time spent at various locations and alarm notifications for individuals moving outside of the allowed area. This UI has a clear analogue with accelerator controls: the livestock are items of equipment, the UI shows performance and status data, and alarms check parameters against operating limits.

Where IoT dashboards can differ from classical controls synoptic UIs is extensibility and visual design. Dashboards need to be created by the user, so UI items are modular and easy to reconfigure. Most dashboard projects solely focus on the UI software itself, so more time is devoted to making the UI visually appealing and attractive. Although for many controls applications the latter point is not important, sleek synoptic displays are often desired around control rooms and high-level status screens. Another benefit of IoT dashboards is that they are often built to be fully responsive to different screen sizes and aspect ratios, including mobile devices. Existing UI solutions, such as Control System Studio [1], offer scaling of the display but are still not optimised for large interfaces on small screens. Where IoT dashboards are typically hosted as a web page, this means that a single interface can be designed and easily accessed on both workstations and mobile devices.

Node-RED Dashboard

Node-RED [2] is a locally hosted web-based visual programming tool for the IoT, written in Node.js [3] (a Javascript framework). It is similar in function to other websites which aggregate and translate packets from different IoT standards (e.g. IFTTT [4]), except that Node-RED uses flow-based graphical programming for configuration. Included in the base software is a UI dashboard, which communicates with the backend via WebSockets (see Fig. 1). The project is open source (Apache 2.0), and it is possible to implement new UI components. The responsive grid layout is well implemented and orders can be assigned to groups of components to help with arrangement on small displays.

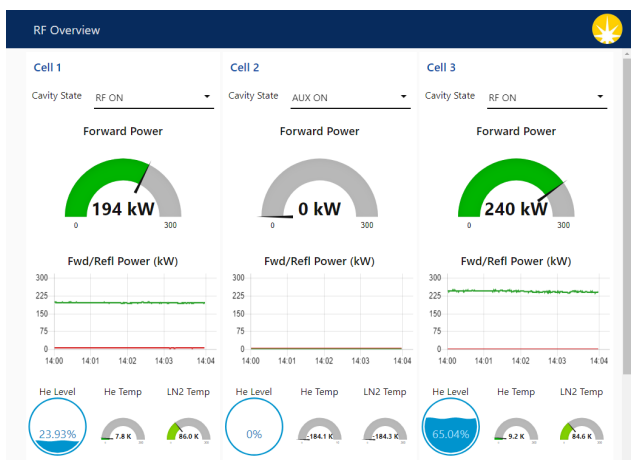


Figure 1: Example Node-RED UI dashboard showing a simple synoptic display.

* laurence.stant@diamond.ac.uk

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Grafana

Although designed for use with monitoring IT infrastructure, Grafana [5] is becoming popular with IoT users due to it's sleek appearance and built-in alarm features. Instead of subscribing to updates as is the case for typical IoT dashboards, Grafana data source providers connect to time-series databases and retrieve archived data. This is similar to existing EPICS tools such as Strip Tool, and therefore it can be used as an alternative. Features which are particularly useful for accelerator controls applications are the alarm handlers and post-processing access. Alarm limits can be set on the dashboard, triggering customisable notifications to a variety of endpoints, and occurrences can be logged to an annotations database for reporting and labelling plots in future. An active alarm UI component can be included to tabulate existing alarms, although they are also displayed on the source component (e.g. graph, gauge). Grafana also provides convenient configuration of the data source "metric" for each UI component, whereby a user can select maths functions to apply to the data for averaging and other functions. Like Node-RED, Grafana is also open source (Apache 2.0), locally hosted and has a responsive layout. An example synoptic screen can be seen in Fig. 2.

Figure 2: An example synoptic UI made with Grafana.

PORTABLE DISPLAYS

Monitoring of process variables (PVs) normally requires a PC workstation to be installed at the location. This poses no issue when many PVs are to be displayed as a screen, or if the workstation is located in a control room or cabin. However, consider the following use cases:

1. Local readout of parameters for headless equipment, both temporary (user) or permanent. This could include both information from the equipment (e.g. position information from stages) and that from the control system, such as operating modes and sequence information.
2. Desktop monitors to allow constant access to crucial values while PC screens may be used for other work, or reduce workspace changes. Rather than repeatedly hunt for a value on a crowded display, a dependent PV could be sent to a dedicated display mimicking a physical readout.
3. Portable maintenance readouts. These devices, preferably wireless, are configured with one or few temporary PVs and can be magnetically attached to equipment such as cabinets and pipework while maintenance is performed. The technician has a real-time monitor of the values read by the machine.

For these use cases, a device smaller than a typical PC workstation is required. We will now look at some options, including new possibilities from IoT developments.

Mobile Phone

Mobile phones capable of internet connectivity have been available for many years, providing controls developers with the possibility to make custom web-based interfaces for portable or desktop monitoring (and control) of PVs. Instead of writing a new interface, IoT developments now provide many dashboard "apps" [6–8] which operate in a similar way to those described in the previous section, although truly optimised for mobile phone displays. These apps connect to IoT protocols, so middleware is required to connect to control system PVs. This will be covered in a later section.

Raspberry Pi

The Raspberry Pi single board computer [9] is a widely used platform for education and development. It is also very convenient to use wherever a linux-capable, small form-factor networked device is required. As such, it fits the use cases listed above well. Many add-on hardware "shields" are available, including displays with button interfaces as shown in Fig. 3. Newer models are Wi-Fi capable, while others can be made so with a usb device not much larger than the port itself. Although there are IoT monitoring software implementations available online [10], the capability of running Linux means the Raspberry Pi can use a standard channel access client without the need for middleware.

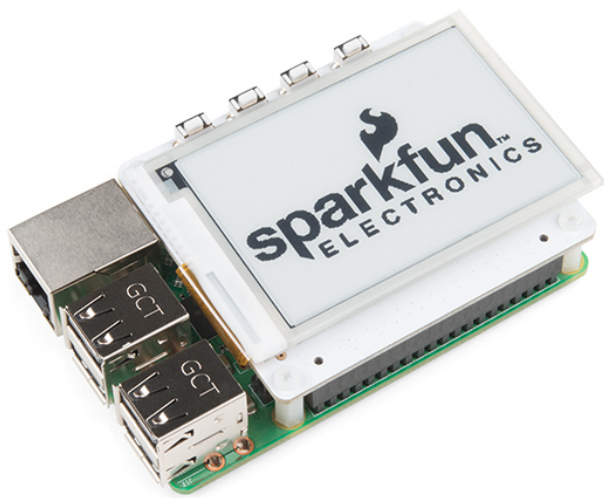


Figure 3: Raspberry Pi single board computer with plugin module featuring an epaper display and tactile button interface.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB4

ESP8266 Platforms

The ESP8266 is a 16-bit microcontroller with built-in Wi-Fi capability, which is either available as a single 5x5 mm QFN32 chip, or, with memory and antenna as a 14x25 mm module. The low price point and ability to deploy code from the popular Arduino IDE has led to it becoming the most popular device for open-source IoT device implementations. The module is often paired with some useful I/O - typically a display, sensor interface, or relay module for driving outputs. It also consumes much less power than Raspberry Pi by using a deep sleep mode, which makes small battery power sources practical (100-300 mAh). Many IoT hardware designs using this module also include a lithium polymer battery charging circuit. There are many open-source implementations of IoT monitoring code for this device, written for a variety of display types and sizes [11–13]. Three popular choices of display are TFT, OLED and ePaper, where the lower power consumption of the latter is advantageous.

ESP8266-based displays appear to be a good solution for portable wireless PV monitors, but where can they be reliably supplied? There are several options available:

1. Develop custom hardware in-house. This is accessible to facilities with standard printed circuit board (PCB) design capability, as there is no specialist ability needed for routing these components. The ESP8266 chip has a reliable supply, as does the alternative 32-bit ESP32. The module can also be procured in several formats, but using a second PCB will increase the size of the finished unit.

2. Procure development boards from online import retailers. There are several well-known importers of electronic goods which stock variations of the same configuration (e.g. ESP8266 and OLED screen with LiPo charger) from many different manufacturers. Although one manufacturer may not be reliable, the low price means that boards can be procured from several to solve any issues. These boards do not include batteries or cases, hence any parts required for a finished device will need to be acquired and assembled separately.

3. Procure complete units. Recently, the same online importers have started to stock the above units complete with cases and batteries. There are fewer manufacturers producing these at the moment, so the supply will be less reliable.

In addition to the above options, 4D Systems® provides an extremely compact device, the IoD-09 [14]. This module, shown in Fig. 4, is only 32x16 mm in size and available from a popular online supplier [15]. Similar to option 2 above there is no battery, charging circuit or case. A 150 mAh battery will fit in the space behind, and a full device with usb charging and magnetic case is being built at DLS.

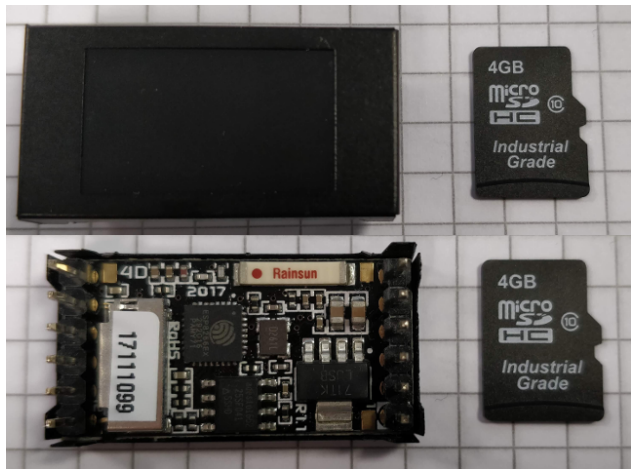


Figure 4: Miniature IoD-09 ESP8266-based TFT display shown alongside MicroSD cards for scale. The square grid has 5 mm spacing.

ARCHITECTURE

Apart from the Raspberry Pi, all of the IoT solutions presented in this paper require either middleware, or plugins to existing software, for them to communicate with the accelerator control system. This section explores these implementations, which are shown connected together in Fig. 5.

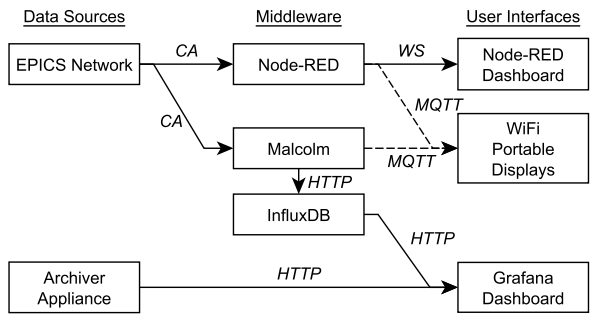


Figure 5: Diagram showing the connections between different components presented in this paper.

MQTT

Message Queue Telemetry Transport (MQTT) is an extremely lightweight broker-based publish-subscribe messaging protocol for the IoT [16]. It is capable of handling large amounts of transactions with reliable transmission via TCP. Because of this, it has been used in applications such as mobile chat [18] and modern railway signalling control [17]. MQTT also includes a quality of service parameter which controls the degree of delivery confirmation.

Client libraries exist for many popular programming languages, and there are several well-established open-source brokers which are simple to deploy and include optional authentication [19, 20]. In addition to PC usage, there are libraries available for IoT hardware devices such as the ESP8266 platform [21, 22]. The combination of a proven

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

track record and widespread support makes MQTT a great option for implementing IoT devices for accelerator controls.

Node-RED

A natural middleware for the IoT, Node-RED's intuitive visual programming style has made it very popular with users. Nodes are dropped onto a "flow" and connected via wires to configure dataflow. Being written in Node.js, it is optimised for event-driven processing which suits message-based IoT communications. To support new devices and web services, Node-RED allow users to easily define new nodes and host them on a public online repository.

For this work, an EPICS client node has been developed which uses an open-source Node.js channel access library [23] which references libca. The EPICS input node attaches a camonitor to the PV specified in the configuration and emits a new message upon update. This is then routed to an MQTT output node which sends a new value to the broker, which in turn publishes an update to all subscribers. Processing nodes are available for filtering and maths functions to be applied.

Malcolm

Another middleware option available at DLS is Malcolm [24]. This software, written in Python, is designed to perform detector scanning in the middle layer and present a standardised virtual device to data-collection software. This is achieved by providing a modular framework which includes channel access and HTTP plugin "blocks". Blocks can be connected like nodes in Node-RED, although the configuration is read from a text file.

A new block for MQTT was implemented using the paho-mqtt library [25]. Supplementary features such as alarms are not currently supported in Malcolm, but this is planned for a future release.

Initial Grafana support was added by pushing PV updates to an InfluxDB instance using a modified HTTP block in Malcolm. InfluxDB is natively supported as a data source by Grafana and can receive updates via HTTP POST requests.

Grafana EPICS Archiver Appliance Data Source

To avoid creating an unmanaged copy of PV archive data using InfluxDB, a new data source was written for Grafana in Typescript (a superset of Javascript). This provides an interface to the time-series database hosted by the EPICS Archiver Appliance [26]. Whenever a screen refresh is triggered in Grafana (either periodically or by changing the time range), the new time range and required metrics are sent to the database for retrieval. The data source translates between the format of the Grafana and Archiver Appliance requests and responses.

Maths functions, allocated to UI components when configured, must be supported by the data source. The Archiver Appliance includes a range of post-processing maths functions which are included with the data source developed in this work.

FUTURE WORK

Management Interface

Configuration of the portable displays is currently achieved by reprogramming the device. This task can be made far easier by providing a web UI which pushes new configurations via MQTT. The device will update immediately to reflect the changes. Users can select the number of PVs and include multiple pages if required, specifying an optional delay for cycling between them in addition to button input.

LPWAN Communications

The IoT is also innovating communication at the hardware layer, driving several new standards in Low Power Wide-Area Networks (LPWANs). These networks provide reliable lightweight communication typically over 5 km in urban environments, and are designed for low power sensing nodes. Although this paper has only studied IoT developments for monitoring purposes, LPWAN technology could provide convenient and reliable wireless inputs to control systems as an alternative to Wi-Fi.

Wearable Devices

Wearable technology is enjoying huge popularity at the moment, with many consumer electronics companies providing new models frequently. These devices, specifically "smart watches", offer a high quality version of a portable display but suffer two disadvantages when compared with those mentioned earlier - they are significantly more expensive and do not offer a consistent development environment for new applications, if at all. However, cheaper versions of these watches are becoming available of which many use the same Wear OS operating system [27]. Some devices do not support Wi-Fi, instead connecting via Bluetooth to a nearby mobile phone. For these devices to connect to EPICS PVs, a service would need to be running on the mobile device.

CONCLUSION

This paper has presented a summary of recent IoT developments which can be utilised by the accelerator controls community to implement synoptic displays and portable PV monitoring devices. We have shown that it is possible to deploy a compatible IoT system with little effort, due to the similarities of the communication requirements between the two areas. Input to the control system has not been addressed due to the lack of maturity of this technology for the application, but may be investigated in the future.

REFERENCES

[1] Control System Studio home page, http://css.desy.de/content/index_eng.html
[2] Node-RED home page, https://nodered.org/
[3] NodeJS home page, https://nodejs.org/en/
[4] IFTTT home page, https://ifttt.com/
[5] Grafana home page, https://grafana.com/

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB4

[6] IoT MQTT Panel app home page, https://play.google.com/store/apps/details?id=snr.lab.iotmqttpanel.prod

[7] Blynk app home page, https://play.google.com/store/apps/details?id=cc.blynk

[8] Cayenne app home page, https://itunes.apple.com/gb/app/cayenne-iot-project-builder/id1057997711?mt=8

[9] Raspberry Pi home page, https://www.raspberrypi.org/

[10] PaPiRus-MQTT Github page, https://github.com/Dom1n1c/PaPiRus-MQTT

[11] esp_mqtt_oled Github page, https://github.com/nathanchantrell/esp_mqtt_oled

[12] e-ink-display-esp8266-mqtt-openwhisk Github project, https://github.com/timwaizenegger/e-ink-display-esp8266-mqtt-openwhisk

[13] AZSMZ-EPAPER Github project, https://github.com/cxandy/AZSMZ-EPAPER

[14] IoD-09 product page, https://www.4dsystems.com.au/product/IoD-09

[15] RS home page, http://rswww.com

[16] MQTT home page, http://mqtt.org/

[17] L. Zhang, "Building facebook messenger", Aug. 2011, https://www.facebook.com/notes/facebook-engineering/building-facebook-messenger/10150259350998920

[18] D. Wood and D. Robson, "Message broker technology for flexible signalling control", Mar. 2014, http://www.irse.org/knowledge/publicdocuments/3.09%20Wood%20-%20Message%20broker%20technology%20for%20flexible%20signalling%20control.pdf

[19] Eclipse Mosquitto MQTT broker home page, https://mosquitto.org/

[20] EMQ MQTT broker home page, http://emqtt.io/

[21] pubsubclient Github page, https://github.com/knolleary/pubsubclient

[22] esp_mqtt Github page, https://github.com/tuanpmt/esp_mqtt

[23] node-epics Github page, https://github.com/RobbieClarken/node-epics

[24] Malcolm home page, https://pypi.org/project/malcolm/

[25] paho-mqtt home page, https://pypi.org/project/paho-mqtt/

[26] M. Shankar, M. Davidsaver, and M. Konrad, "The EPICS archiver applicance", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, pp. 761-764. doi:10.18429/JACoW-ICALEPCS2015-WEPGF030

[27] Wear OS home page, https://wearos.google.com/

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB5

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

INTELLIGENT CONTROLS AND MODELING FOR PARTICLE ACCELERATORS AND OTHER RESEARCH AND INDUSTRIAL INFRASTRUCTURES

S. G. Biedron†,[1], Element Aero, Chicago, IL 60643 USA
A. Edelen, Colorado State University, Fort Collins, CO 80523 USA
E. Meier, H.E.S.T. Australia Limited, Melbourne VIC 3000 Australia
S. Milton, Element Aero, Chicago, IL, 60643 USA
[1]also at University of New Mexico, Albuquerque, New Mexico, 87131 USA

Abstract

We give some perspective on the present state of intelligent control for particle accelerators. This is based on our experience over the past 14 years in developing artificial-intelligence-based tools specifically to address modeling and control challenges found in particle accelerator systems.

INTRODUCTION

Despite the rush to incorporate artificial intelligence (AI) into every component of our lives, not all systems (or subsystems) require intelligent control or data analysis for reliable operation. As the old saying goes, "Just because you can [build it/use it] doesn't mean you should." Some systems, however, can greatly benefit from the responsible and well-architected incorporation of control techniques with intelligent characteristics.

Although other fields, such as computer vision and web search engines, were quick to adopt AI or AI-like tools, the field of accelerator physics has been more reluctant. However, things have advanced in several areas, such as computing and data collection, and these advances have truly enabled the research and subsequent application of AI techniques to particle accelerator. But, as with anything new, the APS News (June 2018, "AI Makes Inroads in Physics") notes there are still challenges [1].

Here we review several examples of complex systems that can benefit from intelligent control methodologies, i.e., cases that were otherwise under-responsive when simpler, less advanced approaches were applied. The case examples on which we focus deal with particle accelerators that are used in many disciplines including fundamental discovery science and engineering.

ACCELERATOR CHALLENGES INTERESTING FOR INTELLIGENT CONTROL AND MODELING SCHEMES

Upon examining the machines used for physics we note that since 1938, accelerator science has an estimated influence on almost one-third of physicists and physics studies, and on average contributed to physics Nobel Prize-winning research every 2.9 years [2]. Further, as one example of the prevalence of particle accelerators, we look to the United States. In support of its science mission, the U.S. Department of Energy (DOE) operates large scientific instruments that support the work of more than 50,000 scientists worldwide. Large particle accelerators are at the core of eleven of the seventeen National User Facilities that DOE operates. Particle accelerators also have roles in the industrial, medical, security, defense, environmental, and energy sectors. The performance goals and complexities of particle accelerators are ever increasing to meet the needs of the science and applications.

Several characteristics of the more complex existing and future particle accelerators are interesting for employing intelligent controls or modeling schemes. For instance, there are many parameters to monitor and control in various sub-systems, and accelerators have many interacting sub-systems. Applications can have on-demand, fast changes in the accelerator operational states. Accelerators can have many small, compounding errors. The accelerator physics of certain machines can be riddled with complex and/or non-linear dynamics having potential detrimental instabilities or other collective effects for effective machine operation. Often, the machine model does not resemble the as-built machine, so there are challenges in diagnosing and/or interpreting the operational behavior from day one of operation. Accelerator systems often exhibit time-varying and non-stationary behavior. Diagnostics for the beam, systems, and peripherals are also often limited because of cost, space, and the inability to physically characterize the overall system at all locations. In addition, even when diagnostics are in existence, they are often not put to full use in control schemes (e.g., images). A final concern is that systems or system peripherals are re-purposed in "new" machines, meaning that the equipment was not originally intended for use in such a configuration/specification.

WHAT DO ARTIFICIAL INTELLIGENCE (AI) AND RELATED TERMS MEAN?

The exact definitions of artificial intelligence and its related techniques are somewhat fluid in terms of interpretation in the community. Figure 1 attempts to help the reader classify the areas of artificial intelligence and related areas covered in this paper [3].

† sgbiedron@elementaero.com

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB5

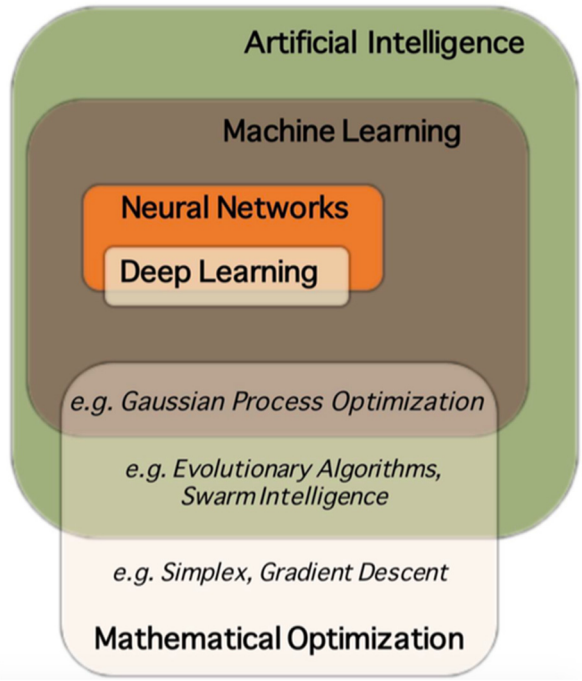


Figure 1: Classification space of artificial intelligence (AI) and related techniques [3].

Artificial intelligence is the concept of enabling machines to have intelligence similar to that of living beings. Where mathematical optimization involves the minimization or maximization of some objective function to select a target with regard to some criteria, machine learning allows some representation of the data to be learned and put to use in a specific task. Neural networks are a common method of machine learning employing many integrated processing units. There are many genres of neural networks and schemes for their training. Deep learning is concerned with hierarchical representations and is often equated with many-layered neural networks.

HOW TO USE AI CONCEPTS FOR ACCELERATORS

AI-based and related concepts can feed into controls engineering and takes inspiration from actual operators of the system. How? These concepts incorporate optimization (finding sweet spots), model learning, planning and prediction, learning control, diagnostic analysis, and system constraints into their models.

EXAMPLES OF RECENT AI-BASED CONTROLS ACTIVITIES

Our team has conducted design architectures and simulations [4] and proof-of-principle experiments in artificial-intelligence-based control of particle accelerators. Here we describe several examples.

On beamlines at the Australian Synchrotron, the Linac Coherent Light Source (LCLS), and the Fermi@Elettra at Sincrotrone Trieste [5-10], we were able to experimentally stabilize the electron beam energy and energy spread in the presence of klystron phase and amplitude jitter. We also demonstrated a novel controller that maintains maximum transmission through the machine.

The resonant frequency of a normal-conducting, radio-frequency (rf) electron gun is controlled thermally with a circulating water system adjusted as needed to the required temperature. At Fermilab, we were able to demonstrate better control of a water system with long, variable time delays due to large thermal time constants and variable rates of water transport. The existing PI controller for this system was unable to stabilize the temperature quickly and produced a large overshoot. For example, in a 1 °C change in the temperature set point of the gun, it takes approximately 25 minutes to settle and produces a maximum overshoot of > 0.5 °C. Using the method of an applied model predictive control (MPC) with a neural network model trained on measured data, the temperature was fixed in under six minutes and without overshoot [11-12].

In another normal-conducting device at Fermilab (a high-intensity, radio-frequency quadrupole (RFQ) to be used for the PIP-II upgrade program) we developed a resonance control system exploring several approaches, including a neural network control policy [13-15].

With the Thomas Jefferson National Accelerator Facility (JLAB) we have been collaborating on approaches for automatic steering tune-up on the now former free-electron laser (FEL) recovery linac. After taking data during tune-up and machine studies, a neural network (NN) model learned the beam position monitor responses to changes in dipole and quadrupole magnets. By interacting with the model, an NN control policy was then used to quickly jump to desired trajectories with specified position offsets [12,16-17].

We are also exploring ways to rapidly switch between operational set-points, such as two output wavelengths in a FEL user facility, or different beam energies in inspection systems. Using a neural network control policy trained over a broad range of operating conditions, one may be able to quickly and automatically execute these actions (for example, switching between different energies while maintaining a match to the undulator [18]).

Another area of exploration involves using a convolutional/fully-connected NN to predict downstream beam parameters for certain machine set-points; this is an important first step toward creating an NN controller that can directly use image diagnostics [19].

CONCLUSIONS

Our collaborations have made some of the first implementations on accelerators around the globe. Additional activities through new grants and contracts will permit additional progress for our community.

In the last ten months, three workshops have been held demonstrating the interest in artificial intelligence techniques as applied to particle accelerators as well as physics in general. The first, “Intelligent Controls for Particle Accelerators,” was held 30-31 January 2018 at Daresbury Laboratory [20]. The second, “ICFA Beam Dynamics Mini-Workshop: Machine Learning Applications for Particle Accelerators,” was held 27 February-2 March 2018 at SLAC

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB5

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

[21]. The third, "Physics Next: Machine Learning," was held by the American Physical Society Physical Review publications group in New York on 8-10 October 2018 [22].

Outside of research and discussions on the technical aspects of artificial intelligence techniques, there are also important ethical discussions underway regarding these powerful tools. The Institute of Electrical and Electronics Engineers (IEEE) is actively engaged in investigating these topics. One significant IEEE activity, established in 2016, is the IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. It is an open, global, and inclusive (regionally, culturally, gender, etc.) community of experts and interested persons from technology and human science. Its mission is to ensure that every stakeholder involved in the design and development of autonomous and intelligent systems is educated, trained, and empowered to prioritize ethical considerations so that these technologies are advanced for the benefit of humanity. One outcome from this initiative is a standard [23]. In addition, the IEEE P7000 series of standardization projects is an ongoing process that seeks to gather more concrete guidelines through consensus in a meaningful time frame [24].

ACKNOWLEDGMENTS

Thank you to our numerous collaborators who have contributed to many of the studies summarized here and described in detail in the references. Many thanks to Sincrotrone Trieste, The Australian Synchrotron, the Office of Naval Research, SLAC, Argonne, JLAB, and Fermilab for funding and/or research support.

REFERENCES

[1] S. Chen, "AI Makes Inroads in Physics," *American Physical Society News*, vol. 27, no. 6, June 2018, https://www.aps.org/publications/apsnews/201806/artificial.cfm

[2] E. Haussecker and A. Chao, "The Influence of Accelerator Science on Physics Research," *Physics in Perspective*, vol. 13, p. 146, 2011, doi:10.1007/s00016-010-0049-y

[3] A. Edelen *et al.*, "Applications of Neural Networks to Modeling and Control of Particle Accelerators," *Fermilab Accelerator Physics and Technology Seminar*, 8 June, 2017.

[4] S. G. Biedron, "Free-electron laser activities," presented at *U.S. Navy's MW-Panel Meeting*, Arlington, Virginia, October 2004.

[5] E. Meier, S. G. Biedron, G. LeBlanc, *et al.*, "Electron Beam Energy and Bunch Length Feed Forward Control Studies Using an Artificial Neural Network at the Linac Coherent Light Source," *Nucl. Instrum. Methods A*, vol. 610, no. 3, p. 629, 2009.

[6] E. Meier, M. Morgan, S. G. Biedron, *et al.*, "Development of a Combined Feed Forward-Feedback System for an Electron Linac," *Nucl. Instrum. Methods A*, vol. 609, nos. 2-3, p. 79, 2009.

[7] E. Meier, S. G. Biedron, G. LeBlanc, and M. J. Morgan, "Development of a Novel Optimization Tool for Electron Linacs Inspired by Artificial Intelligence Techniques in Video Games." *Nucl. Instrum. Methods A*, vol. 632, no. 1, p. 1, 2011.

[8] E. Meier, S. G. Biedron, G. LeBlanc, and M. J. Morgan, "Artificial Intelligence Systems for Electron Beam Parameters Optimisation at the Australian Synchrotron," in *Proc. IPAC'10*, Kyoto, Japan, May 2010, pp. 1770-1772, https://jacow.org/IPAC10/papers/tupec025.pdf

[9] E. Meier, S. G. Biedron, G. LeBlanc, *et al.*, "Electron Beam Energy Stabilization Using a Neural Network Hybrid Controller at the Australian Synchrotron Linac," in *Proc. PAC'09*, Vancouver, Canada, May 2009, pp. 1201-1203, https://jacow.org/PAC2009/papers/tu5rfp050.pdf

[10] E. Meier, S. G. Biedron, G. LeBlanc, *et al.*, "Electron Beam Energy Stabilisation Test Results Using a Neural Network Hybrid Controller at the Australian Synchrotron and LINAC Coherent Light Source Projects," in *Proc. FEL2009*, Liverpool, UK, August 2009, pp. 766-771, http://jacow.org/FEL2009/papers/froa02.pdf

[11] A. L. Edelen, S. G. Biedron, S. V. Milton, *et al.*, "Initial experimental results of a machine learning-based temperature control system for an RF gun," in *Proc. IPAC'15*, Richmond, VA, USA, May 2015, pp. 1217-1219, doi:10.18429/JACoW-IPAC2015-MOPWI028

[12] A. Edelen *et al.*, "Recent Application of Neural Network-Based Approaches to Modeling and Control of Particle Accelerators," presented at IPAC'18, Vancouver BC, Canada, April-May 2018, slides THYGBE2, http://www.JACoW.org.

[13] A. Edelen, S. Biedron, S. Milton, *et al.*, "Resonant Frequency Control for the PIP-II Injector Test RFQ: Control Framework and Initial Results," in *Proc. NA-PAC2016*, Chicago, IL, USA, October 2016, pp. 109-112, doi:10.18429/JACoW-NAPAC2016-MOPOB17

[14] D. Bowring, B. Chase, J. Czajkowski, *et al.*, "Resonance Control for Fermlab's PXIE RFQ," in *Proc. IPAC'16*, Busan, Korea, May 2016, pp. 447-450, doi:10.18429/JACoW-IPAC2016-MOPMW026

[15] A. L. Edelen, S. G. Biedron, S. V. Milton, *et al.*, "Neural Network Model of the PXIE RFQ Cooling System and Resonant Frequency Response," in *Proc. IPAC'16*, Busan, Korea, May 2016, pp. 4131-4133, doi:10.18429/JACoW-IPAC2016-THPOY020

[16] A. Morin, S. G. Biedron, S. V. Milton, *et al.*, "Trajectory Response Studies at the Thomas Jefferson National Accelerator Facility Energy Recovery Linac and Free Electron Laser," in *Proc. 16th Annual Directed Energy Symposium*, 10-14 March 2014, Huntsville, Alabama, Directed Energy Professional Society, https://protected.networkshosting.com/depsor/store/merchandise/TOCs/DEsymp14-TOC.html

[17] A. Morin, C. Tennant, S. Biedron, D. Douglas, and S. Milton, "Control Systems Development for the Thomas Jefferson National Accelerator Facility Free Electron Laser: Preliminary Results for Trajectory Response Data," in *Proc. 15th Annual Directed Energy Symposium*, 26-30 November 2012, Albuquerque, New Mexico, Directed Energy Professional Society.

[18] A. Edelen, S. Biedron, J.P. Edelen, S.V. Milton, and P.J.M van der Slot, "Using A Neural Network Control Policy For Rapid Switching Between Beam Parameters in an FEL," in *Proc. FEL2017*, Santa Fe, NM, USA, August 2017, pp. 480-483, doi:10.18429/JACoW-FEL2017-WEP031

[19] A. Edelen and S. Biedron, "First Steps Toward Incorporating Image Based Diagnostics into Particle Accelerator Control Systems Using Convolutional Neural Networks," in

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THCB5

Proc. NA-PAC2016, Chicago, IL, USA, October 2016, pp. 390-393, doi:10.18429/JACoW-NAPAC2016-TUPOA51

[20] www.cockcroft.ac.uk/events/ICPA/

[21] conf.slac.stanford.edu/icfa-ml-2018/

[22] https://journals.aps.org/physics-next/2018/machine-learning

[23] Private discussions Alpesh Shah, alpesh.shah@ieee.org, and Sam Sciacca, s.sciacca@ieee.org

[24] https://ethicsinaction.ieee.org/standards/p7000/

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP02

CURRENT STATUS OF THE RAON MACHINE PROTECTION SYSTEM DEVELOPMENT

Hyunchang Jin*, Yongjun Choi, Sangil Lee
Rare Isotope Science Project, Institute for Basic Science, 34047 Daejeon, Korea

Abstract

For the RAON accelerator that transports beams with high energy and power, a machine protection system (MPS) that protects each device from sudden beam loss is necessary. For this reason, we have been preparing for the development of the MPS with the start of the RAON accelerator construction. For effective MPS operation and stable accelerator operation, we divided the MPS into four subsystems: fast protection system, slow interlock system, run permit system, and post-mortem system. Among them, the FPGA-based fast protection system and the PLC-based slow interlock system have been tested by prototypes and are currently working on the mass production. The run permit system and the post-mortem system are also undergoing basic design and software development. In this paper, we will describe the progress of the MPS development through detailed hardware and software development in the RAON accelerator and explain the future plans.

INTRODUCTION

The RAON accelerator [1] produces various kinds of beams from proton to uranium to accelerate and transport to the target in the experimental hall. These beams can be accelerated to an energy of about 200 MeV/u, with a power of up to 400 kW. Various devices such as ion sources, magnets, superconducting cavities, etc. are installed to generate and accelerate beams, and sudden interlocks in the devices lead to the beam loss during operation. This beam loss can cause significant damage to the device, and therefore a machine protection system (MPS) is required to minimize such damage in the RAON accelerator. Currently, we are developing the MPS into four parts for efficient operation: a fast protection system (FPS), a slow interlock system (SIS), a post-mortem system (PMS), and a run permit system (RPS). First, the FPS is fabricated on an field programmable gate array (FPGA) basis and collects interlock signals from the beam loss monitor (BLM), beam current monitor (BCM), low-level radio-frequency (LLRF) system, etc. within a few tens of microseconds and sends the beam stop signal to the mitigation devices like a chopper, RFQ LLRF, and so on. Secondly, the SIS is fabricated on programmable logic controller (PLC) basis and collects interlock signals from devices such as cryogenics, vacuum, etc., and transmits interlock information to the FPS within a few tens of milliseconds. Thirdly, the PMS is a system for storing and analyzing information of interlock signals to prevent future accidents. At last, the RPS is a system for determining beam operation status by checking the state of accelerator devices corresponding to a machine mode and a beam mode before operation. Figure 1 shows the configuration of the RAON MPS.

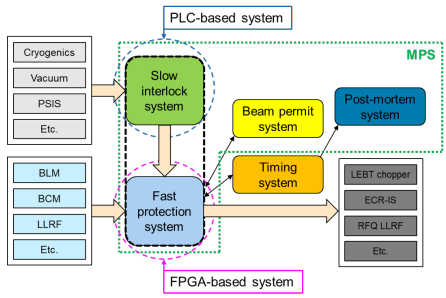


Figure 1: Layout of the RAON machine protection system. The machine protection system consists of the fast protection system, the slow interlock system, the post-mortem system, and the run permit system.

FAST PROTECTION SYSTEM

The RAON FPS consists of a mitigation node acting as a master and an acquisition node acting as a slave. This FPS consists of two mitigation nodes including a back-up and dozens of acquisition nodes as shown in Fig. 2. The prototype of the FPS was developed in 2016 and successfully tested at the RISP test facility. The test results of the prototype were presented at ICABU2017 workshop [2]. Based on these results, The FPS product, including one mitigation node and seven acquisition nodes, is currently under development [3] and will be finished at the end of 2018.

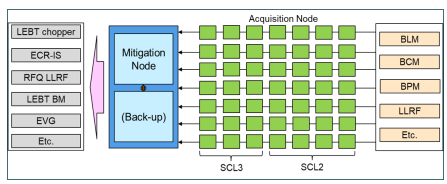


Figure 2: Layout of the fast protection system which has mitigation nodes and acquisition nodes. The acquisition node collects the interlock signals, and the mitigation node sends signals to mitigation devices.

The Mitigation node is manufactured using Xilinx Zynq ultrascale+ XCZU9EG chip. In this node, eight SFP+ transceivers are installed for optical communication, and 24 sub-miniature version A (SMA) ports are located on the back panel for signal transmission with mitigation devices. The printed circuit board (PCB) of the mitigation node is now being manufactured, and the test is in progress. Figure 3 shows the wiring diagram of the mitigation node. The

* hcjin@ibs.re.kr

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP02

main board where the Zynq chip is located is connected to the daughter board which sends the output signal, and power is supplied through the switched-mode power supply (SMPS). Unlike mitigation node, the acquisition is manufactured using Xilinx Zynq XC7Z100 chip, and the wiring diagram of the acquisition node is shown in Fig. 4 and the electric power is supplied through SMPS with lower power than mitigation node. Sixty-four interlock signals collected through the daughter board are sent to the main board.

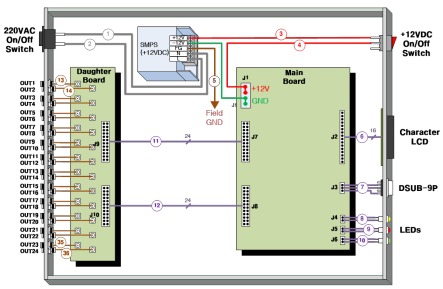


Figure 3: The connection diagram of the mitigation node.

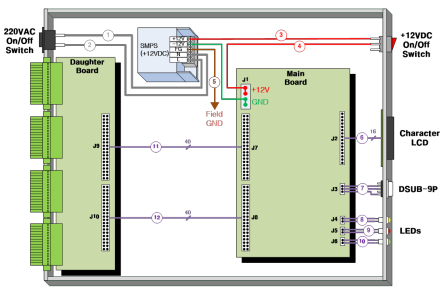


Figure 4: The connection diagram of the acquisition node.

SLOW INTERLOCK SYSTEM

RAON SIS collects interlock signals from the PLC controlled devices such as cryogenics, vacuum, etc. and then transmits the information to the acquisition node of the FPS. The SIS prototype was also developed in 2016, and the performance verification was successfully completed at the RISP test facility [2]. Based on the prototype results, the SIS production is currently in progress. The configuration of the SIS is shown in Fig. 5. The SIS PLC in a control rack consists of a CPU for signal processing, a module for 16 digital outputs, and modules for 144 digital inputs. The bit processing time of the CPU is less than 48 nanoseconds. In addition, a signal relay, a circuit protector, and a SMPS are located below the SIS PLC. These SIS sets will be located in 23 control racks for low energy superconducting linac section and collect nearby local interlock signals.

POST-MORTEM SYSTEM

The development of the PMS is in progress with the development of the FPS, and the signal processing of the PMS is as shown in Fig. 6. When the interlock signal is collected at the acquisition node, the signal is transmitted from the

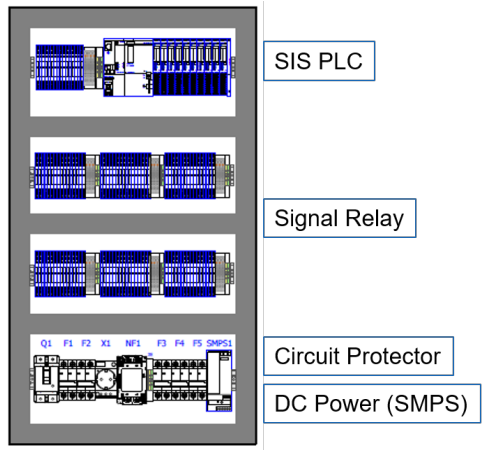


Figure 5: Layout of the slow interlock system in the control lack. It consists of a SIS PLC, a signal relay, a circuit protector, and a DC power.

mitigation node to the event generator (EVG) of the timing system. After that, when the signal is transmitted from the EVG to the acquisition nodes, the interlock data is saved at the acquisition node memory for ±200 milliseconds in a unit of 1 microsecond. The stored information can then be verified at the CS-Studio (CSS) operator interface (OPI) through the EPICS IOC. To develop and verify this PMS, we are testing using FPS hardware under development, and the block diagram in this test is shown in Fig. 7. The CSS screen of the PMS is shown in Fig. 8.

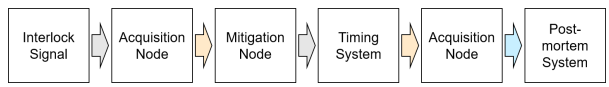


Figure 6: Signal flow of the post-mortem system. The PMS data is saved at an acquisition node memory and sent to the data server through the EPICS IOC.

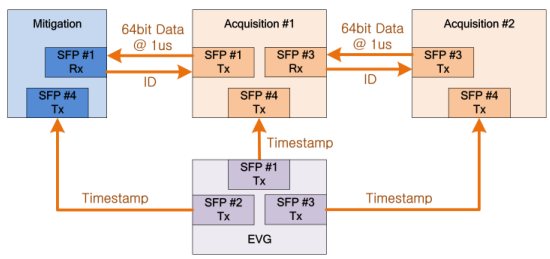


Figure 7: Block diagram of the post-mortem system test.

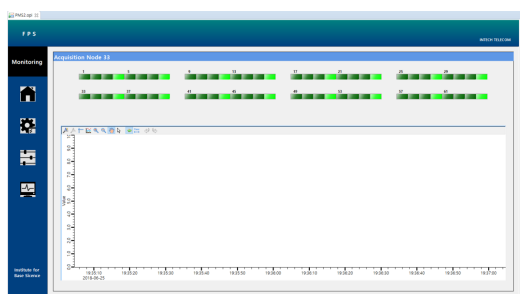


Figure 8: CSS screen of the post-mortem system.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP02

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

RUN PERMIT SYSTEM

The RAON accelerator has various operating modes because various types of beams are used for experimental purposes. Therefore, for efficient operation, the accelerator operation mode is divided into a machine mode for determining the devices to be used and a beam mode for determining the type and state of the beam. Tables 1 and 2 show examples of machine mode and beam mode in the RAON accelerator, respectively. More accurate information will be determined as commissioning proceeds with the beam physics group. Figure 9 shows the CSS test screen of the current RPS. This RPS operates with standby, run, fault, and stop states. Since 2019, The accelerator equipment will be installed in accelerator tunnels, and the RPS upgrade will be continued.

Table 1: Machine Mode Example

Mode	Description	Remark
MM0	Maintenance	No beam
MM1	LEBT tuning	Ion source, RFQ
MM2	MEBT tuning	Rebuncher
MM3	Linac tuning	SCL3, SCL2
MM4	P2DT tuning	Buncher, charge stripper
MM5	Secondary beam test	Target
MM6	User operation	Target

Table 2: Beam Mode Example

Mode	Description	Remark
BM0	Maintenance	No beam
BM1	Pulse mode	Time structure, beam intensity
BM2	CW mode	Normal operation, beam power

SUMMARY

We have presented the current development status of the RAON MPS which includes the FPS, the SIS, the PMS, and the RPS. Based on the prototype results of the FPS, the FPS product is under development using Zynq chips, and it will be finished at the end of 2018. The FPS collects the interlock signals from BLM, BCM, LLRF, etc. and sends a beam stop signals to the mitigation devices within a few tens of microseconds. The SIS product is also being manufactured using a PLC and will collect the interlock signals from the PLC controlled devices. The development of the PMS is proceeding with the FPS development, and a data saving testing is in progress. For the efficient accelerator operation, the RPS is also being developed with the machine mode and the beam mode, and the upgrade of the RPS will be continued with the installation of the accelerator equipment after 2019.

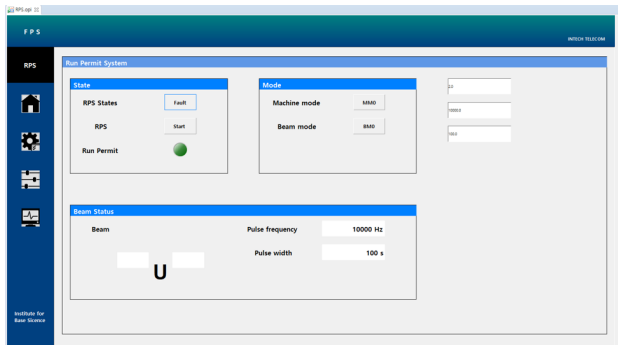


Figure 9: CSS test screen of the run permit system.

ACKNOWLEDGEMENTS

This work was supported by the Rare Isotope Science Project of Institute for Basic Science funded by Ministry of Science, ICT and Future Planning and National Research Foundation of Korea (2013M7A1A1075764).

REFERENCES

[1] D. Jeon *et al.*, "Design of the RAON accelerator systems", *Journal of the Korean Physical Society* 65.7 (2014).

[2] ICABU2017, https://indico.cern.ch/event/647155/

[3] H. Jin *et al.*, "Progress and plan of the fast protection system in the RAON accelerator", *in Proc. HB'18*, Daejeon, Korea, June 2018. doi:10.18429/JACoW-HB2018-WEP2PO015

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP03

MARVIN UPDATE – THE ROBOTIC SAMPLE MOUNTING SYSTEM AT THE EMBL-HAMBURG

U. Ristau, S. Fiedler, European Molecular Biology Laboratory, EMBL, Hamburg Unit, Germany

Abstract

This article aims at giving an overview about the controls of the robotic sample mounting system MARVIN (MultiAxesRoboticVersatileINstaller) that are installed at EMBL beamlines at the PETRA III synchrotron (DESY, Hamburg, Germany). Currently, two inhouse built systems are in user operation at the beamlines P13 and P14 dedicated to macromolecular crystallography (MX). The different sub-systems and the embedding into BICFROK, the EMBL Hamburg beamline's control framework [1], and especially, new developments to decrease downtimes, as well as system recovery routines, will be described in detail.

INTRODUCTION

Robotic sample mounters are state-of-the-art instrumentation at protein crystallographic beamlines [2]. In general, their function is to mount and dismount sample holders with cryogenically frozen protein crystals onto the head of a diffractometer axis. Samples are stored in liquid nitrogen (LN2) filled dewars and rapidly transferred by the robotic system into a cold gaseous nitrogen stream in order to increase their lifetime in the synchrotron beam. The standardized SPINE sample holders [3], that are used with this instrument and that consist of a cap and a pin with the microscopic crystal attached to its tip, are individually inserted into containers called pucks [4]. In the pucks used for MARVIN and developed by EMBL [5] ten sample holders can be inserted see Fig. 1.

As the robotic systems can considerably improve speed and reliability of the mounting, the EMBL in-house development of a first similar system started already in 2002 and was initially in user operation at a former DORIS synchrotron beamline BW7b at DESY [6]. In 2013, the commissioning of the first new MARVIN system equipped with a Stäubli industry robot TX60L [7] began. The first system is in user operation since 2014 and the second system serving a diffractometer with vertical spindle axis operates since 2016 at the P14 beamline. In 2017, about 30000 samples have been mounted with the two systems.

SUB-SYSTEMS AND INTERFACING

The robot is installed overhead inside a closed cage centred above the LN2 storage dewar. In this dewar 170 SPINE samples in 17 pucks can be stored. The sample storage dewar is equipped with a pneumatically driven lid to open and close the dewar. The LN2 level inside the dewar is maintained by an automatic filling system fed from a central LN2 supply.

Apart from mounting the samples the system has also other functions. After a user has inserted a puck manually on a dedicated loading base, the robotic system distributes it automatically to one of the storage bases. The sample mounting takes always place from the central base. Therefore, a puck has to be shuffled automatically onto the central base before sample mounting.

In order to guarantee a high degree of automation and a high sample throughput as well as a high level of safety, the interaction with several other components is required. Apart from the sub-systems belonging to the core of the MARVIN system like the industry robot, the storage dewar with position sensitive puck switches, the cryogenic system for the dewar refilling, the pneumatic system for the control of lid, robot gripper and a guillotine shutter installed at the robot cage and the safety system, independently controllable instruments have to be interfaced and procedures synchronised. These devices are the diffractometers like the 'MD3' [8], experimental tables, the area detectors mounted on detector translations, cryogenic gasflow sample cooling units, and fluorescence detectors for anomalous phasing experiments.

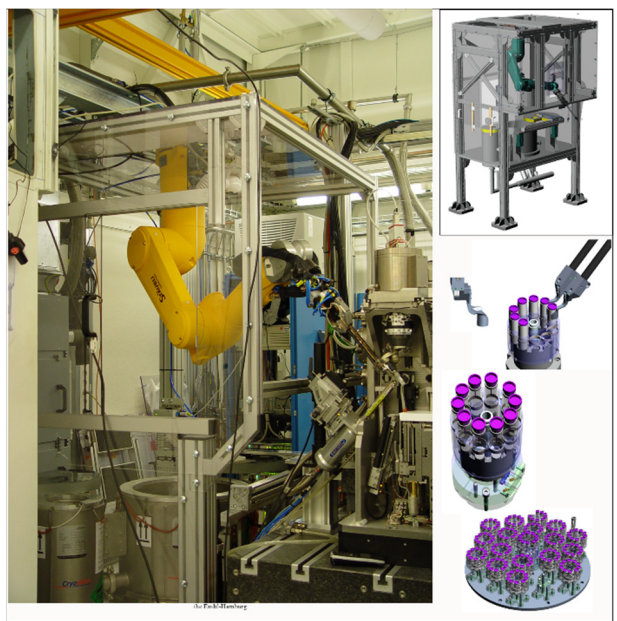


Figure 1: MARVIN mounting a sample on the MD3 Goniometer. And robot gripper, samples Pucks and Puck storage plate on the right.

Redundant signals and sensor cross-checks have been added in order to improve the reliability of the system. Examples are a vial detection switch at the robot grippers, gripper temperature control to avoid icing of the robot grippers, detection of the opening state of the grippers, tool changer detection signal and a crash protection state signal. While in the first sample changer system no detection of samples and pucks was present redundant sensors check in the current system the positioning accuracy to avoid sample losses and deformations of the grippers and potential damaging of the other instruments involved in the process.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP03

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

As an example, two independent sensors detect in the LN2 environment the presence of a puck on the central base. Hardware signals are used in general for the handshakes between the robot controller and the devices to be interfaced. The only exception is implemented for the change between two distinct mounting positions on the P14 diffractometer (toggling between two optical setups) for which a software handshake is in place.

The MARVIN system comprises also a personnel safety module. An important part is to ensure that the sample changing takes place only if all users have left the experimental area and the hutch to this area is interlocked. This precaution has been taken in particular to avoid crushing hazards with the heavy duty detector stage that is moved during the sample changing process. It is guaranteed in the standard mode for external users by electrical safety switches that are integrated into the door interlock of the experimental hutch. However, during the loading of the pucks onto the storage bases users have to be present inside the experimental hutch. For this process, only movements of the robot inside the safety cage are carried out with the door of the cage closed and interlocked. There exists also a password protected maintenance mode in which samples can be mounted with open experimental hutch door.

SYSTEM CONTROL

The instrument control of the instruments at the EMBL beamlines is performed by a modular control software package BICFROK and by standardized control electronics as a Master Electronic Unit installation for each major component like the robotic sample changer.

Controller for the TX60L robot is the Stäubli CS8C. The Stäubli Robotics Suite offers VAL3 as API to program the robot trajectories and to all DAQ signals to operate the robot. Both, the CS8C and the Master Electronic Unit have a Profibus fieldbus interface card to enable communication between them. The fieldbus protocol of the Master Electronic Unit is EtherCAT. There are about 50 electrical signals digital and analogous in use to run the MARVIN sample changer.

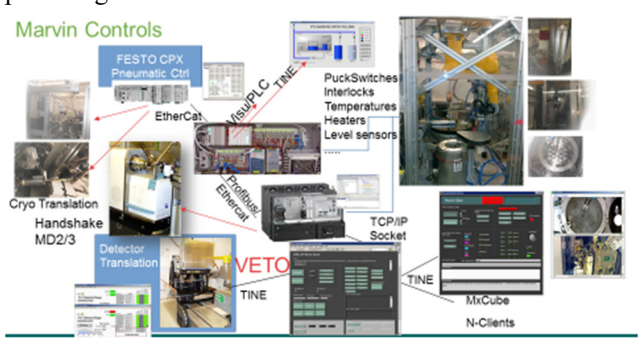


Figure 2: Sketch of the control devices and Server and Client Graphical User Interfaces.

Tasks of the Master Electronic Unit see Fig. 2 are the control of the automated nitrogen filling system for the cryogenic dewar, the robot cage access control and the alarming in case of low oxygen level inside the enclosed robot cage. An alarm can be active only if a user access request to the Robot cage is requested. In case of a too low oxygen level inside the robot cage visible and acoustical alarms are produced.

In the following the electrical handshake signals between robot and MD3 are described. On command to the robot Server to mount or dismount a sample the request signal is set as an electrical high signal to the MD3 controller.

Next the MD3 controller sets all its involved axis to predefined positions called transfer position and replies after finishing that the device is ready for the sample transfer. The CS8C Robot server program changes a robot idle digital signal to busy and the robot starts a motion to mount or dismount a sample depending on which request has been send. As soon as the mount/dismount process has finished the signal of the CS8C controller is set back to the idle state and the mount or dismount request is erased.

As an additional important diagnostic, a sample detection signal is implemented on the MD2/3 which proofs that the robot mount or dismount process has been successful. In case of unsuccessful motions, the motion will be repeated by the robot. If there is still no success the experimental user has the chance to correct the situation by manual interaction.

Inside the dewar position sensitive switches are in place to detect that pucks inside the dewar are proper placed. Malfunction of these switches can lead to robot motion errors or crashes of the robot. In case of inconsistence between software state and hardware state a warning or error message is raised by the robot server and a user interaction is demanded.

All controls related to air pressure are driven by an FESTO [9] control island with EtherCAT fieldbus. The actuator control and the position detection are part of the system. The functionalities of lifting of the dewar lid, opening the robot gripper, actuation of a pneumatic actuated translation and a control of a proportional valve to control the air pressure are beside others part of this control unit.

CONTROL SOFTWARE

The control concept at the EMBL Petra beamlines is based on the described approach of a hierarchical structured server architecture. Heart of the controls is the TINE [10] control system. In the layered control structure the CS8C robot server is the front end controller device server.

Servers are written in Val3 for the robots and in G-Code using LabVIEW [11] for the MARVIN TINE Server. The Marvin controls as part of the beamline control software is visualized in Fig. 3. The MARVIN server is part of the framework of the EMBL instrumentation framework BICFROK. Clients programmed with LabVIEW, the PLC programming of the Beckhoff [12] fieldbus system is implemented with the ‘structured text’ language.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP03

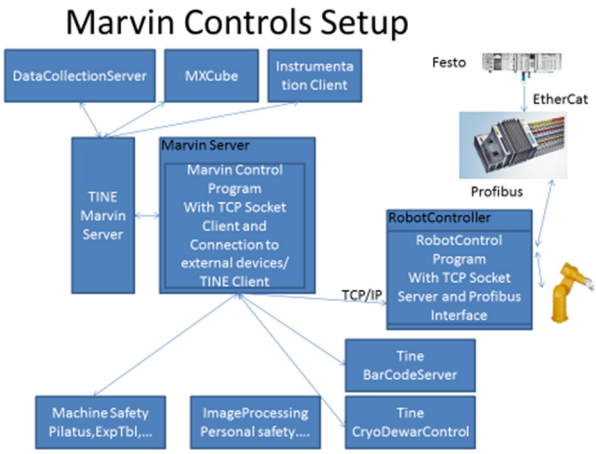


Figure 3: The MARVIN controls integrated in the beamline controls.

Server

The MARVIN TINE server is acting as a sequencer which coordinates the I/O signals and provides the sequencing of the commands called on the CS8 controller. The low-level device server of the robot is executed on the robot controller itself. The code splits the process in tasks of defined priority where each of the tasks can execute n programs.

The implemented tasks are the communication task, the status task, the state machine task and the exception task. The function blocks called programs are the mirror of the functionality offered by the robot. Examples are the open gripper or the mount sample program.

A TCP/IP Socket server is chosen to communicate remotely with other devices. The server uses a string communication to communicate with the TINE Robot server which is in the control hierarchy one level above the Robot Server.

The TINE Robot server communicates with low level Stäubli server and coordinates the interfacing devices. All functionalities are exported as TINE properties. The Server logs all possible statuses and commands and exports system statuses, warnings, error messages and task progress of the system.

Figure 4: MARVIN Clients as part of the BICFROK project and the User Client as part of a software called MXCube.

Clients

The asynchronous programmed event driven TINE clients Fig. 4 provided by the system are G-Code LabVIEW programmed GUIs. N-Clients can connect to the server. A TINE access control makes sure that only defined user have the right to control the system. The LabVIEW Clients can be exported as web clients for remote monitoring or password protected also as a web control application as part of the EMBL Hamburg BICFROK framework. Beside the main client for the MARVIN operation a LabVIEW web browser client is in place which displays two camera perspectives to monitor the robot actions.

Monitoring and Debugging

The Robot controller is connected via Profibus to the Master Electronic Unit DIOs. All commands send to the robot are logged. For easier debugging of system errors and tracking of mal functions there are two cameras for each of the systems available. Sequences of robot motions can be optionally saved.

All relevant signals of the detector translation, of the MD3 and of the communication handshakes are stored by the TINE archiving system. The TINE Archive Viewer synchronizes all signal and software states and allows to correlate them to each other.

After each of the user groups has finished its experiments an experiment debriefing takes place to track potential errors and to identify and correct them.

SUMMARY & OUTLOOK

Both EMBL MARVIN sample changer at the PETRA III beamline are in daily user operation. Plans for further improvements of the sample changer are the increase of the sample through put due to requests for shorter cycle times and the integration of new pin standards like the mini SPINE standard.

Optional the integration of a robotic plate screening which is available as a prototype presented for the MD2 at Beamline P13 are possible System extensions.

REFERENCES

[1] U. Ristau, A. Kolozhvari and S. Fiedler, "The EMBL Beamline Control Framework BICFROCK", in *Proc. PCaPAC'14*, Karlsruhe, Germany, Oct. 2014, paper WPO012, pp. 60-62.

[2] J. D. O'Hea, M. Burt, S. Fisher, K. M. J. Jones, K. E. McAuley, G. Preece, M. A. Williams, "Development of a Sample Exchange System for MX Beamline", in *Proc. ICALEPCS'17*, Barcelona, Oct. 2017, paper THPHA200, pp. 1920-1922.

[3] SPINE definition https://instrumentation.embl.fr/spinesampleholder/

[4] U. Ristau *et al.* "Control Concept of the new EMBL-Hamburg Sample Changer", in *Proc. PCaPAC'08*, Ljubljana, Slovenia, Oct. 2008, paper WEP019.

[5] F. Cipriani *et al.*, "Automation of sample mounting for macromolecular crystallography", *Acta Crystallographica Section D Biological Crystallography*, vol. 62, pp. 1251-1259. November 2006, doi: 10.1107/S0907444906030587

[6] B. Pohl *et al.*, "Automation of the EMBL Hamburg Protein Crystallography Beamline BW7", Oct. 2004, *Journal of*

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP03

Synchrotron Radiation, vol. 11, pp. 372-377. doi: 10.1107/S090904950401516X

[7] Site of Stäubli Company: www.staubli.com

[8] Site of Arinax MD3: https://www.arinax.com/md3-high-precision-x-ray-microdiffractometer/

[9] FESTO company site, http://www.festo.de/

[10] TINE web page at DESY: http://adweb.desy.de/mcs/tine/

[11] Site of the National Instruments Company: http://www.ni.com/Labview

[12] Site of the Beckhoff Company: http://www.beckhoff.com/

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP04

REAL-TIME AND DETAILED PROVISION OF J-PARC ACCELERATOR OPERATION INFORMATION FROM THE ACCELERATOR CONTROL LAN TO THE OFFICE LAN

S. Yamada*, KEK / J-PARC Center, Tokai, Ibaraki 319-1195, Japan

Abstract

J-PARC Main Ring (MR) is a high-intensity proton synchrotron whose control system is developed based on EPICS. Its beam operation started in 2008, and since 2009 has been delivering beam to the T2K neutrino experiment and hadron experiments. Over the past decade, MR have become more sophisticated and more stable driving is required. Along with this, demands arose from users and experts of equipment such that acquiring detailed and real-time information on the apparatus from the office LAN. On the other hand, the accelerator control system is quarantined from the office LAN with firewall for security reasons. Therefore, despite being intentional or not, manipulating any equipment in the accelerator control LAN shall be prohibited from the office LAN. This article describes construction and prospects of such an one-way gateway system such that information is relayed via EPICS from accelerator control LAN to the office LAN while minimizing influence in the opposite direction.

INTRODUCTION

The office LAN in J-PARC is named “JLAN”.

J-PARC accelerator operation information has been provided to JLAN in a web page containing two items as following:

- “Latest Operation Status” which is manually typed by the accelerator shift leader as necessary.
- A image file showing summary of operation status, which is updated every one minutes.

Its example is shown in Fig. 1.

As MR become more sophisticated since it stated its operation in 2008, yet more stable operation is required. Accordingly, demands from equipment experts and users are increasing to acquire real-time and detailed status of the accelerators and equipment from their office. Those information such as

- detailed status of power supplies for magnets and RF,
- Present value of beam pipe vacuum as well as its history,
- Temperatures in power supply buildings and their history,

were only available in the control LAN but in JLAN.

EPICS AND CA GATEWAY

The control system of J-PARC Accelerator is based on EPICS [1]. Its protocol is called Channel Access (CA). CA is available within the same network under normal usage.

A front-end computer, which provides control points to access equipment under its control, is called I/O controllers (IOC). The operator interface (OPI), which act as a CA client, broadcasts UDP packet to search for a control point of interest. IOC will reply to the client when it is hosting the requested control point.

CA Gateway [2] is a standard EPICS utility to relay CA between two networks. It works as a CA client in one network and a as server in the other, so that accelerator operation status information in the control LAN will be available in JLAN in realtime.

MINIMIZATION OF INFLUENCE TO THE ACCELERATOR CONTROL LAN USING TWO-TIER GATEWAY

Accelerator control LAN is connected to JLAN via a firewall device. There is DMZ in the middle of those two networks, which is called “control-DMZ”. The policies in communications among those three networks are as following:

- No bridge connection is allowed other than the firewall device. Therefore any communication shall go through the firewall.
- Any direct communication between control LAN and JLAN is prohibited.
- Communication between control LAN and control-DMZ is allowed only if source IP address, destination IP address, and port number is listed in a whitelist.
- Communication between control-DMZ and JLAN is also limited by another whitelist.

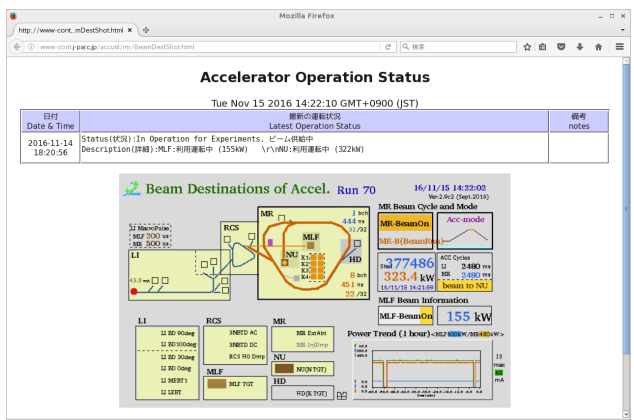


Figure 1: A web page which shows J-PARC accelerator status.

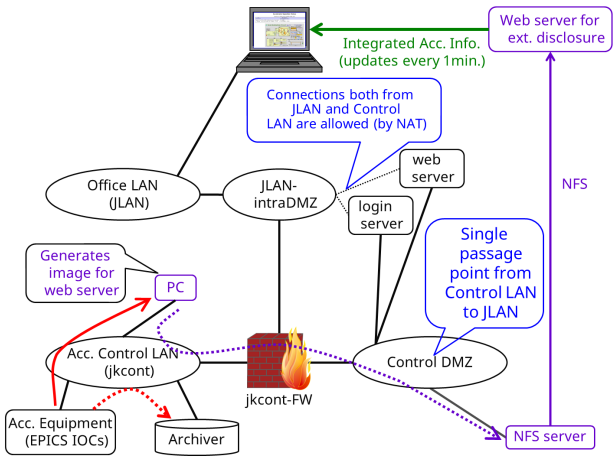
* shuei@post.kek.jp

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

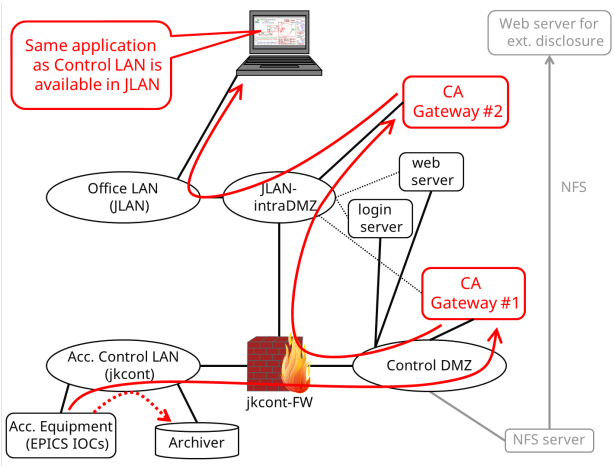
ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP04

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Figure 2a shows traditional path of accelerator operation status information from control LAN to JLAN. Every one minutes a PC connected to the control LAN generates an image file containing accelerator operation summary and saves to a file server in control-DMZ. Then a web server reads the image file and publishes to JLAN. There is no chance to operate any accelerator component from JLAN, because only an image file is provided from control LAN to control-DMZ and the web server has read-only access to the file server.



(a) Before introduction of the gateway system.



(b) After introduction of the gateway system.

Figure 2: Transmission path of information from the accelerator control LAN to JLAN.

As mentioned above, CA Gateway makes it possible to relay the CA protocol used by the control system to JLAN. Accelerator operation information is provided to JLAN in real time with the route shown in Fig. 2b.

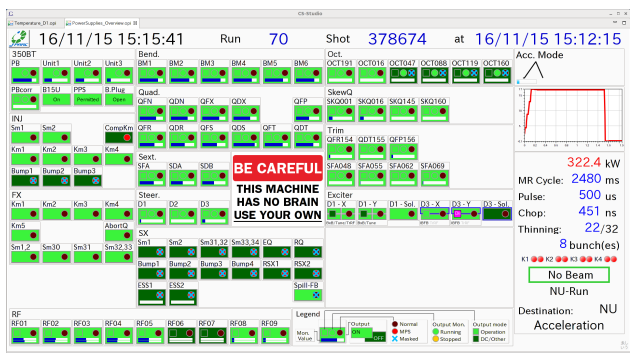
A two-tier, one-way gateway system is constructed, aiming only to read accelerator operation information from JLAN in realtime. It is not expected to manipulate any accelerator component from JLAN, regardless of whether being intentional or not. In order satisfy those requirements, each of those gateway is configured as following:

- CA Gateway #1 (located in control-DMZ):
  Provide accelerator operation information in read-only mode from the control LAN to CA Gateway # 2. An additional IP address is assigned for JLAN-intraDMZ using NAT function of the firewall. CA Gateway #1 is configured so that any I/O access to control-LAN is read-only. Connections between CA Gateway #1 and JLAN-intraDMZ is limited to CA Gateway #2, and only CA protocol is allowed. SSH access to CA Gateway #1 is limited only from the login-server.
- CA Gateway #2 (located in JLAN-intraDMZ):
  It provides read-only CA connections to clients in JLAN. CA Gateway #2 is also configured so that any I/O access to CA Gateway #1 is read-only. Connections between CA Gateway #2 and JLAN is limited to CA protocol. SSH access to CA Gateway #2 is limited only from the login-server.

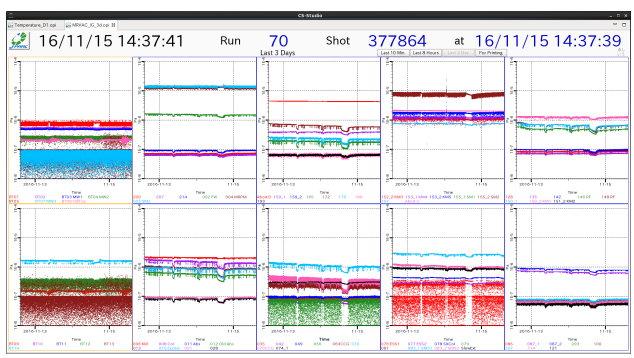
CA Gateway #2 is the sole server which is accessible for CA clients in JLAN. Therefore no CA client in JLAN is able to access equipment in control-LAN directly.

UTILIZING HIGHER LEVEL CONTROL APPLICATIONS IN JLAN

Historically, GUI applications in MR control system have been developed with EDM [3] and MEDM [4], which are GUI builders running on X Window System. There are more than 100 screen definition files used in MR operation, there-



(a) A screenshot of an OPI showing operation mode of MR and status of power supplies.



(b) A screenshot of an OPI showing time variation of pressure in MR beam pipe.

Figure 3: Example of OPIs actually used for MR operation.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP04

fore it was expected to be complicated for both developers and users to use EDM or MEDM in JLAN. Each user's PC in JLAN needs X Windows System installed on it to use those applications. In addition, screen definition files have to be distributed and installed on each user's PC.

In 2015 CS-Studio (CSS) [5] was introduced to MR for development of GUI application in its control system [6]. This solved those issues of distributing the latest of screen definition files to JLAN, installation to user's PC, and the prerequisite of X Window System.

CSS is a framework to build GUIs for EPICS-based large scale control system, available on Linux, macOS and Windows. Various kinds of modules are able to be cooperated in CSS, such as:

- BOY: GUI builder and runtime environment,
- Data Browser: real-time trend graph as well as historical data from archive system,
- Alarm system,
- Archive system.

CSS is configured appropriately for JLAN and distributed to JLAN in a zip-file. Screen definition files are automatically downloaded from the web server on the fly. Thus the latest version of screen definitions, identical to those used in operation of MR, are always available. Figure 3 shows examples of CSS display which are used in MR operation.

SUMMARY OF GATEWAY SYSTEM OPERATION AND FUTURE PROSPECTS

The gateway system started its service in April 2018. Two commercial tiny fanless servers, PiNON Sabataro ® Type-P, were deployed for the gateway system as shown in Fig. 4. Its specification is shown in Table 1.

Figure 4: Two server PCs running the gateway system.

Figure 5 shows operation status of Gateway #2, from the start of its service in April 2018 to the summer shutdown of the accelerator in July 2018. It will be summarized as following:

- There were constant access to the gateway from 10 – 20 clients, 8 out of them are archive system which records status of the gateway.

Table 1: Specifications of PiNON Sabataro® Type-P

Processor	Celeron J1900 (2–2.42 GHz), 4 cores
RAM	8 GB (1333 MHz DDR3L SO-DIMM × 1)
Storage	128 GB (mSATA SSD × 1)
Network	GbE × 2
Display I/F	HDMI
USB ports	USB 2.0 × 2
Dimension	W80.6 mm × D110.6 mm × H34.4 mm
TDP	max 15 W
Power Supply	DC 12 V (AC/DC Adapter)

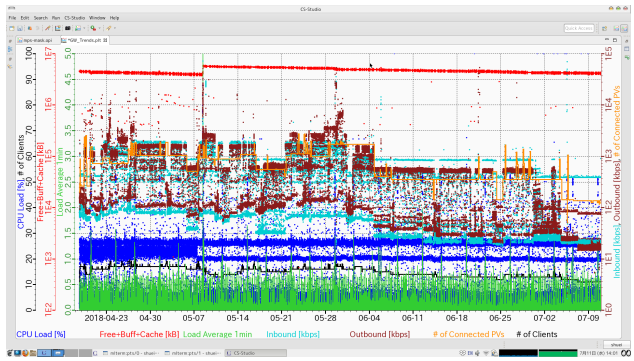


Figure 5: History of load of the gateway system for three months.

- 100 - 3000 control points have been accessed.
- Typical data rate of 1 – 10 Mbps went through the gateway.
- There are still plenty of root in CPU and memory; high CPU load of 25% was caused by anti-virus software scanning the gateway.

The gateway system have been operating stably so far. It is expected that gateway system is more utilized and the number of users of the gateway system will be increased.

REFERENCES

[1] EPICS - Experimental Physics and Industrial Control System, https://epics-controls.org

[2] Channel Access Gateway, http://epics.anl.gov/extensions/gateway

[3] EDM - Extensible Display Manager, http://ics-web.sns.ornl.gov/edm/edmUserManual/

[4] MEDM - Motif Editor and Display Manager, http://www.aps.anl.gov/epics/extensions/medm/index.php

[5] CSS – Control System Studio, http://controlsystemstudio.org/

[6] S. Yamada *et al.*, "Deployment of Control System Studio at J-PARC Main Ring", in *Proceedings of the 8th Annual Meeting of Particle Accelerator Society of Japan*, WEP103, pp. 543, 2011.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP05

DEVELOPMENT AND CURRENT STATUS OF KURAMA-II*

M. Tanigaki†
Institute for Integrated Radiation and Nuclear Science, Kyoto University,
590-0494 Kumatori, Osaka, Japan

Abstract

KURAMA-II, a successor of a carborne gamma-ray survey system named KURAMA (Kyoto University RAdiation MApping system), has been developed and applied to various activities related to the nuclear accident at TEPCO Fukushima Daiichi Nuclear Power Plant in 2011. KURAMA-II has established its position as an effective method for the radiation monitoring method in environment. The development of KURAMA-II is still on the way to extend its application areas such as the trial to port the system to a single-board computer or the development of a new cloud service. In this paper, the current status of KURAMA-II on its developments and applications along with some results from its applications are introduced.

INTRODUCTION

The magnitude-9 earthquake in eastern Japan and the following massive tsunami caused a serious nuclear disaster for the Fukushima Daiichi nuclear power plant. Serious contamination by radioactive isotopes was caused in Fukushima and surrounding prefectures, but the existing radiation-monitoring schemes were incompetent for this situation due to damage and chaos caused by the earthquake.

KURAMA [1] was developed to overcome difficulties in radiation surveys and to establish air dose-rate maps during and after the incident. The design of KURAMA was intended to enable a large number of in-vehicle apparatuses to be prepared within a short period of time by using consumer products. The in-vehicle part of KURAMA consists of a conventional radiation survey meter, a laptop PC, a USB-type GPS dongle, and a 3G pocket wi-fi router. The data-sharing scheme based on Dropbox, a cloud technology, has enabled high flexibility and scalability in the configuration of data-processing hubs or monitoring cars. KURAMA succeeded in the simultaneous radiation monitoring extended over a wide area such as Fukushima prefecture and the eastern Japan, in contrast to other conventional carborne survey systems lacking of scalability.

As the situation became stabilized, the main interest in measurements moved to the long-term (several tens of years) monitoring of radiation from radioactive materials remaining in the environment. KURAMA-II [2] was developed for such purpose by introducing the concept of continuous monitoring from vehicles moving around residential areas, such as local buses and postal motorcycles. The ruggedness, stability, autonomous operation and compactness were well taken into consideration in its design, and an additional measurement capability of pulse-height information along with location data was also introduced. KURAMA-II has been successfully introduced to the continuous monitoring in residential areas and other monitoring activities.

In this paper, the outline and the current status of KURAMA-II along with some results from its applications are introduced.

KURAMA-II

System Outline

The system outline of KURAMA-II is shown in Fig. 1. The in-vehicle part is based on CompactRIO to obtain sufficient ruggedness, stability, compactness and autonomous operation feature. The radiation-detection part of KURAMA-II is the C12137 series by Hamamatsu Photonics [3], a CsI(Tl) detector series characterized by its compactness, high efficiency, direct ADC output and USB bus power operation. The size of CsI(Tl) scintillator varies depending on the usage, typically 3.4 cc for conventional carborne surveys. The ambient air dose rate, $H^*(10)$, is calculated from the pulse height spectrum obtained for each measurement point by using G(E) function method [4–6]. Since the energy dependence of detector efficiency is properly compensated by the G(E) function, more reliable results are expected in the case of environmental radiation that is dominated by γ-rays scattered by air, soil, or buildings etc. All components of the in-vehicle part are placed in a small tool box (34.5 cm × 17.5 cm × 19.5 cm) made of wood covered with thin aluminum sheet for the better handling.

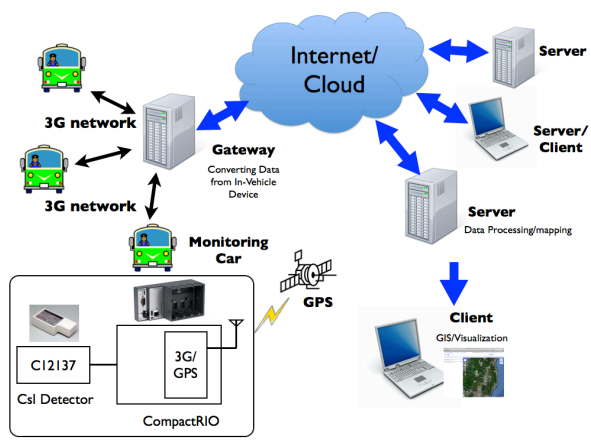


Figure 1: The system outline of KURAMA-II.

* Work partly supported by the distribution-mapping projects organized by Nuclear Regulation Authority, Japan, JSPS KAKENHI Grant Number JP16K00543, and a grant from Agriculture, Forestry and Fisheries Research Council, MAFF, Japan.
† tanigaki@rri.kyoto-u.ac.jp

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP05

Data Communication in KURAMA-II

The file transfer protocol used in KURAMA-II has been designed to send data without any loss under the poor coverage of the mobile network expected in emergency situations, as well as to comply to the standard protocols that are widely used in today's networks, such as Web Services.

In this protocol, two timestamped files, a text file for the air dose rates and a 32-bit binary file for the pulse-height spectra are separately produced for every three measurement points as data files. Generated data files are transferred to a remote "gateway server" by the POST method. All communications between in-vehicle units and a remote "gateway server" are based on RESTful API. Unsent files are archived inside an in-vehicle unit as a single zip file and wait for the next available network connection.

Once data files are received by the "gateway server", two text files on Dropbox, one is for the air dose rate and the other is for the pulse height spectra, are updated. These updated files are shared with remote servers via Dropbox, as was done in KURAMA.

A New Cloud Service for KURAMA-II

Now we are moving to a new cloud data storage service based on ownCloud [7] especially for KURAMA-II, allowing the full control of the cloud system to the users for the immediate recovery and easier operation of the system under emergency situations. The scheme used for the new cloud service is shown in Fig. 2. Two virtual servers, a RESTful API server with the database for received data and a cloud server for ownCloud, are implemented on a FreeBSD physical server. All the data sent from in-vehicle units are stored in a database on an API virtual server as well as data files on a cloud virtual server. The uploaded data is shared with remote servers via ownCloud. Other methods like WebDAV or SFTP may be used for the communications with other existing data sharing scheme.

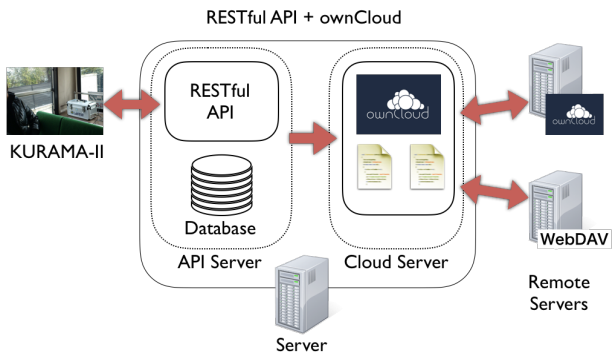


Figure 2: The outline of a new cloud service for KURAMA-II.

This new cloud service shows high reliability and flexibility for the use with several in-vehicle units on local buses and several units for walk survey in Fukushima prefecture, but the file synchronization with remote servers is slow due to the lack of delta synchronization in ownCloud. We expect improved file synchronization speed with the next release of ownCloud that is expected to implement delta synchronization [8].

APPLICATIONS OF KURAMA-II

Air Dose Rate Estimation in Residential Areas in Fukushima

A huge amount of data have been compiled for the air dose rate in eastern Japan by extending and continuing radiation monitoring activities by KURAMA-II [9, 10], and the characteristics of air dose rate in residential areas in eastern Japan has been studied. Saito and Andoh compared results from carborne surveys with those from walk surveys by KURAMA-II in residential areas (Fig. 3) and from the measurements in undisturbed flat fields [11, 12]. The air dose rates measured by walk surveys in residential areas are always between those from the car-borne surveys and the undisturbed field measurements (Fig. 4). Additionally, the measured air dose rates decrease more rapidly than expected from the physical decay, indicating the acceleration of decrease caused by various human activities.

Figure 3: Walk survey by KURAMA-II [12].

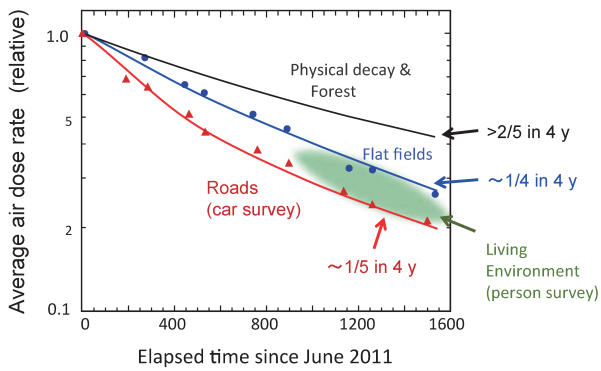


Figure 4: Temporal change of the average air dose rate for different conditions [11]. Air dose rates measured by walk survey are always between those measured by carborne survey and those by the undisturbed field measurements.

Recovery of Farmlands near Fukushima Daiichi Nuclear Power Plant

A differential measurement technique has been developed for the evaluation of soil contamination [13, 14] based on

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP05

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

a high flexibility in hardware configuration of KURAMA-II. In this technique, the contribution from surroundings measured by a nondirectional detector is subtracted from the intensity measured by a detector collimated towards the ground surface to obtain only the contribution from the radioactivity on the ground.

For the practical application of this technique, a three-year project starting from the fiscal year of 2018 has been approved by Ministry of Agriculture, Forestry and Fisheries, Japan, to build a “robot" for the recovery of contaminated farmlands (Fig. 5). This “robot" will be a faming tractor equipped with KURAMA-II for radioactivity, an optical spectrum sensing system to measure the fertility and chemical property of soil, and the high precision guidance system for the operation of tractor in farmlands. All the data will be shared over the Cloud system developed for KURAMA-II and simultaneous visualization of radioactivities and fertility of farmlands will be performed on the smart phones and tablet-type devices to help farmers to recover their farmlands.

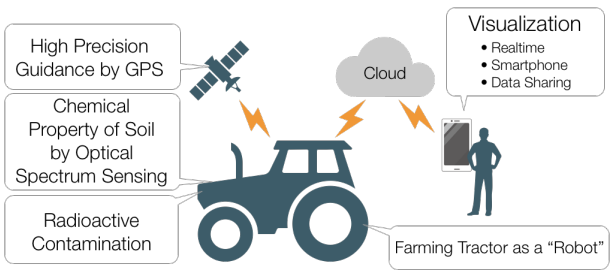


Figure 5: A conceptual scheme of a “robot" for the recovery of farmlands.

Development of a Single-board KURAMA-II

KURAMA-II is now porting to a single-board computer, to extend its applications to extreme conditions including the disposable usage for the immediate measurement in areas extremely contaminated by radioactivity. In such case, many single-board KURAMA-IIs will be deployed from unmanned aerial vehicles to the target area and continue measurements while their power last.

Spresense from Sony [15] is chosen as the platform of KURAMA-II because of its integrated GPS, audio processing capability applicable to pulse height analysis, and a powerful multi-core micro controller. C12137 is used as a detector in the beginning for feasibility studies, and a CsI(Tl) detector and a shaping amplifier will be implemented and the audio input will be used for the pulse height analysis.

ACKNOWLEDGEMENTS

Nuclear Power Safety Division of the Fukushima prefectural government gives continuous supports for developments and field tests of KURAMA-II. Gratitude is also expressed to Dr. Saito, Dr. Tsuda, Dr. Yoshida, Dr. Takemiya and other JAEA staff members involved in KURAMA activities for the discussions regarding KURAMA-II. This work has been supported by “Monotsukuri Fukkou Shien Project”, a support program for the recovery from the Great East Japan Earthquake by National Instruments Japan Corporation. Dr. H. Hanai at S2 factory made contributions for a new cloud service for KURAMA-II and for a single-board KURAMA-II.

Finally, a special gratitude is expressed to Mr. and Mrs. Takahashi and the staff members at “Matsushimaya Ryokan”, an inn at Iizaka hot spring in Fukushima city, for their hospitality and support regardless of their severe circumstances due to the earthquake and the following nuclear accident.

REFERENCES

[1] M. Tanigaki, R. Okumura, K. Takamiya, *et al.*, “Development of a car-borne γ-ray survey system, KURAMA”, *Nucl. Instrum. Meth.* A, vol. 726, pp. 162–168, 2013. doi:10.1016/j.nima.2013.05.059

[2] M. Tanigaki, R. Okumura, K. Takamiya, *et al.*, “Development of KURAMA-II and its operation in Fukushima”, *Nucl. Instrum. Meth.* A, vol. 781, pp. 57–64, 2015. doi:10.1016/j.nima.2015.01.086

[3] Hamamatsu Photonics Corporation, http://www.hamamatsu.com/jp/en/C12137.html

[4] S. Moriuchi and I. Miyanaga, *Health Phys.*, vol. 12, no. 4, pp. 541–551, 1966.

[5] S. Moriuchi, “A Method for Dose Evaluation by Spectrum-Dose Conversion Operator and the Determination of the Operator”, JAERI, 1209, 1970.

[6] S. Tsuda, T. Yoshida, M. Tsutsumi, *et al.*, *J. Environ. Radioactiv.*, vol. 139, pp. 260–265, 2015.

[7] ownCloud, https://owncloud.org

[8] ownClowd, https://owncloud.org/news/welcome-delta-sync-for-owncloud/

[9] Extension site of Distribution Map od Radiation Dose, Nuclear Regulation Authority, Japan, https://ramap.jmc.or.jp/map/eng

[10] Results from carborne survey in Fukushima prefecture, Fukushima prefectural government, http://www.pref.fukushima.lg.jp/site/portal/ps-soukou.html (in Japanese)

[11] K. Saito, H. Yamamoto, S. Mikami, *et al.*, *Global Environ. Res.*, vol. 20, pp. 15–22, 2016.

[12] M. Andoh, H. Yamamoto, T. Kanno, *et al.*, *J. Environ. Radioact.*, vol. 190/191, pp. 111–121, 2018.

[13] M. Yuda, “Technical Information for the Assistance to Radiation Problems”, Fukushima Agricultural Technology Centre, (2014), http://www4.pref.fukushima.jp/nougyou-centre/kenkyuseika/h26_radiologic/h26_radiologic_35_kaju_orchard_mapping.pdf (in Japanese).

[14] M. Tanigaki, R. Okumura, K. Takamiya, *et al.*, “Development and Current Status of a Carborne gamma-ray Survey System, KURAMA-II ”, in *Proc. IRPA'14*, Cape Town, South Africa, 9—13 May 2016, pp. 1818-1825.

[15] Sony Corporation, https://developer.sony.com/develop/spresense/

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP06

RELIABILITY IMPROVEMENT FOR THE INSERTION DEVICE CONTROL IN THE TPS

C. Y. Wu, C. Y. Liao, Y. S. Cheng, J. Chen, K. H. Hu, K. T. Hsu
NSRRC, Hsinchu 30076, Taiwan

Abstract

Insertion devices (ID) are essential components in third-generation synchrotron light sources, which can produce highly-brilliant, collimated and quasi-monochromatic radiation over a broad energy range for experiments. Reliable operation of the insertion devices is important to users of beamlines. The most unpredictable fault is due to a soft error in optical absolute encoders due to radiation. There are several solutions to avoid such faults, e.g. by increasing the distance of the encoder from the beam, by a lead shield cover and finally by adopting an auxiliary position sensing device to help recovery from a fault. Efforts to improve operational reliability of the TPS ID controls will be discussed.

INTRODUCTION

The TPS includes one EPU46, two EPU48s and seven IUs (In-Vacuum Undulator) which are installed in seven straight sections to meet experimental requirements in phase I beamlines of the TPS project [1-4]. The ID control system was developed including gap/phase motion, protection system (hardware and software) and GUI development. All control systems for insertion devices are done by the NSRRC control team for economic reasons and delivery of a similar ID control environment is the goal.

The EPU48 includes six axes servo motors. Two servo motors control the gap, one on the upper girder and one on the lower girder. The other four servo motors control the phase, two on the upper and two on the lower girder. Each axis servo motor is connected to a rotary absolute encoder. In addition, each servo axis is tracked with a TR absolute linear encoder with 0.1 μm resolution, providing direct gap sensing and frame reference to eliminate effects of backlash. The EPICS IOC performs a software tilt calculation based on the linear encoder feedback in addition to the tilt sensors. With gap / phase change, the corrector magnets for IDs require very sensitive power supply controls to maintain very stringent beam stability requirements.

Our control plan for the phase-I insertion devices is based on the standard TPS cPCI EPICS IOC. The motion controller is based upon the Galil DMC-40x0 series Ethernet based motion controllers [5]. The controller is a full-featured motion controller packaged with multi-axis drives in a compact, metal enclosure. It controls the motors based on commands via Ethernet and receives commands from the EPICS IOC to handle motor motion and to read encoder positions, limit switches, position error and other states for monitor and software protection.

The motion controller is suitable for servo motors (EPU46, EPU48) and stepper motors (IU22). Closed loop gap adjustment is needed for varying phases of the EPU48 and EPU46. It can be copied with changing forces between upper and lower magnetic arrays. All motion axes include a synchronous serial interface (SSI), where the optical encoder connects to the motion controller directly. Each motion axis is accompanied with limit switches for over-travel protection. Synchronization of gap motion axes is essential to prevent tilt of the beam.

The hardware configuration for the TPS ID control is shown in Fig. 1 including cPCI EPICS IOC, 128 bits DI/DO module, ADC/DAC IP (Industry Pack) modules, motion controller, temperature monitoring solution and RS232/422/485 based devices of the insertion frame. High precision power supplies are used to control corrector magnets from feed forward look-up tables. Current designs include also the control interface for the beamline, e.g. the IU22 controls include an ion pump and ion gauge interface.

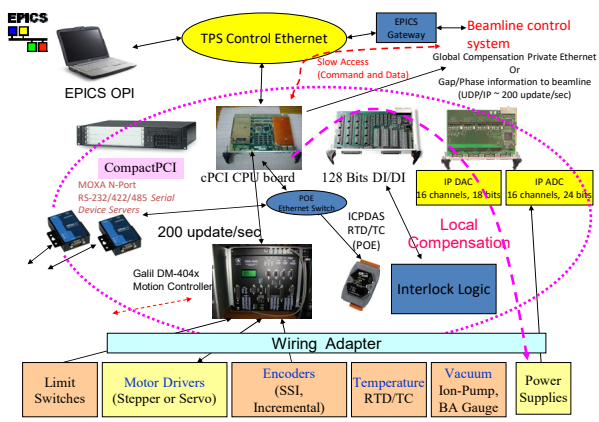


Figure 1: Basic hardware configuration for TPS insertion devices in Phase-I.

RADIATION EFFECT ON OPTICAL ABSOLUTE ENCODERS

Particle accelerators produce high energy protons and electrons, and the secondary particles produced by their interactions produce significant radiation damage on most semiconductor electronics components.

A single event effect (SEE) may be an electrical disturbance that disrupts the normal operation of a circuit. It is caused by the passage of a single ion through or near a sensitive node in a circuit. Single event effects can be either destructive or non- destructive.

Single-event upsets (SEU) or transient radiation effects in electronics are state changes of memory or register bits caused by a single ion interacting with the chip. They do not cause lasting damage to the device, but may cause lasting problems to a system which cannot recover from

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP06

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

such an error. The error in device output or operation caused as a result of the SEU is called a soft error.

In very sensitive devices, a single ion can cause a multiple-bit upset (MBU) in several adjacent memory cells. SEUs can become Single-event functional interrupts (SEFI) when they upset control circuits, such as state machines, placing the device into an undefined state, a test mode, or a halt, which would then need a reset or a power cycle to recover.

TPS ID operation faces problems with absolute optical encoders fitted to the EPU48 and EPU46. The position of the encoders is close to beam level. The SSI encoders easily suffer soft errors by radiation when a beam dump or beam trip occurs. The encoders fail during a beam dump or beam trip as a result of a shower of particles disrupting the internal electronics.

There are three soft errors for SSI encoders: (1) the encoder generates erroneous position data and then auto recovers as shown in Fig. 2; (2) the encoder generates a wide range of position data that keeps jumping; (3) the encoder always sends a same serial data stream as shown in Fig. 3. Case (2) and (3) require a power cycle to recover.

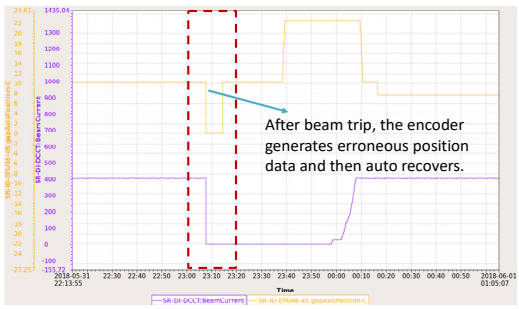


Figure 2: Case (1): the encoder of the EPU46 generates erroneous position data and then recovers.

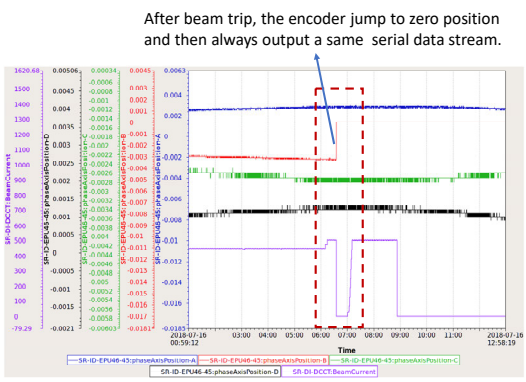


Figure 3: Case (3), the encoder stops sending a serial data stream.

Several safety protections are performed by the motion controller, which are over-travel limit switches, torque limits, stall, and close-loop position error limits. A limit threshold check of the encoder read-back is done by the EPICS IOC db-scan. According to encoder values, upper or lower beam tilt limits may also be monitored by the EPICS IOC. The encoder value is used to servo control and as input to many protection tasks. When the encoder stops sending an authentic serial data stream, the motion control will suffer unpredictable faults and motion control functions are invalid. In serious cases, it could lead to damage of the mechanical structure of the ID.

To avoid encoder damage and failure due to radiation, some 10 mm thick lead boxes are added around the SSI encoders of the EPU48 and EPU46 as shown in Fig. 4. Encoder soft errors are still occurring but the failure rate of encoders is reduced.

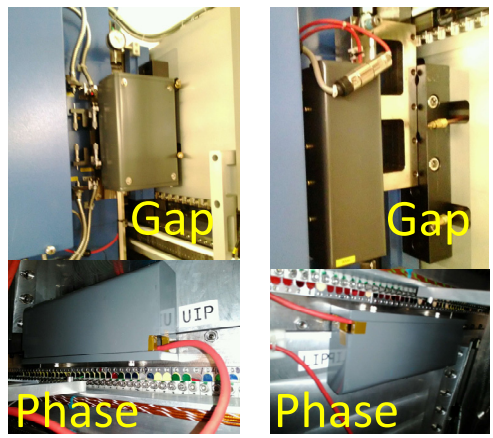


Figure 4: Lead boxes of more than 10 mm thickness are added around the SSI encoders of the EPU48 and EPU46.

DETECTION OF ENCODER ERRORS

The method of detection encoder error is to adopt auxiliary position sensing devices. A potentiometer is used at an auxiliary position sensing the device motion because it has no complex active component inside.

Potentiometers are installed to mirror the absolute encoders on the IDs to provide a robust gap and phase measurement for the protector system as shown in Fig. 5, even if the motion control encoders fail.

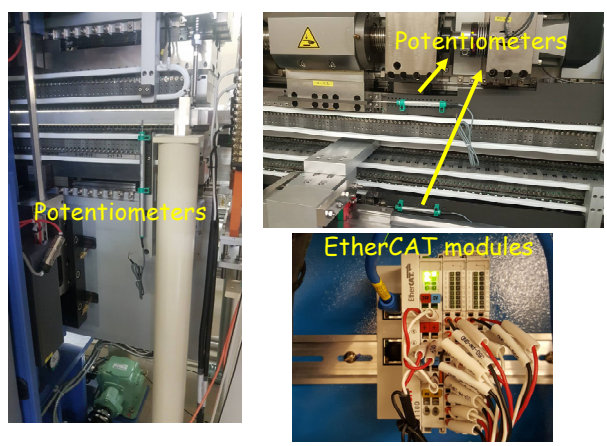


Figure 5: Potentiometers are installed to mirror the absolute encoders on the EPU48 and EPU46.

The reading module for the potentiometers is based on the EtherCAT device [6]. The hardware configuration for potentiometers of the EPU48 and EPU46 is shown in

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP06

Fig. 6. The Beckhoff EL3255 (potentiometer reader) module and potentiometer (GEFRAN, PZ-12-S-200 for Gap, PZ-12-S-075 for Phase) are installed at the ID [7]. After cross-calibration of potentiometer with the SSI encoders, the auxiliary absolute position can be used to crosscheck the absolute SSI encoder health for malfunction protection purpose.

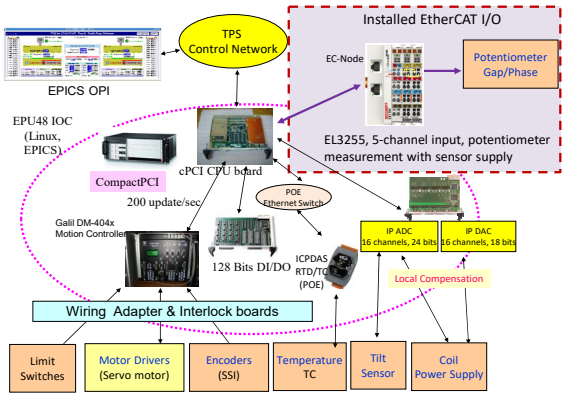


Figure 6: The hardware configuration for potentiometer of the EPU48 and EPU46.

The graphical user interface (GUI) is implemented by using EPICS EDM and Fig. 7 shows the main page for the EPU46. The right side shows encoder status which are bypass, all pass and gap or phase of encoder no update. If the encoder value has not been updated, the protection program can halt the motion of any axis by executing the abort motion command within a few milliseconds. The trip state can only be reenergized if all of the encoder errors are cleared or overridden. The flow chart for encoder error detection and aborting ID motion is shown in Fig. 8. A protection process is developed to compare the position from two sensors, if the difference is too large, the abort motion command will be sent to the motion controller to stop the motor driver.

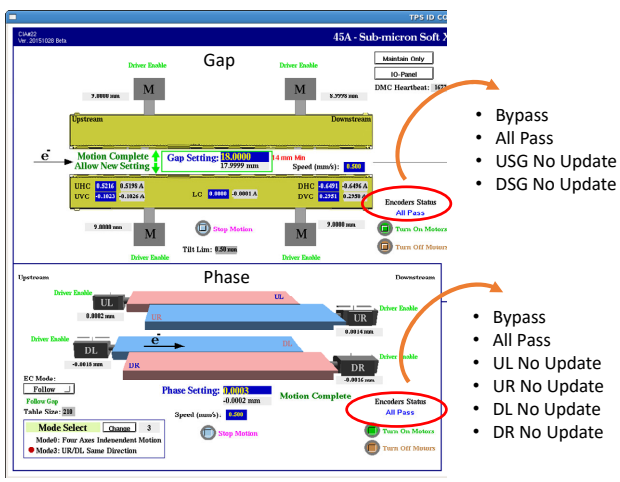


Figure 7: GUI of the EPU48.

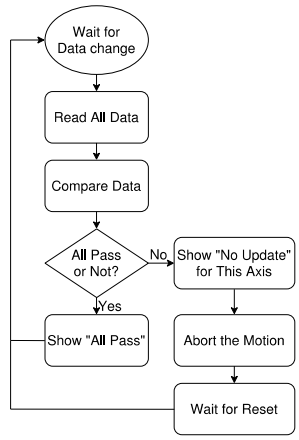


Figure 8: Flow chart for encoder error detection and abortion of ID motion.

CURRENT STATUS

Basic functionalities for the detection of encoder errors has been implemented and tested. EtherCAT EPICS IOC and GUI are developed for the EPU48 and EPU46. The failure rates of encoders are reduced after the encoders are covered by lead shielding. Preliminary tests show its efficacy for early detection of encoder errors and termination of motion.

Moving the encoders away from the electron beam axis to overcome radiation problems will be done during TPS phase II IDs for the EPU66 and EPU168. Potentiometers will also be installed to crosscheck the absolute encoders on the TPS phase II EPU ID to provide a robust gap and phase measurement for the protection system.

We have yet to establish whether we will see long term radiation damage of these encoders. Some of them have been in the ring since 2015. Next, we will develop a program to automatically recover absolute encoder soft errors.

REFERENCES

[1] C. H. Chang *et al.*, "Progress in Insertion Devices for TPS in Phase I", in *Proc. IPAC'11*, San Sebastian, Spain, Sept. 2011, paper THPC176.

[2] C. Y. Wu *et al.*, "Insertion devices control plans for the Taiwan Photon Source", in *Proc. IPAC'12*, May 2012, New Orleans, LA, USA, paper THPPR011.

[3] C. Y. Wu *et al.*, "Control System of In-vacuum Undulator in Taiwan Photon Source", in *Proc. IPAC'13*, Shanghai, China, May 2013, paper WEPWA053.

[4] C. Y. Wu *et al.*, "Status of the TPS Insertion Devices Controls", in *Proc. ICALEPCS'13*, Oct. 2013, San Francisco, CA, USA, paper THPPC063.

[5] Galil motion control: `http://www.galilmc.com`

[6] EtherCAT Technical Introduction and Overview, `www.controldesign.com/assets/Media/MediaManager/EtherCAT_Introduction.pdf`

[7] Beckhoff Automation, `www.beckhoff.com`

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP07

LONG-TERM STABILITY OBSERVED BY ELECTRON BPM AND PHOTON BPM FOR TAIWAN PHOTON SOURCE

P. C. Chiu, K. H. Hu, C. H. Huang, K. T. Hsu
NSRRC, Hsinchu 30076, Taiwan

Abstract

TPS is 3-GeV synchrotron light source which has opened for public users since September 2016 and now offers 400 mA top-up mode operation. The requirements of the long term orbit stability and orbit reproducibility after beam trip have been gradually more and more stringent and become a challenge from users' request. Furthermore, the thermal effect would be expected to be worsen after 500 mA top-up operation which should deteriorate the orbit drift. The report investigates the long-term orbit stability observed from electron BPM and X-ray BPM and also evaluates the possibility of the local XBPM feedback to improve photon beam stability.

INTRODUCTION

The TPS requires beam position stability of less than 10% of the beam size to provide advanced experimental capabilities at the 3rd generation light source. Therefore, FOFB and RF frequency compensation have been adopted to stabilize the electron orbit [1][2]. Besides, to monitor the position and stability of photon beams, two-blade type X-ray beam position monitors (XBPMs) are installed in beamline frontends and beamlines[3][4]. It is observed that the thermal effect would cause the mid-term orbit disturbance at the first 30 minutes after the beginning of beam stored and long-term slowly drift for the following 4~5 hours before it achieves the equilibrium, especially in the vertical plane. Besides, there are also obvious daily position change along with temperature variations and periodic 4-minutes variation consistent with injection cycle. Furthermore, insertion device (ID) gap/phase change is also significantly affect position stability where it is partly caused by deformation and resulted in BPM mechanics displacement and partly still due to thermal effect. The position drift/fluctuation seemed to be able to be controlled below several microns or even sub-micron in electron BPM. However, the errors would be amplified several times observed at the end of beamline XBPM. This report summarized the long-term orbit drift observed by electron BPM and photon XBPM. The preliminary local XBPM feedback test is also proposed to minimize the errors.

PHOTON BEAM POSITION MONITOR LAYOUT AND ELECTRONICS

There are seven beamline open to users in TPS now. For each beamline, there are different types of X-ray or photon beam position monitors are used to detect the synchrotron radiation. The blade-type X-ray BPMs (XBPM) [3] is standard equipment installed at each front-end; quadrant PIN photodiode BPMs (QBPMs) [4] are adopted by few experimental end station. The layout of front-end instrumentation is shown as Fig. 1. XBPM1 is completed calibration and observed reliable for a while. However, the calibration of XBPM2 is not yet completed and it was observed that the horizontal and vertical readings of XBPM2 had serious coupling. Therefore, only XBPM1 is presented and included for feedback in this report. About acquisition electronics, three types of electronics had been used and evaluated. The first one uses the FMB Oxford F-460 to convert current to voltage and read the voltage with a NI-9220. The second type of electronics is a home-made device with a 0.5 Hz update rate. The third type is the commercial product and now our majority: Libera Photon which could provide different data flow for different purpose of analysis, including 10/25 Hz streaming, 5 kHz/578kHz waveform with trigger as well as post-mortem functionalities.

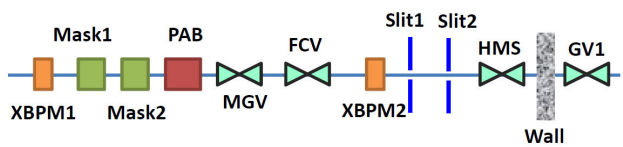


Figure1: Layout out of front-end instrumentation.

OBSERVATION OF ORBIT STABILITY BY BPM AND XBPM

Position Drifts After Beam Restored

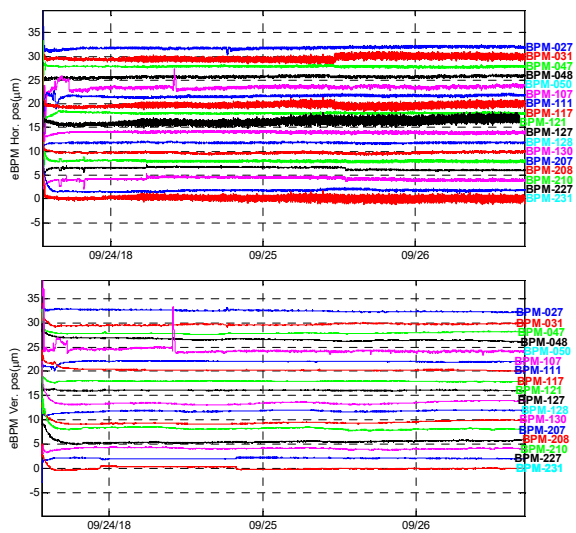


Figure 2: The upstream and downstream electron BPM reading nearby IDs for three days after beam restored.

It could be observed the upstream and downstream electron BPM nearby 7 IDs as Fig. 2 at first beginnings of beam stored, the position drift of some BPMs would be

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP07

up to around 2~4 μm for the following 2~3 hours before it achieves an equilibrium as shown in the detail of Fig. 3 for the vertical plane. Then it could be remained below submicron. However, this 2~4 μm drift of the first 3 hours would be amplified in the beamline observed by front-end XBPM as Fig. 4 by 3~20 times. XBPM at the FE 41 even could up to 80 μm. This kind of drift is especially apparent in the vertical plane.

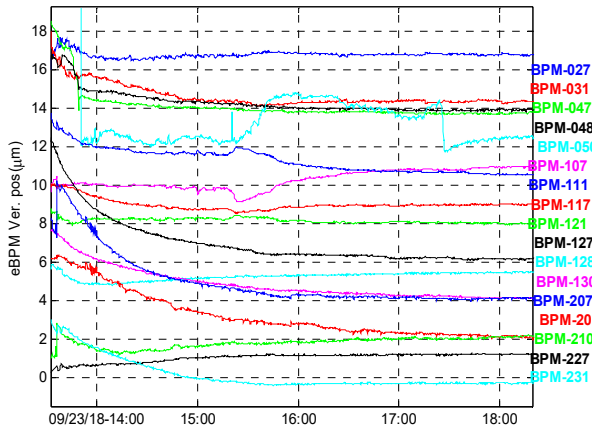


Figure 3: The vertical position change of BPM nearby IDs during the first 5 hours of beam stored.

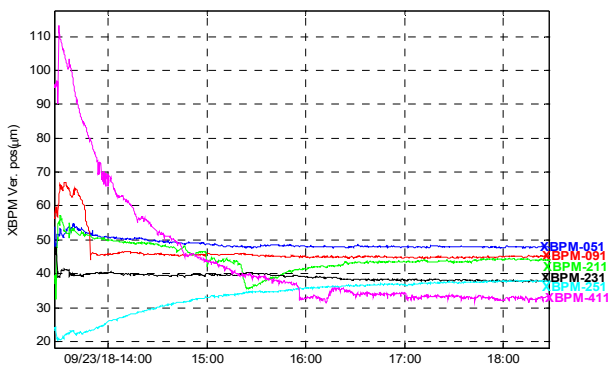


Figure 4: The vertical position change of XBPM during the first 5 hours of beam stored.

The patterns of these position distortions display quite similarly every time. But the range of the drift depends on different operational and environment condition. The major factor to affect the drift range is the time interval of the previous beam dump. Figure 5 shows the vertical position change of XBPM-41 on FE 41 and its adjacent upstream and downstream BPM during 2.5 hours from the beginnings of beam restored for Aug-21, Sep-10, Sep-23 this year. They are all operated on the standard User Mode. However, the amplitude of the XBPM and eBPM position drifts for Sep-23 (green line) is obvious 2-4 times larger than the other two. The cause is time interval (3 hours) between the beam stored is longer than the other two (0.5/1 hour). As beam dump is longer, the position drifts larger and it could take longer time to stabilize as well. Therefore, it is advised that the accelerator would be heated for a while before open to users. On the other hand, most of the beamlines actually also need to take a while for thermal equilibrium too. Besides, we'd make efforts to improve the machine reliability to decrease the accidental beam trip.

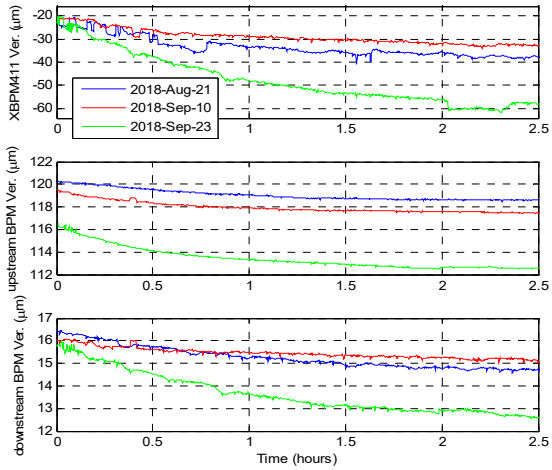


Figure 5: The vertical position change of XBPM-41 and the adjacent upstream and downstream BPM for 2.5 hours at initial beam stored.

Position Variation with Daily Change

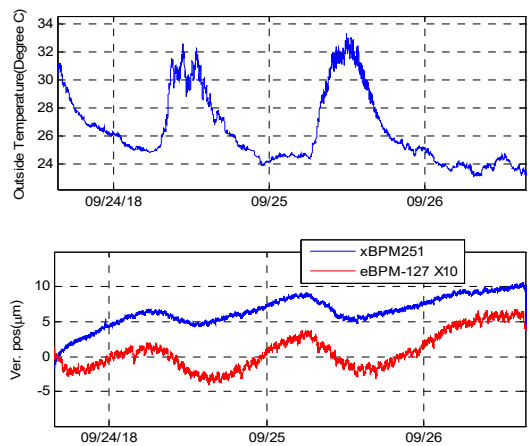


Figure 6: The upper plot is the temperature change for three days. The lower plot is the vertical position change of XBPM-251 and the adjacent upstream BPM.

It's also observed that there are some daily periodic variation for some eBPM and xBPM,. In FE-25, for example of the most visible case, the vertical position change of XBPM251 is highly correlated by it's upstream BPM, but amplified by a factor of ten as Fig. 6 shows. It could be inferred to be caused by daily temperature variation.

Position Fluctuation with Injection Cycle

Last, although in top-up mode, there is still effect appeared due to beam current change for some front-end XBPM. Figure 7(a) shows the vertical position of XBPM251 fluctuates around 5 um accompanying with beam current change during top-up while its upstream and

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP07

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

downstream BPM have no such variation. Furthermore, strangely, these phenomenon has not always existed. In Fig. 7(b), the same XBPM and eBPM, also in standard User Mode, had no such kind of behavior. The reason had not yet been clarified.

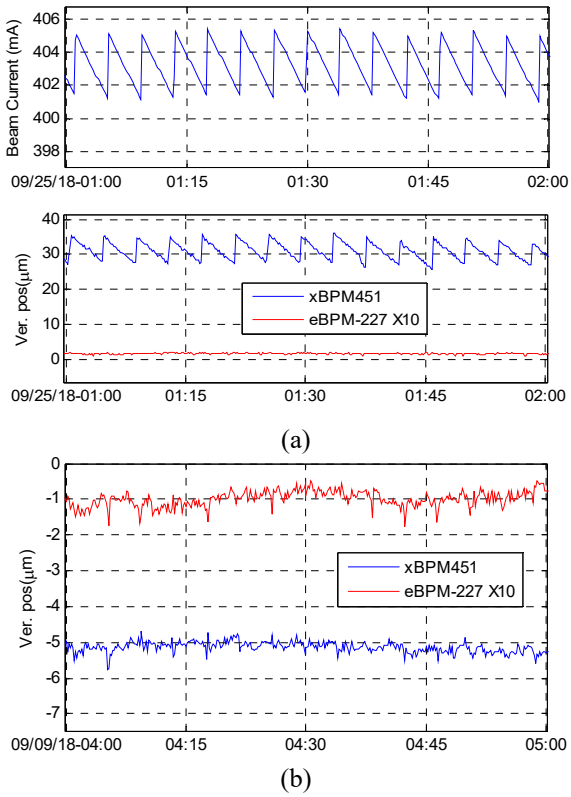


Figure 7: (a) The upper plot is the beam current. Beam injects every four minutes. The lower plot is the vertical position change of XBPM-451 and the adjacent upstream BPM on Sep.25 2018. (b) Data on Sep.09 2018

PRELIMINARY TEST OF LOCAL XBPM FEEDBACK

To further improve XBPM position stability, the local feedback to use XBPM is considered and proposed. Although the XBPM position is quite depended on ID gap/phase, the gap/phase has been not moving too often for most beamlines of TPS in practice. Therefore, the preliminary tests for XBPM local feedback is ongoing. Figure 8 shows the results of the experiments for FE21. It was 100 mA top-up mode. The upper plot is XBPM vertical position; the lower is its upperstream and downstream BPM. Feedback started at beginning and stopped at 16:20. The position fluctuation due to beam current change looked improve efficiently by feedback. Figure 9 shows the other FE09 case for applying XPM local feedback. It seemed the position drift improved somewhat as well. However, the reliability of XBPM is still an issue. Besides, sometimes FOFB and XBPM local feedback would be interfered with each others which would results the correction of corrector accumulated rapidly to saturation or even worsen, deteriorate the drift.

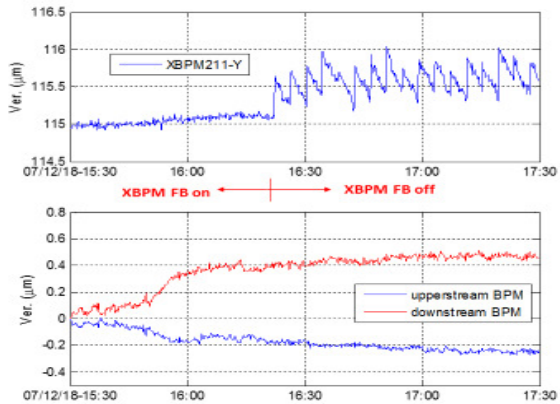


Figure 8: Local XBPM vertical feedback for FE21. Feedback stopped at 16:20. 100 mA at top-up mode.

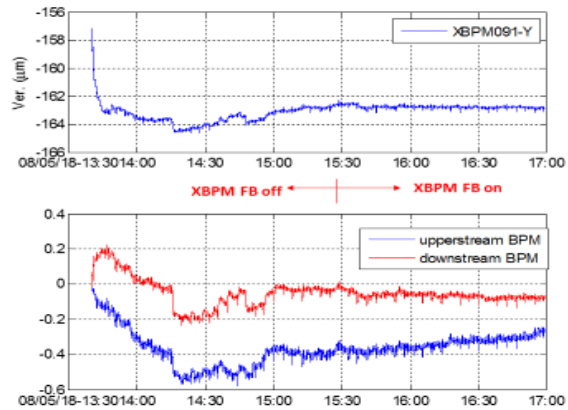


Figure 9: Local XBPM vertical feedback for FE09. Feedback started at 15:30. 300 mA top-up mode.

CONCLUSION

The long-term position stability observed from electron BPM and XBPM is presented. Three types of orbit variation are summarized. The major concern from users is the orbit drift would remain 1~3 hours at different ranges for different beamline after beam dump and restored. As beam dump longer, the drift become larger and longer. Therefore, the local XBPM feedback is tested to improve the drift. The result is promising while there are still some practical issues required to be resolved.

REFERENCES

[1] P. C. Chiu *et al.*, "Fast Orbit Scheme and Implementation for TPS", *in Proc. IPAC'13*, Shanghi, China, May 12-17 2013, paper TOUCB202, pp.1146-1148.

[2] P. C. Chiu *et al.*, "Orbit Correction with Path Length Compensation Based on RF Frequency Adjustments in TPS", *in Proc. IPAC'17*, Copenhagen, Denmark, May 14-19 2017, paper TUPAB103, pp.1553-1555.

[3] J. Morse *et al.*, "Diamond X-ray beam-position monitoring using signal readout at the synchrotron radiofrequency", *J. Synchrotron Rad.*, vol. 17, pp. 456-464, Jan. 2010. doi: 10.1007/S0909049510016554 7

[4] D. Shu *et al.*, "Development of an x-ray beam position monitor for TPS EPU beamline front ends", *Journal. Physics.: Conference Series,* vol. 425, 042003, Mar. 2013. doi: 10.1088/1742-6596/425/4/042003/

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP08

DESIGN AND IMPLEMENTATION OF STEPPER MOTOR CONTROL OF THE LINAC HIGH POWER RF SYSTEM BASED ON FPGA

R. Rujanakraikarn†, C. Dhammatong, W. Phacheerak
Synchrotron Light Research Institute, Nakhon Ratchasima, Thailand

Abstract

In this paper, the new motion control system that governs the position of high power attenuators and phase shifters in the linac's RF system at SLRI is described. The drive system, which was originally driven by a set of AC reversible motors, is replaced by a new set of stepper motors. The hardware selection and installation is presented in detail. The digital control circuits are designed in VHDL and implemented on a commercial Field Programmable Gate Array (FPGA) board. The main software part, implemented in MicroBlaze Microcontroller System (MCS), is coded in C to control the position of stepper motors relative to the DC voltage reference points of the hardware system. A LabVIEW GUI is designed to interface with the control system to provide reference points and display position values via RS-232 and PLC interfaces. This stepper motor control system can be used to effectively implement the phase and amplitude control system of the linac's RF signals in the future.

INTRODUCTION

Synchrotron Light Research Institute (SLRI) has been operating its synchrotron facility complex called SIAM Photon Source (SPS) to serve various research activities since December 2001. Its 40-MeV linac has been regularly operated only twice a day, with an approximate duration of one hour each, for electron injection. The linac consists of five different parts to accelerate the electron beam. These parts are pre-buncher 1, pre-buncher 2, buncher, a 20-MeV accelerating tube 1, and a 20-MeV accelerating tube 2 [1]. The phase and amplitude of a 2,856 MHz pulsed-RF are controlled manually by adjusting the high power phase shifters and attenuators, respectively. The mechanical parts of these high power elements consist of seven piston rods along the RF distribution waveguides depicted in a diagram in Figure 1. They are operated as controlling actuators for adjusting the phase shifters and amplitude attenuators. They had been driven by reversible AC induction motors since the beginning of the first light.

Even though the phase and amplitude of the RF signal along the waveguides can be adjusted, the actual numerical values of these RF parameters are not directly measured. During the normal operation, the position of each of the piston rods; i.e., the position of each of the phase shifters and amplitude attenuators, is referenced to the DC voltage across potentiometer in a motor drive circuit. This voltage is displayed at the electronic front panel in a control room as shown in Figure 2. Each day this set of values is recorded as RF operating point for beam injection and future reference. The operating point is occasionally changed if beam injection problem occurs or machine parameters are needed to be changed.

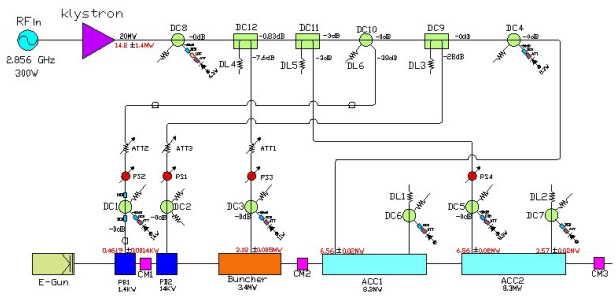


Figure 1: Linac RF distribution diagram.

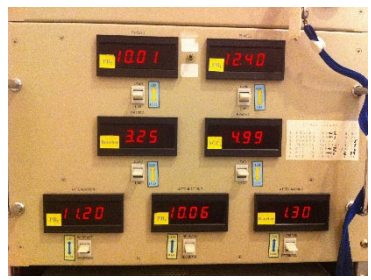


Figure 2: Electronic front panel display of the phase shifters and amplitude attenuators as DC voltages.

In this paper, new hardware selection and installation is described in the next section. Digital control circuit design of the system on FPGA board is also included. Software implementation and GUI are presented in the following section. System performance and conclusion are presented at the end of this paper.

HARDWARE

Stepper Motors, Electronic Drivers and Controllers

During annual shutdown in 2013 all reversible AC induction motors driving the high power phase shifters and amplitude attenuators were replaced by commercial stepper motors and their electronic controllers. The selected stepper motors and controllers are Oriental Motor's RK Series with Pulse Input Type and RKII Series with Built-in Controller Type, depending on the size and location of the phase shifters and amplitude attenuators in the linac system. The new motors were carefully chosen to meet mechanical characteristics and intended to provide better position control of the phase shifters and amplitude attenuators. Figure 3 shows an example of the motor replacement in the waveguide system. All electronic motor drivers and

† roengrut@slri.or.th

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP08

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

controllers including power supplies are installed in a standard equipment rack as shown in Figure 4.

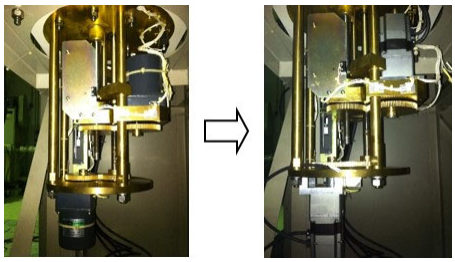


Figure 3: Stepper motor replacement for one phase shifter and one amplitude attenuator in the waveguide system.

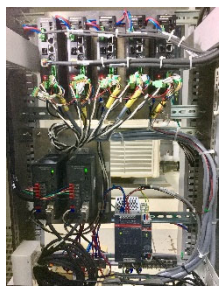


Figure 4: Installation of electronic drivers and controllers of stepper motors in a standard equipment rack.

In order to control the selected stepper motors, appropriate electrical signals must be generated to interface with their commercial electronic controllers. For the Pulse Input Type controllers [2], at least a motor direction signal and a pulse signal with specific frequency must be generated. For the Built-in Controller Type controllers [3], at least start and stop signals are needed. Other output signals from the motors, for example, Ready, Move, and Timing are considered for position feedback and motor rotation check purposes.

FPGA-Based Control Implementation

In this development, FPGA evaluation board is our choice for fast digital system implementation. Xilinx's ML605 Evaluation Kit [4] featuring a Virtex-6 XC6VLX240T FPGA is chosen to be a main controller of the design. Because of a number of signal interface wires to all seven stepper motors, several I/O pins of the FPGA board are needed. Xilinx's FMC XM105 Debug Card [5] is our choice for this purpose. Installation of the FPGA board, the debug card, and opto-isolator modules in a standard equipment box is shown in Figure 5.

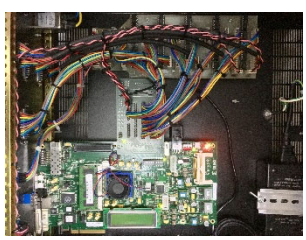


Figure 5: Virtex-6 ML605 FPGA board and FMC XM105 Debug Card daughter board connected to opto-isolator circuits.

In the FPGA, several digital circuit blocks, for example, frequency divider, frequency counter, comparator, multiplexer, and de-multiplexer, are designed using VHDL [6] in Xilinx's ISE 14.7 IDE. They are carefully implemented and interconnected in order to perform pulse processing tasks in this stepper motor control system. In this particular system, the frequency of the pulses for all seven stepper motors are fixed, depending on the types of motors and their controllers. Once the number of pulses and the motor IDs are specified by a LabVIEW GUI, appropriate signals are generated in this digital circuit part of the FPGA and sent to stepper motors through XM105 Debug Card. To complete this embedded control system, the main controller of the design is a MicroBlaze Micro Controller System (MCS). It is a complete standalone processor system intended for controller applications. Also, it is highly integrated which includes the MicroBlaze processor, local memory, as well as a tightly coupled IO module [7].

Several input and output signals are required to provide complete interface with all seven stepper motors. A complete list of these signals connecting MicroBlaze MCS, digital circuit blocks, and electronic drivers and controllers in this design is shown in Table 1.

Figure 6 shows the digital circuit part for determining motor direction and generating pulse signals with specific number of pulses for the motors with Pulse Input Type controllers. Motor start/enable and stop signals, together with specific number of pulses needed, for the motors with Built-in Controller Type controllers are generated in a similar fashion.

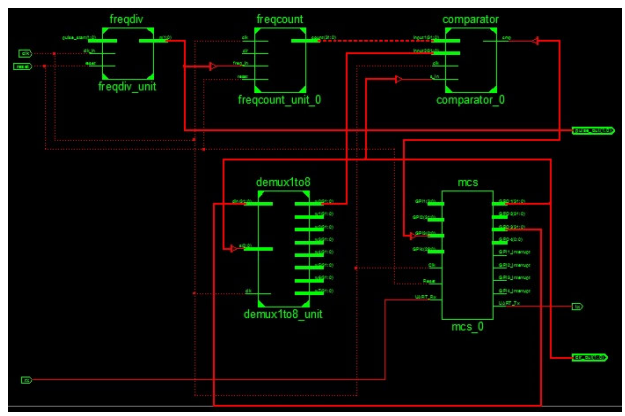


Figure 6: Digital circuit for generating direction and pulse signals.

Table 1: General Purpose Input and Output Signal

Port(bits)	Signal Description
GPI1(3,…,0)	Switches on ML605 board
GPI2(31,…,0)	32-bit value from Frequency Counter
GPI3(0)	Frequency Counter Done Flag
GPI4(29,…,0)	Motor Output Signals

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP08

GPO1(31,…,0)	32-bit Motor Control Signals (Motor0, Motor1, Motor2, Motor3, Motor4)
GPO2(31,…,0)	32-bit Motor Control Signals (Motor5, Motor6, clear data)
GPO3(31,…,0)	32-bit value of Number of Pulses
GPO4	Not used

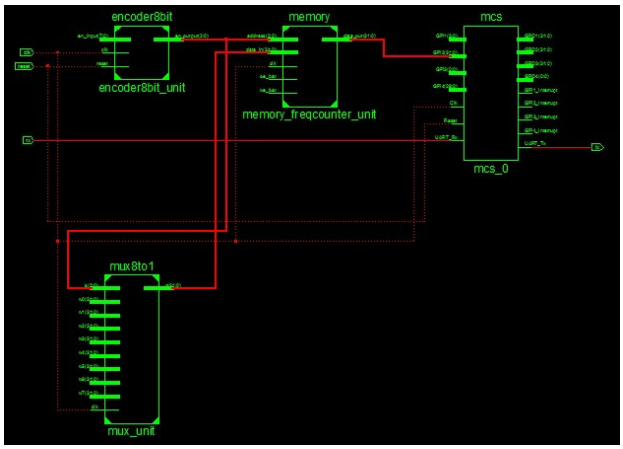


Figure 7: Digital circuit for receiving the number of pulses from GUI and storing in a memory for each motor.

After user determines motor position values through GUI, which are referred to the DC voltage across potentiometers, the calculated numbers of pulses are sent to the FPGA board and stored in the memory for further processing in the MicroBlaze MCS. Figure 7 shows the digital circuit part of this design.

SOFTWARE

The software of the system can be separated into two parts, LabVIEW GUI and MicroBlaze MCS software. The GUI part is responsible for receiving the position and direction setpoints, and displaying the current position of all phase shifters and amplitude attenuators. The second part on the MicroBlaze MCS is written in C. Its main purposes are to receive the setpoints from the GUI for further processing and to communicate with other digital circuit blocks in the FPGA logic fabric.

LabVIEW GUI

Figure 8 shows the main LabVIEW GUI of the system. User can determine the desired stepper motor position as the DC voltage through this GUI by either manually entering numeric setpoint, or clicking the arrow buttons to increase or decrease the desired position value of each of the phase shifters and amplitude attenuators. Each desired setpoint is compared to the current position of each motor. The rotation direction, either clockwise or counter-clockwise, is determined, and the number of needed pulses are calculated.

Figure 8: LabVIEW GUI of the system.

The pulse calculation of each of the motor is highly dependent on the hardware and software tests and parameter settings. A full-scale traveling distance of each motor from the minimum value to the maximum value of the DC voltage defines the maximum number of pulses. The full-scale traveling distance of all seven motors were carefully recorded. Several trials were performed to obtain the best full-scale reading. This gives us all necessary maximum number of pulses for all motors, thus providing the crucial information for pulse calculation. Once the calculation is finished, motor IDs, rotation direction, and number of pulses are sent to the FPGA board via UART. In addition, the current motor positions, which are displayed as numeric values on the screen, are obtained from PLC. Figure 9 shows program flowchart of the LabVIEW GUI.

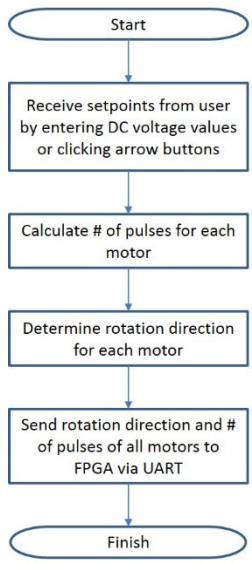


Figure 9: Program flowchart of LabVIEW GUI.

MicroBlaze MCS Software

The control software part on the MicroBlaze MCS is written in C using Xilinx's SDK. The main purpose of this implementation is to perform decision making on position control. The input and output signals of the MicroBlaze MCS are interfaced via the GPI and GPO ports, respectively, as presented in the Table 1. The detail of the C-program can be explained in the flowchart in Figure 10.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP08

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

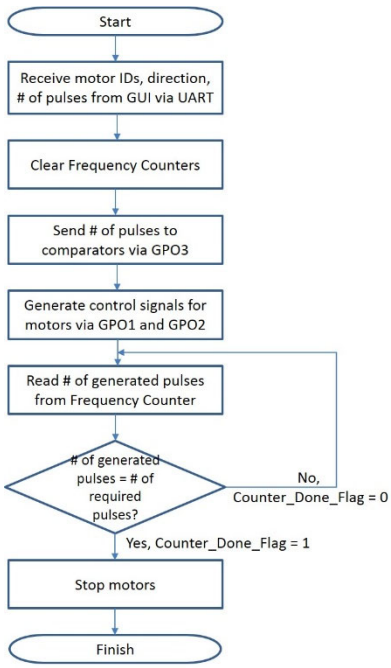


Figure 10: C-Program flowchart implemented on MicroBlaze MCS.

Figure 11: Position of phase shifters and amplitude attenuators after system implementation.

SYSTEM PERFORMANCE

The digital circuit blocks designed by VHDL were carefully tested to make sure they function correctly. The tests were performed both by using standard peripherals on the FPGA board and by connecting with multiple stepper motors and their electronic drivers and controllers in laboratory. By implementing several digital circuit blocks in the FPGA, the complication in designing the main software written in C on the MicroBlaze MCS can be considerably reduced.

The accuracy of the position control is the main important part of this stepper motor control system. Hardware and software tests and parameter settings play a very crucial role in the system implementation. In order to obtain the correct number of pulses required for any setpoint value, several trials were performed for all of the motors that were already installed in the linac's waveguide system. Figure 11 shows the result of final motor positions from the electronic front panel display after testing the new control system. We can see that, when compared to the reference inputs from LabVIEW GUI shown in Figure 8, the result is very satisfactory. The accuracy of the position control is within 0.4 % error.

CONCLUSION

The motion control system that governs the position of high power phase shifters and amplitude attenuators in the linac's RF system is implemented. The old drive system using reversible AC induction motors was replaced by a new set of commercial stepper motors with electronic drivers and controllers. Several digital control circuit blocks are designed in VHDL and implemented on a commercial FPGA board. The Xilinx's MicroBlaze MCS is the main controller to perform control decision based on the LabVIEW GUI commands and the digital signals from other digital circuit blocks in the FPGA fabric.

A number of tests were conducted to successfully achieve the desired result. The digital circuit blocks in the FPGA help achieve the appropriate control action and reduce the size and complication of the main C program on the MicroBlaze MCS. The design is good for our system because there are a number of signals between the FPGA board and electronic drivers and controllers of seven stepper motors in the system. The LabVIEW GUI is used to communicate with the FPGA via UART to provide reference points and display position values. The overall performance of the control system, both hardware design and software implementation, is very satisfactory. This stepper motor control system is also planned to be used with phase and amplitude detectors described in [1] in order to implement a linac's RF phase and amplitude control system in the future.

REFERENCES

[1] R. Rujanakraikarn, "FPGA-Based Pulsed-RF Phase and Amplitude Detector at SLRI", in *Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS2017)*, Barcelona, Spain. doi:10.18429/JACoW-ICALEPCS2017-TUPHA068

[2] 5-Phase Stepping Motor Unit RK Series, Operating Manual, http://www.orientalmotor.com

[3] New 5-Phase Stepping Motor and Driver Package RKII Series, User Manual, http://www.orientalmotor.com

[4] ML605 Hardware User Guide (UG534 v1.7), Jun. 2012, http://www.xilinx.com

[5] FMC XM105 Debug Card User Guide (UG537 v1.3), Jun. 2011, http://www.xilinx.com

[6] K. L. Short, *VHDL for Engineers*, Pearson Ed. 2009.

[7] LogiCORE IP MicroBlaze Micro Controller System Product Specification (DS865 v1.0), Jan. 2012, http://www.xilinx.com

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP10

COLLIMATOR MOTION CONTROL SYSTEM UPGRADE FOR MEDICAL LINEAR ACCELERATOR PROJECT AT SLRI

R. Rujanakraikarn†, P. Koonpong, S. Tesprasitte
Synchrotron Light Research Institute, Nakhon Ratchasima, Thailand

Abstract

A prototype of the 6-MeV medical linear accelerator has been under development at Synchrotron Light Research Institute (SLRI). A set of secondary collimators is utilized with different size arrangement for beam shaping purpose. To produce the desired field size of the beam, the FPGA-based collimator motion control is designed in VHDL for simultaneous control of the collimators while the main PI control is implemented in the FPGA's main processor. In this paper, hardware and software upgrades of the collimator motion control system are presented. A custom drive hardware for individual collimator is designed to implement with the existing FPGA controller board. Interface between the custom hardware parts and the FPGA's programmable logic (PL) part is described. Communication between the motion control subsystem and the main LabVIEW control software on PC is modified to send and receive parameters wirelessly. Software modification of the FPGA's main processor part and that of the LabVIEW GUI part is also reported.

INTRODUCTION

Currently, SLRI has been developing a prototype of the 6-MeV medical linear accelerator for cancer treatment. Reverse engineering approach has been employed in this research and development via a donated machine. The prototype consists of several subsystems, for examples, a linear accelerating structure, a 3.1-MW magnetron, a solid-state modulator, drive stand, a linac treatment head, and a central control system. All subsystems are connected to the central control system in a private network as shown in Figure 1. Introductory detail of this machine prototype can be seen in [1].

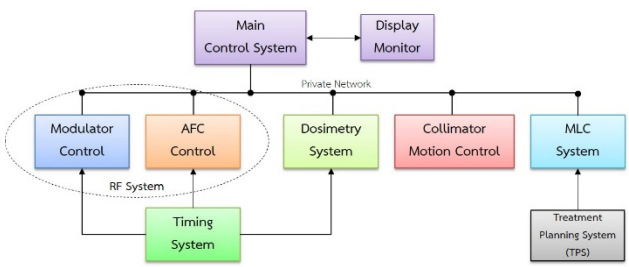


Figure 1: Network diagram of the machine prototype.

The FPGA-based motion control of collimators, both hardware platform and software implementation, is explained in detail in [1]. The main hardware parts consist of the secondary collimators and their drive PCBs, the Digilent's Zedboard and the 8-channel bipolar simultaneous sampling ADC (Analog Devices' EVAL-AD7606SDZ), and the digital IP blocks designed by VHDL in the programmable logic (PL) part of the FPGA. Digital feedback control utilizing PID controller for each of the collimator and the interface between hardware and control software, including LabVIEW GUI, are also described. The performance of the developed system is very satisfactory in order to control all collimators simultaneously.

In this paper, hardware modification and upgrade are presented. In-house electronic parts and custom drive system for individual collimator and the implementation with the existing FPGA controller board is described. Wireless communication and software development, together with the data transfer to the LabVIEW GUI for monitoring purpose, are explained. System performance and conclusion are discussed at the end of the paper.

HARDWARE

Several hardware parts of the motion control system have been designed and upgraded. Major modifications for this new system are new custom electronic PWM drive PCBs, DC power supply modules, wireless development boards, and a new ADC module. Each of them is explained in this section.

Custom PWM Drive PCBs

Since there are two adjustable pairs of collimator jaws in the linac treatment head with one pair installed above the other to provide symmetric and asymmetric fields, the motors that move the collimators must be driven independently. Traditional H-Bridge circuit based on IR8200B is used to provide appropriate voltage to control the motor speed with PWM signal. A custom PWM drive PCB is designed to drive a motor and to receive four signals from external controller board to control IR8200B. These four signals are PWM, Direction, Brake, and Motor Enable, and they are provided by the PL part of the Zedboard controller through the rear connector of the equipment box. This custom PCB is installed in a standard equipment box as shown in Figure 2. Each box also contains a wireless development board which will be described in the subsequent subsection.

† roengrut@slri.or.th

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP10

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Figure 2: Custom PWM drive PCB installed in a standard equipment box.

The PWM drive PCB was tested using the Zedboard controller and DC power supply as shown in Figure 3. In this initial test, a small dc motor was chosen just for verification of the circuit design on the new PCB. Figure 4 shows the result of the test. One LED is turned on when the motor rotates in a clockwise direction. Similar result occurs to the other LED when the motor rotates in a counter-clockwise direction.

Figure 3: Test setup for PWM drive PCB with Zedboard controller and DC power supply.

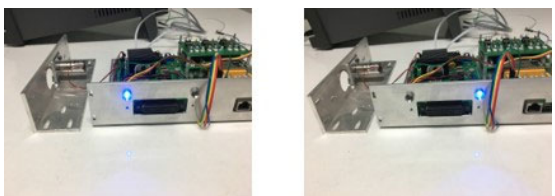


Figure 4: Test result of the new PWM drive PCB.

DC Power Supply Modules

Several DC voltage levels are required for the collimator motion control. For the collimators, 24 volts and 27 volts are needed to supply the motors driving the upper jaws and lower jaws, respectively. The other voltage levels for electronic circuits are ±15 volts, ±12 volts, and 5 volts. A custom set of DC power supply modules are built for the system. They are shown in Figure 5. The installation in the standard equipment rack is shown in Figure 6.

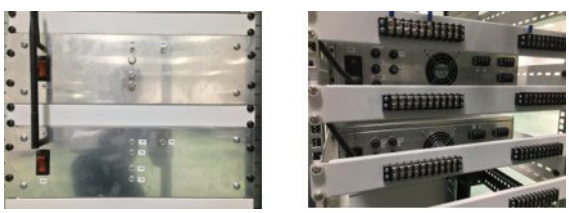


Figure 5: DC power supply modules designed in standard equipment box.

Figure 6: Equipment rack for PWM drive PCBs and DC power supply modules.

Wireless Development Board

LinkIt Smart 7688 Duo development board [2] is chosen to be our module to communicate wirelessly with the Main Control System Software on the PC. It is an open development board based on the OpenWrt Linux distribution, MT7688 microprocessor (MPU), and ATMega32U4 microcontroller (MCU). Arduino Breakout for LinkIt Smart 7688 Duo [3] is also used to make prototyping easier through simplified wiring in order to use additional Arduino shields. The installation of the boards is also shown in Figure 2. The main reason of using this wireless module is to send collimator position data over 2.4 GHz WiFi for monitoring and redundancy purposes.

New ADC Module

A new ADC module for sampling the position data of the collimator, which is in a form of bipolar DC voltage between -10 volts to 10 volts, is needed. An Extended ADC Shield [4] is chosen to complete this task. The 16-bit version of this shield utilizing Linear Technology LTC1859 is chosen so that it gives the same sampling resolution as that of AD7606 with the same number of channels (8 channels) but slightly lower sampling rate of up to 100 kSPS. It interfaces with the LinkIt Smart 7688 Duo development board over SPI.

SOFTWARE

Software implementation of the upgraded system is mainly developed for LinkIt Smart 7688 Duo development board. The primary purposes are to communicate wirelessly with the Main Control System Software and to sample the collimator positions using the Extended ADC Shield. Slight modification is performed for the LabVIEW GUI on PC to receive and display this new set of data. The main C program running the feedback control algorithm on the PS part of the Zedboard controller and the AD7606 ADC Interface IP designed by VHDL on the PL part remain the same to reduce complexity of the overall design.

Implementation and interface between LinkIt Smart 7688 Duo and Extended ADC Shield is shown in Figure 7. All eight potentiometer voltages representing collimator position signals are sampled by the Extended ADC Shield. The ATMega32U4 MCU is used to communicate with the shield by SPI interface, and the control software is developed in C on Arduino IDE using Library Functions described in [4].

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP10

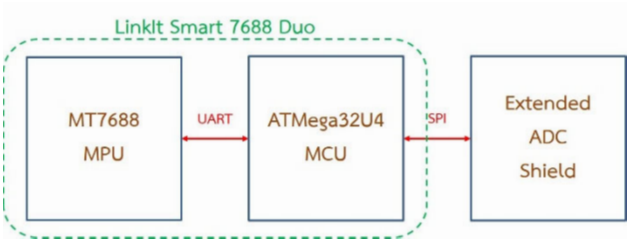


Figure 7: Interfaces between wireless development board and ADC.

Once all collimator position values are obtained by the MCU, they are sent to MT7688 MPU using UART interface. The MPU controls when to receive the data by sending either a "START" or a "STOP" command. This part of the software is developed in Python.

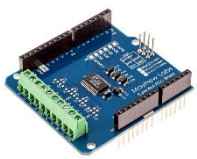


Figure 8: Extended ADC Shield [4].

On the Zedboard controller, the main C program and the AD7606 ADC Interface IP remain the same for real-time measurement and digital feedback control purposes, which can be referred to the block diagram in Figure 8 in [1]. All data from eight ADC channels of the wireless development board are sent to the GUI using 2.4 GHz WiFi. This set of data is used for position monitoring and redundancy purposes. Figure 9 shows the block diagram of the upgraded system. The LabVIEW program running a Main Control System Software is modified to receive the data via WiFi. All collimator position values received from the Extended ADC Shield are comparable to those sampled from the AD7606 daughter board of the FPGA. They are displayed on the GUI as shown in Figure 10.

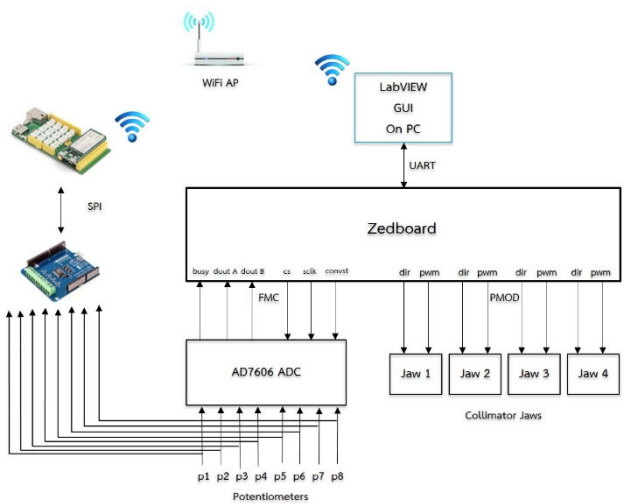


Figure 9: System block diagram of the updated system.

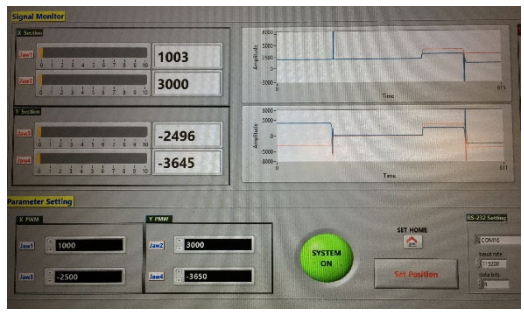


Figure 10: Main Control System Software showing collimator position values received wirelessly.

SYSTEM PERFORMANCE

In the hardware tests, all custom PWM drive PCBs and DC power supply modules were tested very carefully. In addition to the initial test described in the Hardware section, the PWM boards were supplied with external DC voltages from the power supply modules at 24 volts for upper collimators and 27 volts for lower collimators. The Zedboard controller was used to control the motion of the collimator. This ensures the compatibility of the new hardware with an existing system. In the software tests, all programs were carefully tested and debugged to make sure all processors and controllers, together with their interfaces, work correctly. Two sets of collimator position values, one from the existing FPGA system used in feedback control and the other from the wireless system used in monitoring purpose, were compared. The error between these two data set is within 1%. Even though the sampling frequency of the ADC of the wireless system is smaller, this error is acceptable for this prototype development because its ADC has the same number of bits (16 bits) as that of the FPGA system.

DISCUSSION AND CONCLUSION

The motion control system of the secondary collimator is upgraded. The hardware and software systems are modified to apply IoT technology to the existing system. The custom PWM drive system and DC power supply modules are designed and built. A number of tests have been performed to ensure system compatibility. The new wireless development board and 8-channel ADC module are chosen for sending collimator position values wirelessly. Initially, this new sampling system is used for monitoring and redundancy purposes only while the existing FPGA-based sampling system is used for digital feedback control. The existing LabVIEW GUI is modified to receive collimator position values from the wireless system. The resulting collimator position values from both systems are found to be comparable with small and acceptable error.

It is planned to perform further upgrades to this motion control system. It can be seen that GUI sends the desired setpoints and receives the sampled collimator position values via UART. To allow remote installation of all hardware of this motion control system, these data can be transferred wirelessly, with an appropriate interface between the Zedboard controller and the wireless development board. The

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP10

selection of either the existing ADC module driven by the FPGA logic or the new ADC shield with the SPI interface is left as an open choice since they have the same sampling resolution. In either case, real-time and simultaneous control of all collimator jaws can still be achieved.

REFERENCES

[1] R. Rujanakraikarn, P. Koonpong, "FPGA-Based Motion Control System for Medical Linear Accelerator Development at SLRI", in *Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS2017)*, Barcelona, Spain, Oct. 2017. doi:10.18429/JACoW-ICALEPCS2017-TUPHA069

[2] MediaTek LinkIt™ Smart 7688 Developer's Guide (v1.1), Feb. 2016, https://labs.mediatek.com/en

[3] Arduino Breakout for LinkIt Smart 7688 Duo, https://www.seeedstudio.com

[4] Extended ADC Shield User Manual, 2015, http://mayhewlabs.com

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP12

UPGRADING THE SYNCHRONISATION AND TRIGGER SYSTEMS ON THE VULCAN HIGH-POWER Nd:GLASS LASER

D. A. Pepler, I. O. Musgrave and P. B. M. Oliveira, STFC Rutherford Appleton Laboratory, Harwell Campus, Didcot, OXON. OX11 0QX, UK

Abstract

The Vulcan Neodymium-Glass High-Power Laser Facility at the Central Laser Facility [1] in the UK has been operational for over 40 years providing a world-leading and high-profile service to the international researchers in the field of Plasma Physics. Over that time the facility has had many modifications and enhancements to the buildings, the laser hardware and to the computerised control, synchronisation and timing systems.

As the laser systems have developed and the user experiments have continued to become much more complex and demanding, many new operational conditions have been required. The use of four independent laser oscillators with different properties - including temporal, spectral and operating frequencies - have meant that the optical and electrical multiplexing and the timing and synchronisation systems have all had to be adapted and extended to cope with these additional needs. However, these changes have resulted in the build-up of the overall system jitter to ± 250 ps between long (nanosecond) and short (picosecond) optical pulses and this is a limiting factor for time-critical experiments.

This paper will present some of the key changes and improvements that have recently been made.

INTRODUCTION

To significantly enhance the performance of the laser a number of key elements have had to be addressed - the number of different short-pulse oscillators have been reduced to two, namely a Spectra Physics [2] Tsunami Titanium: Sapphire oscillator and a Spectra Physics InSight DS oscillator and also the synchronisation and electronic timing systems have been replaced.

With the use of a commercial "Lok to Clock®" system on the Tsunami oscillator, the number of RF frequencies have effectively been reduced to one as this oscillator is now forced to operate at and in phase with the 79.926 MHz ± 80 Hz RF provided by the InSight oscillator.

This has enabled the fundamental timing system to be replaced with a commercial Master / Slave system from Greenfield Technology Ltd [3]. Figure 1 shows a schematic of the laser facility complex and the distribution of the new timing system. The system comprises of a model GFT 3001 Master Oscillator and

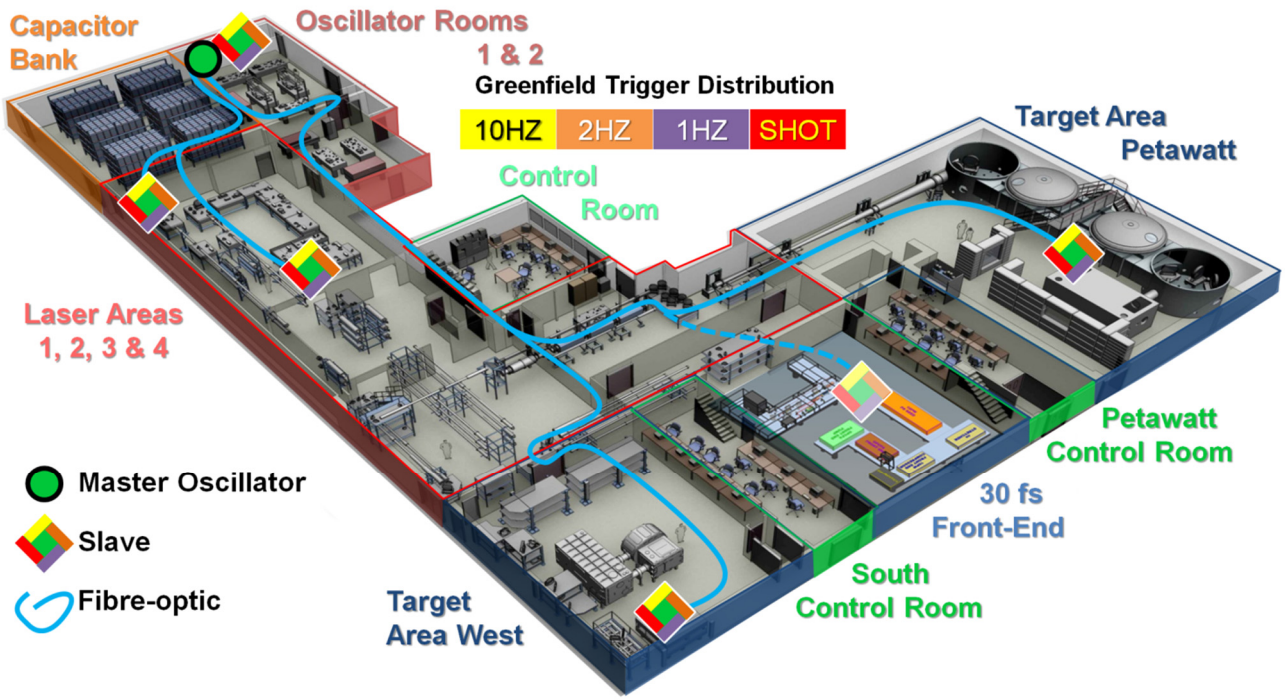


Figure 1: Schematic of the Vulcan Laser Facility showing the installation of the new trigger system. The Greenfield GFT 3001 Master Oscillator is located in Oscillator Room 2 with the GFT 1004 Digital Delay Generator modules widely distributed around the Facility and interconnected to the Master by a network of fibre optics.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP12

multiple GFT 1004 Digital Delay Generators (Slave) units. The installation has been carefully managed and for comparison purposes was initially arranged to be able to be operated in parallel with the existing timing system.

INSTALLATION

Figure 2 shows a photo of the Master Oscillator, single Slave, fibre-optic splitter and distribution modules installed near to the InSight and Tsunami oscillator table.

Figure 2: Installation of the GFT 3001 Master Oscillator and GFT 1004 Digital Delay Generator in Oscillator Room 2.

The testing revealed a number of issues to do with the frequency of the external oscillator that was being used to source the Master's external RF clock, occasional timing jumps of a few ns on some output channels and some inconsistencies with the single-shot sequences. However, we have always found Greenfield very responsive to investigating these problems and making firmware updates to resolve the issues.

COMMISSIONING

Following an extensive period of testing, the system was able to be switched to full operational status with minimal disruption to the user experimental programme. The system now provides options for 1 kHz, 100 Hz, 10 Hz, 2 Hz and 1 Hz repetitive signals as well as having single-shot capability.

A number of fibre-optics have been installed around the Facility, both within the laser bays and the target areas to accommodate the distribution of the Slave units.

CONCLUSION

The timing system is now much more immune to electrical noise; is also being integrated into the Vulcan Computer Control systems and provides a vastly superior performance in that the timing jitter is now sub 25 ps rms - a full order of magnitude reduction, as shown in Fig. 3.

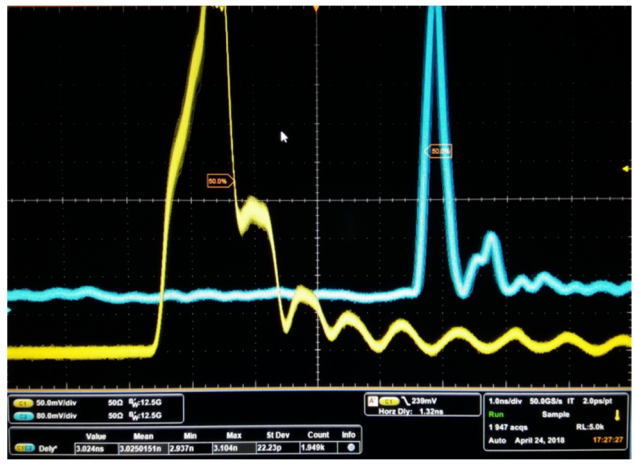


Figure 3: Oscilloscope traces of photo-diode signals comparing the relative timing of a short (ps) and long (ns) optical pulse demonstrating the jitter to be sub 25 ps rms.

As more Slave units are commissioned to replace other lower performance timing hardware, further timing improvements and possibilities for expansion are anticipated.

REFERENCES

[1] Central Laser Facility, www.clf.stfc.ac.uk

[2] Spectra-Physics, www.spectra-physics.com

[3] Greenfield Technology, www.greenfieldtechnology.com

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP13

THE DEVELOPMENT OF A FPGA BASED FRONT END SAFETY INTERLOCK SYSTEM

J. Y. Chuang, Y. Z. Lin, C. M. Cheng, Y. C. Yang, C. C. Chang, I. C. Sheng
National Synchrotron Radiation Research Center, 300 Hsinchu, Taiwan

Abstract

A front end (FE) safety interlock control system was designed to protect humans and the machine integrity during operation. Since stability and reliability are an important requirement in this system, we developed a FPGA based system to control a safety logic for interlock protection. The integration of the FPGA, Real-time and redundant fail-safe system in the FE interlock system enables us to provide a safe protection with EPICS communication and hardware protection functions.

INTRODUCTION

In the phase II TPS project, there are seven insertion device FEs, three bending magnet FEs, and one diagnostic FE. After the first two years of TPS operation, an event occurred when the interlock system crashed in the TPS FE05, 23, 41, and 45 [1] (see Fig. 1). As a consequence, some safety control system design faults were reviewed and modified. In order to enhance the stability of the interlock system, the original Real-time system with systematic risks was replaced by the FPGA based safety interlock system.

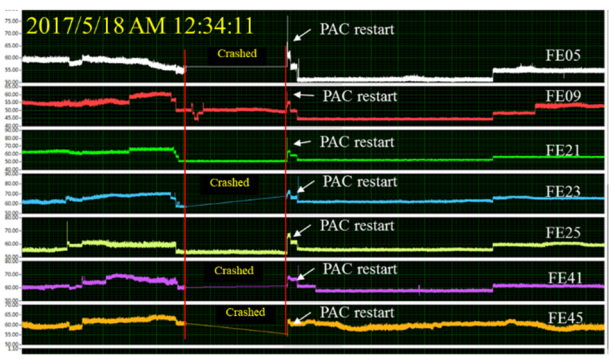


Figure 1: Record of a safety interlock controller crash (verified by CPU loading percentage).

STRUCTURE DESIGN OF THE SAFETY CONTROL SYSTEM

In the original TPS FE design, we used the National Instrument compact-RIO 9074 as the main controller for the safety interlock system and the logic program deployed in the Real-time system was operated by VxWorks. This design may cause a crash of the safety interlock system due to CPU loading or network risk [2]. In case of the TPS, the CPU loading of the cRIO 9074 was increased by using the distributed system manager (DSM) software to monitor the controller status. Due to safety concerns, the system needs to be separated into a safety logic program and a network protocol function into two independent systems. Therefore, the TPS FE interlock was upgraded from cRIO 9074 to cRIO 9030 and the logic program is now located on a field programmable gate array (FPGA) system. The Real-time system, operated by Linux, is engaged to publish the FE status by EPICS protocol. FPGA I/O nodes were created and its indicator for each corresponding share variable in the Real-time system to publish the EPICS protocol by cRIO 9030 network function. This structure means that the connection between FPGA and Real-time is a one-way transmission with no system resource concerns when the safety interlock system is operating as Fig. 2 shows. Therefore, the FPGA based safety interlock system combined with Real-time results in both stability and feasibility of safety and network function.

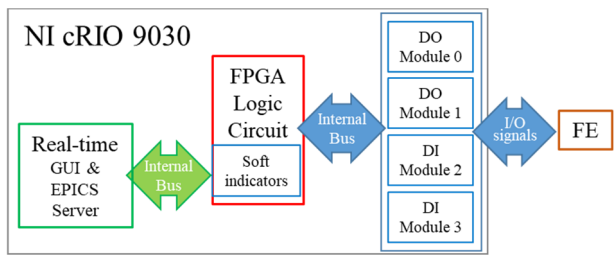


Figure 2: The structure of the FPGA based safety interlock control system.

In the TPS FE, the personal protection system (PPS) and machine protection system (MPS) were integrated into one safety controller NI cRIO 9030. Considering a fail-safe feature, a redundant system, which is controlled by a YOKOGAWA programmable logic controller (PLC) FAM3, is used to monitor the NI cRIO 9030 by hardwiring signal connections. The main controller for the safety interlock system NI cRIO 9030 and the redundant system FAM3 monitor each other by a 2 Hz heartbeat and watchdog individually. If any signal check detects a failure, the other controller will shut down the FE until FE staff resets the system manually. The control logic flow is shown in Fig. 3.

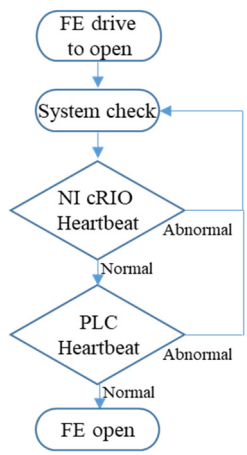


Figure 3: NI cRIO9030 and PLC FAM3 heartbeat logic checking flow.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP13

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

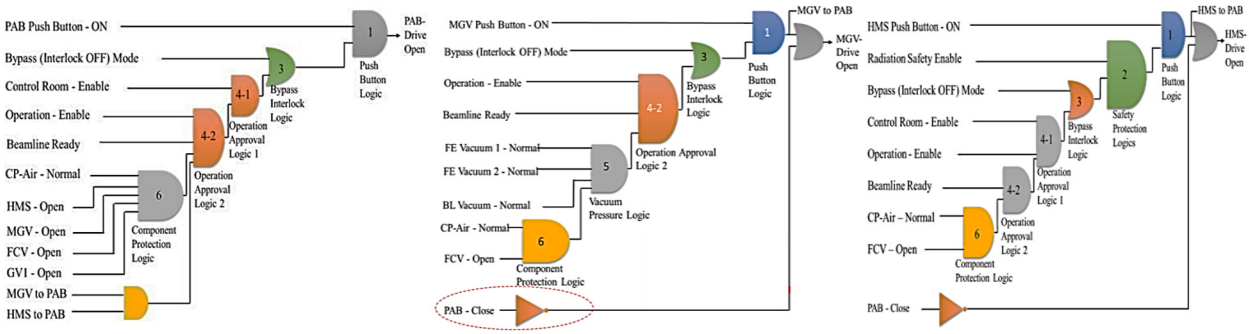


Figure 4: The main logic of PAB, MGV and HMS in TPS FE.

INTERLOCK LOGIC PROGRAMMING

The interlock program is a product of LabVIEW (LV) 2015 SP1 and was used to create the project in FPGA mode for all I/O node safety controls. This project is managed by auto-populating functions so that all sub-functions (in LV, it calls sub-VI) can be managed systematically without any effect when the file dictionary is changed. The main logic is divided into three main parts which are as follows:

- FE enable logic: this logic controls the photon absorber (PAB), all-metal gate valve (MGV) and heavy metal shutter (HMS) open or closed and also relates to beamline user operation allowance (as shown in Fig. 4) [3]. The logic contains all essential conditions of FE operation, such as cooling water, vacuum pressure, BL status, radiation safety enable, and central control system permission. All conditions in this logic are detected by TTL signals with 24 VDC and hardwire connections. The PAB, MGV, and HMS logic is shown in Fig. 5.

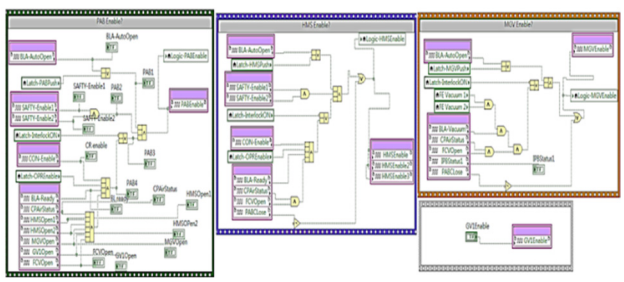


Figure 5: Program for the safety interlock logic in FPGA.

- Emergency control logic: because of the FE location being between BL and storage ring, its control system becomes an important link for machine safety protection. For personal protection, the FE belongs to the accelerator side and when a BL emergency is triggered, the FE is the first protector to block radiation by HMS and the electron beam is tripped at the same time. Therefore, the emergency logic must be very stable and reliable for operation. The emergency logic diagram is shown in Fig. 6.

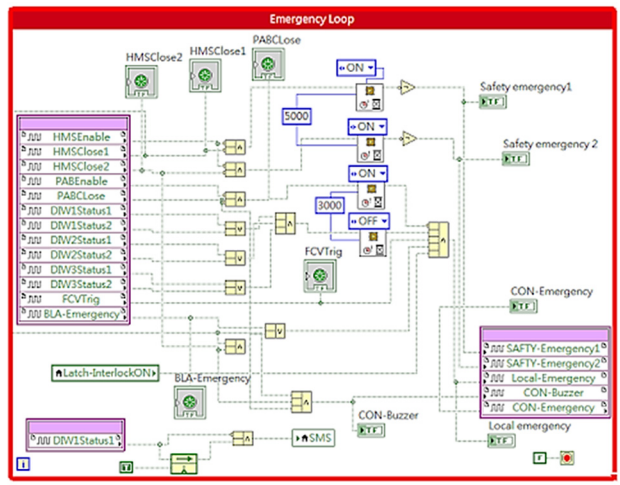


Figure 6: Emergency logic program for the safety interlock logic in FPGA.

- Interface logic: there were some control signals from others subsystems, the flip-flop logic and the denounced logic was widely used in the FPGA program to detect many different kinds of signals from subsystems such as BL or central control group. A self-hold circuit logic was used to detect the falling edge or rising edge status and complete the FPGA based safety interlock system. Figure 7 shows the status monitor for the logic program.

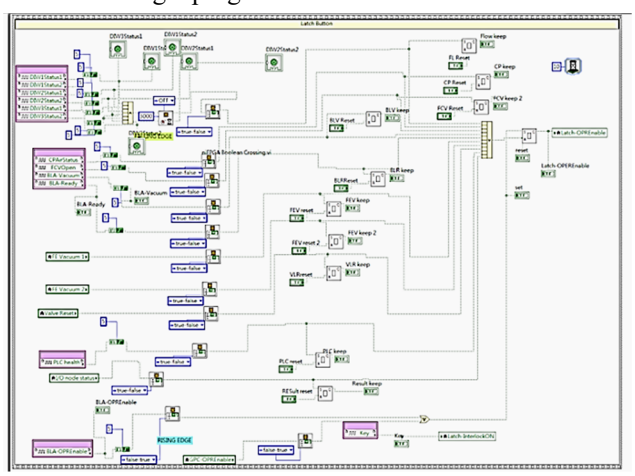


Figure 7: Interface signal detection program for the safety interlock logic in FPGA.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP13

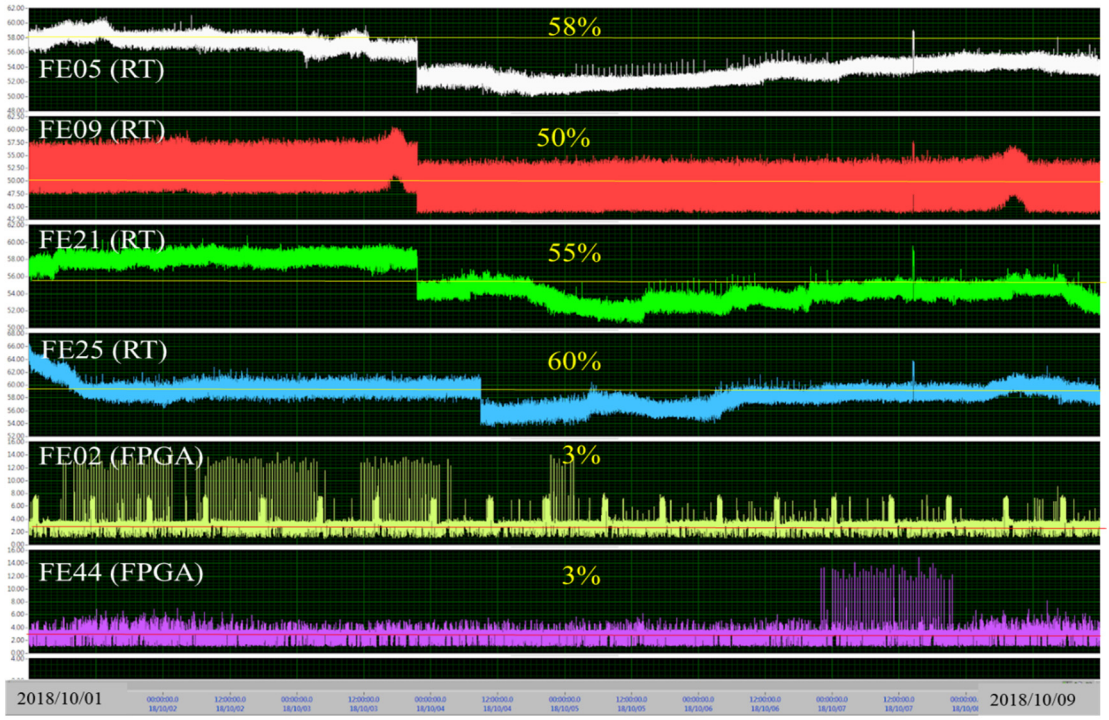


Figure 8: CPU loading comparison between RT based and FPGA based safety interlock system.

SYSTEM STABILITY ANALYSIS

The FPGA based safety interlock system was installed in phase II FE 02, 07, 44, and 39 respectively. To compare with phase I FE, the CPU loading of the FPGA based system is obviously lower than the Real-time based system. Figure 8 shows the CPU loading record, where the Real-time based system operated on average at 55%, while the FPGA based system was below 10 %. Although the FPGA based system is an independent safety interlock system without affecting the Real-time system, there are concerns about the NI cRIO 9030 being integrated into the same controller and we monitor the CPU loading to make sure the controller status and the EPICS system operate normal. To prevent network risks due to interlock instability, the communication system EPICS and Modbus servers are used as high-level firewall controllers between the server and the safety interlock controller. Finally, a redundant PLC system for the safety control system is utilized to monitor the main system and to prevent the crash of the safety interlock system. In addition, there is a hardwired connection between the FE HMS, BL and central control room, thus representing a highly reliable protection loop for beamline users.

CONCLUSION

From TPS FE commissioning experience, we find the NI cRIO 9030 provides a higher flexibility for EPICS communication and user interface development than traditional PLCs, but one has to separate the safety interlock and network function for safety concerns. The FPGA system in the cRIO 9030 can be a reliable hardware solution for safety interlock functions and it can also publish EPICS variables for monitoring system records and user information. After the upgrade of the TPS FE interlock system, the risk of failure is minimized with a redundant monitoring system and hardwired protection. This system should largely improve safety in TPS.

REFERENCES

[1] J.C. Liu, C.R. Chen, J.Y. Chuang, "TPS Beamline Radiation Safety Interlock, a Bizarre Incident and its Remedy Actions", 9th international workshop on Radiation safety at synchrotron radiation source, Hsinchu, Taiwan, Apr. 2017.

[2] National Instrument, https://forums.ni.com/t5/FIRST-Robotics-Competition/cRIO-Flashing-Error/td-p/3372316

[3] J.Y. Chuang, C.K. Kuan, I.C. Sheng, Y.Z. Lin, Y.M. Hsiao, Y.C. Yang, C.K. Chan, "Development and Construction of Safety and Control System for The TPS Front End Interlock", in *Proc. IPAC'17*, Copenhagen, Denmark, May 2017, paper TUPIK101.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP14

DEVELOPMENT of the NEW SPILL CONTROL DEVICE for J-PARC MR*

T. Kimura†, J-PARC-Center, KEK, Tokami-mura, Ibaraki, Japan

Abstract

J-PARC Main Ring (MR) is there are two operation modes of the fast extraction (FX) and slow extraction (SX) [1,2]. The SX operation used the spill control system. It consists of two kinds of Extraction Quadrupole Magnet (EQ), Ripple Quadrupole Magnet (RQ) and Spill Control Device with Digital Signal Processor (DSP) which calculates and controls it the optimal current pattern using the monitor signal of an extraction beam. It is used to make flatten the extraction beam structure and reduce the ripple noise. The present Spill Control Device needs to be reviewed from the aspect of service life etc. In this presentation, we will focus on improving the versatility of the device and the operability of the DSP program, and explain the development of the next-generation spill control device.

INTRODUCTION

The SX of the J-PARC MR utilizes third integer resonance at Qx = 22.333 [3]. After acceleration of MR, the beam is extracted slowly by betatron tune ramping of main quadrupole magnets in the flattop period of about two seconds. By constant tune ramping speed, the spill structure of slow extraction beam is like Gaussian shape. In physics experiment, the trigger rate or the counting rate of data acquisition system is limited by the hardware and software architecture. In some cases, the detectors cannot separate multiple events, due to collisions with too many particles and the target. In other cases, the data acquisition efficiency can be bad, due to a large dead time when too much beam is extracted. Therefore, the spill beam should be flat and stable sufficiently in extraction period. In order to make flat spill structure, we control the horizontal betatron tune by using the EQ magnets via spill control system. On the other hand, the ripples of main magnet power supply affect to spill structure by spike noise. We reject the ripple noise by RQ magnet.

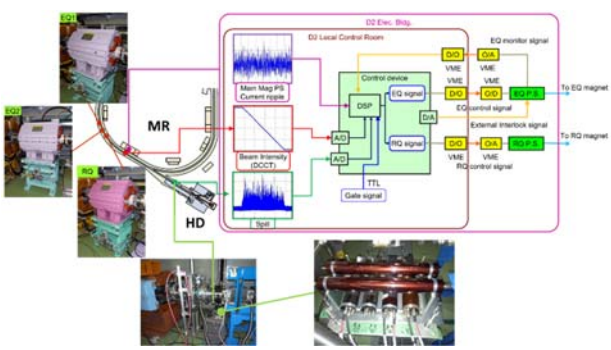


Figure 1: Signal flow of the spill control system.

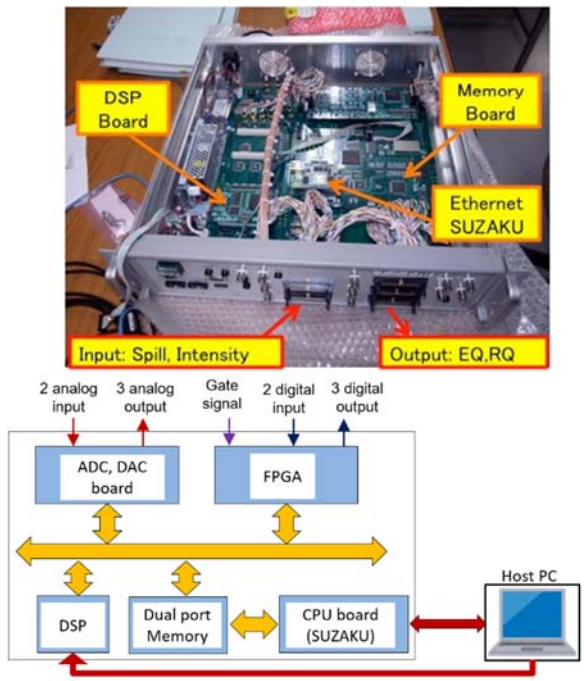


Figure 2: Spill control unit and signal flow.

SPILL CONTROL SYSTEM

Figure 1 shows the signal flow of the spill control system. The spill control unit provides control patterns to EQ and RQ power supply (PS). It is composed of circuit board for DSP, five analog/digital signal inputs, six analog/digital outputs interfaces and RS232 base serial communication interface. The circuit board consists of DSP card, dual port memories and FPGAs. Figure 2 shows the DSP spill control unit and signal flow. The DSP is in charge of spill control calculation. Dual port memory is used to connect the DSP and FPGA, and to share and read spill control calculation data. Table 1 shows the spill control unit spec. The analog input signals consist of control timing gate, residual beam intensity from DCCT and spill intensity of extracted beam from spill monitor.

Table 1: Spill Control Unit Spec

DSP board	TMS320C6713 (TEXAS INSTRUMENTS
ADC, DAC board	C6713DSK (TEXAS INSTRUMENTS)
Sampling frequency	100kHz
Analog Input/Output	±1V, 10kΩ
Digtal Input/Output	LVTTL3.3V, 32bit Positive logic

† kimurata@post.j-parc.jp

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP14

The digital input signals consist of EQ PS current signal for verifying control operation and Current ripple data of all main magnet PS for ripple canceller. The analog output signals consist of control value of EQ PS current for monitoring, control value of RQ PS voltage for monitoring and external interlock for stopping EQ PS. The digital output signals consist of EQ PS current control signal, RQ PS voltage control signal and EQ PS current control signal for Dynamic Bump system.

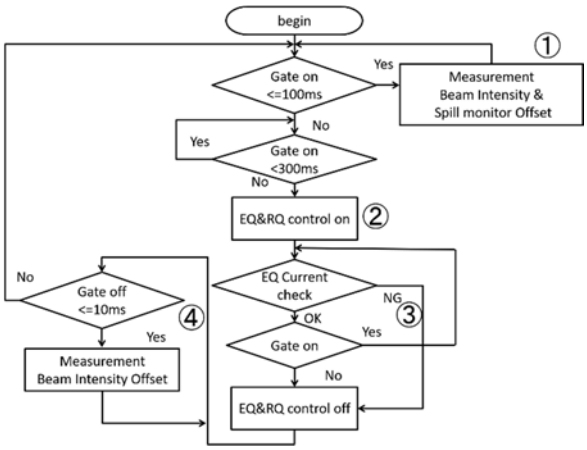


Figure 3: Spill control program flow.

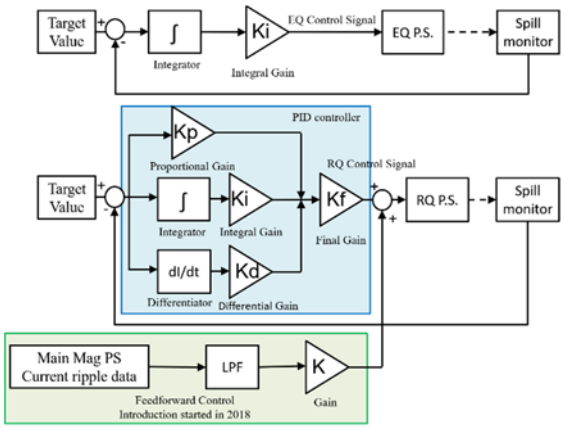


Figure 4: Block diagram of the spill control.

SPILL CONTROL ALGORITHM

Figure 3 shows the spill control program flow. Firstly, when the gate signal on, calculate circulating beam intensity before extraction and spill monitor offset in 100 milliseconds. Also, the target spill height to be controlled so that the beam is extracted in 2 seconds from these data is determined. Secondly, control of EQ and RQ PS is started 300 milliseconds after gate on, and beam extraction is performed. Figure 4 shows the block diagram of the spill control for EQ and RQ PS. Control of EQ makes flat spill structure by using integral control. Control of RQ reduce high frequency ripple noise by using PID control. In 2018, we started a test to cancel the ripple noise by feedforward control by inputting the current ripple data of the main magnet PS. Thirdly, when the gate off, measure the beam intensity offset in 10 milliseconds and prepare for the next beam extraction. Finally, when the EQ PS is being controlled, if the current control value and the output current readback value do not match, control of EQ and RQ PS is stopped. This is a countermeasure the malfunction of the EQ power supply that occurred in 2013. Figure 5 shows beam intensity, EQ current, RQ current, and spill at each timing.

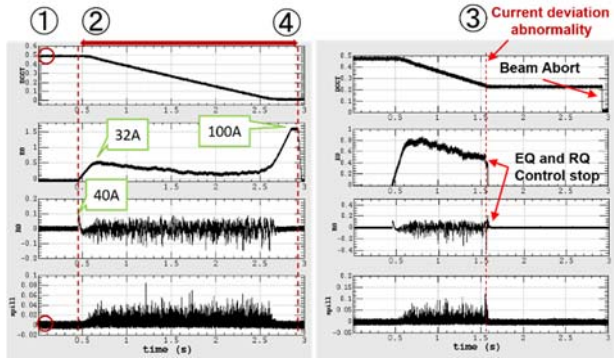


Figure 5: Spill control results.

NEW SPILL CONTROL SYSTEM

Ten years have passed since the development of the present spill control unit. Also, all prepared input/output interfaces including spares are in use, input/output cannot be added even when necessary., Therefore, we need to develop or select a new spill control unit. We selected DSP called SEAGULL made by MTT Corporation as a new spill control unit. Figure 6 shows the new spill control unit.

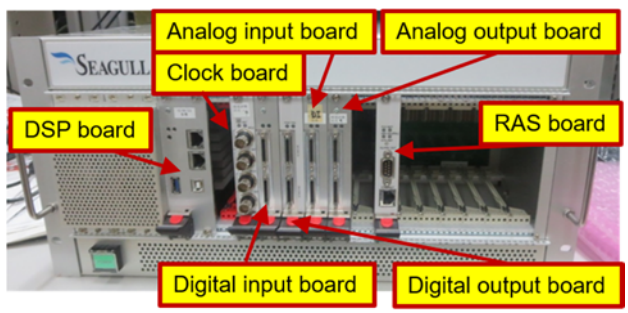


Figure 6: New Spill control unit.

It is composed of DSP board, clock board, RAS (Reliability, Availability, Serviceability) board and each input/output board. Table 2 shows the specifications of each board of the new spill control unit. Each board and the DSP board perform data communication using PCIe DMA transfer. Since each board is modularized, it is easy to add it when input/output needs to be added in the future. Also, we plan to rebuild the spill control system when updating the spill control unit. In the present spill control unit, we can write control program, change control parameters, monitor control state using only dedicated PC. In the new spill control unit, we only perform maintenance, such as changing control programs, on a dedicated PC.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP14

Table 2: Specification of the New Spill Control Unit

Backplane	PCI Express Gen2
DSP board	KeyStone□ 66AK2H06AAAW24 (TEXAS INSTRUMENTS) ARM Core 1.4GHz×2 DSP Core 1.2GHz×4
Clock board	System Clock ~16MHz
Sampling Frequency	100kHz
Analog Input/Output	32ch 16bit ±10V, 100kΩ
Digtal Input/Output	LVTTL3.3V, 32bit Positive logic

We will make improvements to change control parameters and monitor control status via EPICS IOC from general EPICS client. Figure 7 shows Network of the New Spill control system.

CONCLUSION

The present spill control system introduced in 2009 has contributed to improving the quality of slow extraction beam. However, it took ten years from the development and new development became necessary. In consideration of future input/output scalability, a module type DSP was selected. We would like to develop it aiming for operation from 2019.

REFERENCES

[1] K. Hasegawa *et al.*, "Performance and Status of the J-PARC Accelerators", *Proc. of IPAC 2017*, Copenhagen, Denmark, May 14-19, 2017, pp. 2290-2293.

[2] S. Igarashi *et al.*, "Accelerator Based Neutrino Experiments T2K J-PARC", *Proc. of 13rd Annual Meeting of Particle Accelerator Society of Japan*, Chiba, Aug. 8-10, 2016, pp. 14-18.

[3] M. Tomizawa *et al.*, "Present status and future plans of J-PARC slow extraction", *Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan*, Chiba, Japan, Aug. 1-3, 2016, pp. 70-74.

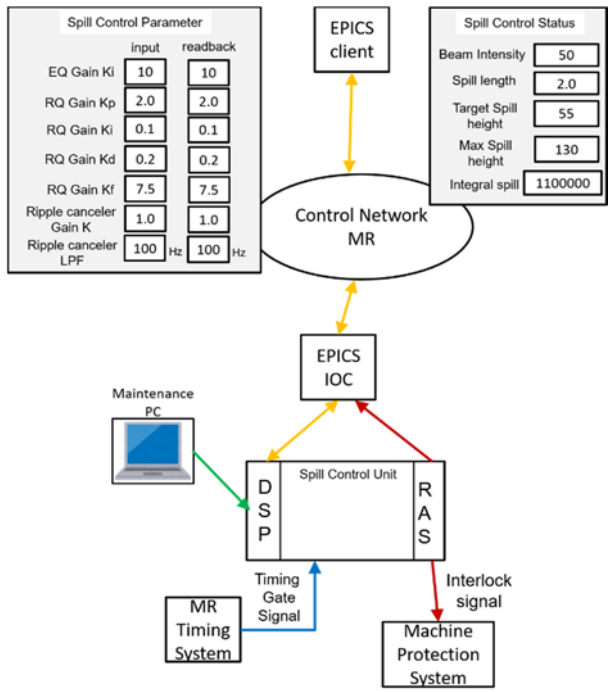


Figure 7: Network of the New Spill control system.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP15

DESIGN AND IMPLEMENTATION OF FPGA BASED PROTECTION SYSTEM FOR BEAM ACCELERATION IN LINEAR IFMIF PROTOTYPE ACCELERATOR

Y. Hirata†, A. Kasugai, National Inst. for Quantum and Radiological Sci. and Technology, Rokkasho, Japan
H. Takahashi, Japan Atomic Energy Agency, 319-1195 Tokai, Japan
A. Marqueta, A. Jokinnen, Fusion for Energy, 039-3212 Rokkasho, Japan

Abstract

The IFMIF (International Fusion Materials Irradiation Facility) Prototype Accelerator (LIPAc) has been developed in the Engineering Validation and Engineering Design Activity (EVEDA) phase. The LIPAc is designed to produce a deuteron CW beam with a current and an energy of 125 mA and 9 MeV. After the injector campaign, the LIPAc is entering the RFQ (Radio-Frequency Quadrupole) commissioning phase in which such subsystems as the RFQ, RF system and Beam Instrumentation systems have been attached.

The LIPAc control system consists of local control systems (LCSs) and the central control system. The LCSs haven been developed by Europe and delivered with the subsystems; and the central control system, including personnel and machine protection, timing, archiving and alarming, has been developed by Japan. Japan and EU are jointly integrating them to control the whole accelerator in an organized manner.

Fabrication of the control system is made on the EPICs platform to reduce the risk in integration. Some part of the protection systems has been implemented in FPGA to satisfy both the speed and sophisticated control. The basic idea and implementation of the control system will be presented.

INTRODUCTION

The IFMIF Prototype Accelerator has been developed in the Engineering Validation and Engineering Design Activity phase. The final target of the LIPAc is to generate a heavy deuteron CW beam with a high intensity and 125 mA and an energy of 9 MeV for an average power of 1.125MW.

The control system of the LIPAc consists of six subsystems: Central Control System (CCS), Local Area Network (LAN), Personnel Protection System (PPS), Equipment Protection System (MPS), Timing System (TS), and Local Control System (LCS) of subsystems. Europe and Japan have been developing the control system jointly, in which Europe is in charge of LCSs and Japan five other systems. Upon completion of the implementation, the LIPAc is planned to be operated from the CCS and the operational parameters and experimental data can be monitored from the CCS and archived by the archive system.

After the completion of the injector campaign (Phase A), the RFQ, RF system, LLRF, MEBT and BI (beam instrumentation systems) are being attached to the injector (Fig. 1) and the development of the LIPAc has entered the RFQ commissioning phase (Phase B) [1].

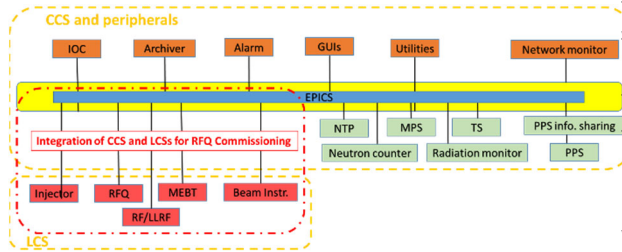


Figure 1: Control system configuration for RFQ commissioning.

PPS and MPS have an aspect of final protection scheme. Parameters such as thresholds, etc., are not supposed to be changed once they are determined. So, they are implemented basically using PLC-based technology for the general slow interlocks. For faster interlocks, such custom-made cards as those with Application-Specific Integrated Circuits (ASICs) have been needed when both speed and reliability are mandatory.

In recent years, an architecture using FPGA is frequently used to achieve both time performance and reliability with more flexibility. As the logic required some systems are more complicated, FPGA is advantageous in faster development, adaptability, reusability and therefore reduced final costs.

In such a case as IFMIF, subsystems are developed and delivered by different countries and detail specs are not fully prepared/agreed in advance, since some interlock conditions and their criteria could be complicated and likely determined from experiments. From these reasons some part of the LIPAc protection system uses an FPGA architecture.

In this workshop, the authors will present cases of such inter locks as chopper [2], beam counting and overload protection for diagnostics system used in LIPAc.

CHOPPER INTERLOCK

A chopper is used to allow interceptive diagnostics after the RFQ. Due to the high power and long rise time of the beam extracted from the ion source, pulsed mode operation is used (Fig. 2) to prevent damage on the diagnostics. A high voltage is applied to the chopper which deflects the beam usually. In response to the timing sig-

† hirata.yosuke@qst.go.jp

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP15

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

nal, the voltage applied to the chopper turns off to pass the beam toward the RFQ. The beam is trimmed or "chopped" to generate a sharper pulse (Fig. 2).

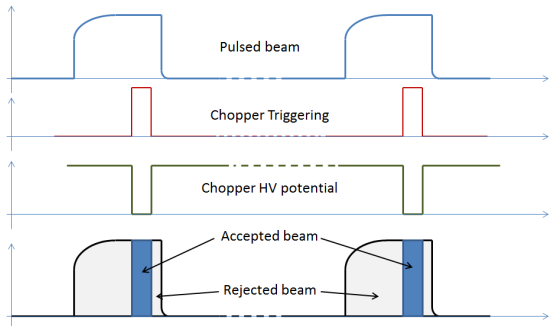


Figure 2: Beam chopping image to create a sharper pulse.

The operation of the chopper is interlocked with several conditions, meaning that the CIM needs to interface with different systems to ensure correct operation. Due to this complexity and trial-and-errors anticipated in the development phase, an architecture using FPGA based on RIO hardware from National Instruments [2] has been adopted.

At the time of development of the Chopper Interlock Module (CIM), LIPAc was already running in the injector commissioning phase. The CIM had to be integrated with the control system (based on EPICS) and other subsystems without interference. Consequently, the CIM was designed to have “armed” status before operation rather than an always-on system.

The first purpose of the chopper is to generate precise pulse lengths prevent overload to the downstream diagnostics, etc. When the CIM is armed, it oversees the chopper triggering gate pulse and takes proper actions depending on the status of the gate signal: gate signal off, gate-on rising, gate-on and gate-off falling. The transition and major actions on each status are summarized in Fig. 3. It should be noted that in this figure non-mentioned status signals from PLCs, etc., are assumed all correct.

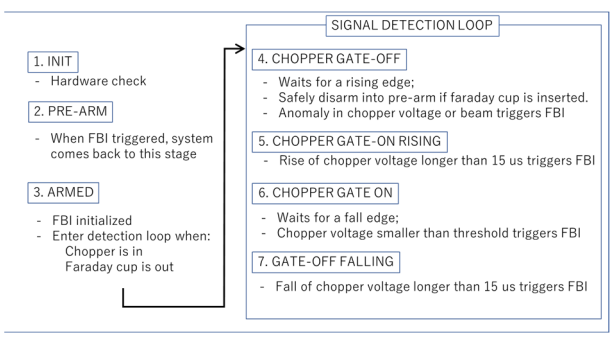


Figure 3: Transition and actions of CIM.

Table 1 summarizes the hardware with major specifications that was selected to meet the requirements for the project. Figure 4 shows the system hardware, together with the signals that the CIM must monitor. The NI CRIO-9068 chassis hosts a microcontroller running a Real-Time Linux. This allows the use of NI EPICS Channel Access (CA) Client and Server so the CIM should be successfully integrated into the LIPAc control system.

The acquired data is transmitted to the LabVIEW RT host and updated on the EPICS Process Variables. The implementation in LabVIEW is a two-stage loop performing read operations and then writing configurable parameters such as thresholds to the FPGA. These configuration parameters will be used in pre-armed state at the beginning of the interlock operation.

Table 1: RIO Components Used in CIM

HARDWARE	PURPOSE
NI cRIO-9068	CompactRIO chassis with embedded controller and 8 slots for modules. FPGA for the interlock, and dual-core CPU for EPICS integration.
NI-9223	Microsecond [-10,+10V] analog 16-bit input module for the High Voltage and ACCT readings. Conditioned by a voltage divider.
NI-9401	High-speed configurable I/O digital TTL module. Configured as a read module to interface with the timing system.
NI-9422	24V industrial logic level input sinking module. This reads signals from the PLC interlocks.
NI-9474	24V industrial logic level output sourcing module. This module drives the signal FBI to low level in the case of an interlock action.

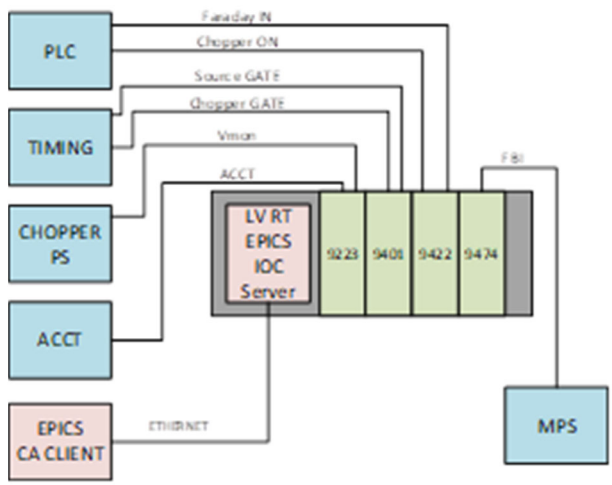


Figure 4: Hardware constitution of CIM.

The solution based on the EPICS CA Client and Server provides a useful mechanism that monitors CIM variables. However, there is limitation in this scheme such that the closed solution does not provide any configuration of the EPICS records and improving this could be a next issue for the extension of FPGA use in the protection systems.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP15

OVERLOAD DETECTION FOR BEAM INSTRUMENTATION

Several interceptive devices such as SEM grid are installed to examine the beam quality. SEM grid uses thin wires with which the beam profile or other parameters can be measured. While measurement, the beam continues to deposit energy and cause damage on the wire.

The overload detector monitors the beam current, caluculate the deposit energy accumulated on the wire and stops the beam once the acuumulated energy exceeds certain values. These thresholds, however, vary depending on various conditions : beam current and size especially, other than the statuses of other devices.

The SEM grid devlopement group carried out thermal simulation and provided maximum beam length allowed as functions of beam current and size. We simplified these conditions and implemented in FPGA (XC7A100T-1FTG256C) with VHDL as a core part of the overload detector. The simplified contiditions are summarized in Table 2. Other conditions such as statuses of other systems—FC, chopper, beam direction—are considered by the software.

Table 2: Maximum Beam Length Acceptable

	$\sigma_{beam} \leq 10.5$ [mm]	$\sigma_{beam} \geq 10.5$ [mm]
I_{beam} = 0~3 [mA]	$247.78 \cdot \frac{{\sigma_{beam}}^2}{15}$	$247.78 \cdot \frac{110}{I_{beam}}$
I_{beam} = 3~15 [mA]	$247.78 \cdot \frac{{\sigma_{beam}}^2}{15}$	$247.78 \cdot \frac{110}{I_{beam}}$
I_{beam} = 15~125 [mA]	$247.78 \cdot \frac{{\sigma_{beam}}^2}{I_{beam}}$	$247.78 \cdot \frac{110}{I_{beam}}$
$I_{beam} \geq 125$ [mA]	$247.78 \cdot \frac{{\sigma_{beam}}^2}{I_{beam}}$	$247.78 \cdot \frac{110}{I_{beam}}$

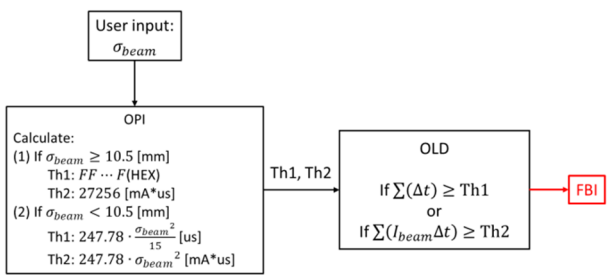


Figure 5: Basic logic of core OLD.

The beam size, which is an input parameter and given by the operator, should be inferred or guessed since it is not precisely known before using the SEM grid. Based on the formulae in Table 2, two threshold values for maximum beam length are calculated in the OPIs and transferred to the FPGA (Fig. 5). The judge as to whether the accumulated current exceeds the threshold is made within the FPGA.

Figure 6 shows the condition of interlock triggering for a beam size of 8 mm. The boundaries are the conditions of Table 2. And if the beam accumulated count exceeds one of the thresholds, the detector triggers an interlock.

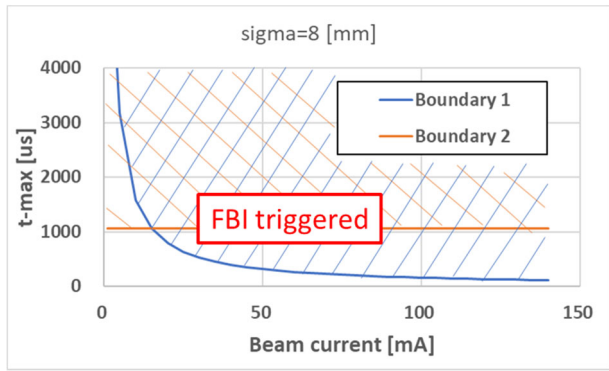


Figure 6: Condition of interlock triggering.

These conditions are complicated and subject to change according to the measurement results. Table 2 could be improved while performing measurements.

SUMMARY

As the development of control systems has been carried out separately in Europe and Japan, a flexible approach is necessary for the protection system to allow effective development. The examples presented in this report shows, the use of FPGA in some complicated conditions is effective in development flexibility in addition to its transaction speed and reliability.

ACKNOWLEDGEMENT

The present work was performed in the framework of the Broader Approach Agreement. The authors gratefully acknowledge the support of their home institutions and research funders for this work. Views and opinions expressed herein do not necessarily reflect those of QST, Fusion for Energy, or of the authors' home institutions or research funders.

REFERENCES

[1] M. Sugimoto *et al.*, "Progress Report on LIPAC," in *Proc. LINAC'18*, Beijing, China, Sep. 2018. doi:10.18429/JACoW-Linac2018-TU2A04

[2] M. Astrain *et al.*, "FPGA-based interlock system for the chopper of the Linear IFMIF prototype accelerator injector", *Proc. 30th Symp. on Fusion Technology (SOFT'18)*, Sicily, Italy, September 2018.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP16

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

REMOTE WAVEFORM ACCESS SUPPORTS WITH EPICS FOR TPS AND TLS CONTROL SYSTEMS

Y. S. Cheng, C. Y. Liao, C. Y. Wu, K. H. Hu, K. T. Hsu
National Synchrotron Radiation Research Center, Hsinchu 30076, Taiwan

Abstract

To eliminate long distance cabling for improving signal quality, the remote waveform access supports have been developed and applied on the TPS (Taiwan Photon Source) and TLS (Taiwan Light Source) control systems. Waveforms include pulse magnets power supplies waveforms, AC waveforms of main power supplies, LLRF waveforms, beam signals, etc., and these are necessary to be monitored during routine beam operation. One is that use the EPICS-embedded data acquisition systems which are formed by the Zynq System-on-Chip architecture to capture the waveform signals; the other is that a dedicated EPICS IOC is used to communicate with the present Ethernet-based oscilloscopes to acquire each waveform data. According to specific purposes use, the different graphical applications have been developed and integrated into the existing operation interfaces. These are convenient to observe waveform status and to analyze the acquired data on the control consoles. The efforts are described at this paper.

INTRODUCTION

TPS is a new and highly bright synchrotron light source constructed at National Synchrotron Radiation Research Center (NSRRC) in Taiwan. It consists of a 150-MeV electron linear accelerator, a booster synchrotron, a 3-GeV storage ring and experimental beam lines. Civil construction began in February 2010 and was completed in the first half of 2013. Installation and integration of the accelerator system began later in 2013. The control system environment was ready in middle of 2014 to support the testing and commissioning of the final subsystem integration without the beam. Commissioning with the beam was successful in December 2014.

TLS is a third generation of synchrotron light source built at the National Synchrotron Radiation Research Center (NSRRC) site in Taiwan, and it has been operated since 1993. The TLS consists of a 50 MeV electron Linac, a 1.5 GeV booster synchrotron, and a storage ring with 360 mA top-up injection. The TLS Control system is a proprietary design [1]. It consists of console level workstations and VME based intelligent local controller (ILC) to interface with subsystems. Hardware and software on console level workstation change several times due to evolution of fast evolution of computer technology. Due to the well design of the original control software structure, port to new Linux platform without difficult.

EPICS (Experimental Physics and Industrial Control System) is a set of open source software tools, libraries and applications developed collaboratively and used to create distributed soft real-time control systems for scientific instruments such as the particle accelerators and large scientific experiments [2].

EPICS were chosen as control system framework for the Taiwan Photon Source (TPS) of 3 GeV synchrotron light source [3]. The TPS control system with EPICS mechanism had been integrated and commissioned. On the other hand, in order to adopt update technology and re-use expertise of manpower, the upgrade and maintenance for TLS control system adopts the EPICS as its framework was decided. Moreover, some new installed subsystems runs EPICS control environment also. Mixed existed TLS control system and EPICS environment were proofed without problem

One of EPICS database records is the waveform record which stored data array acquired from the device. There are many waveforms such as signals which should be observed in synchrotron light sources including waveform variation of booster current, beam current measured by the fast current transformer, and filling pattern, RF pulse, pulse magnet current, etc. Acquiring waveform data should be based upon EPICS waveform supports. Through PV (Process Variable) channel access the client console can observe the waveform by using various toolkits (EDM, CS-Studio, MATLAB, Python, etc.). In addition, one of the benefits on EPICS waveform supports is that LXI-compliant [4] oscilloscopes with Ethernet interfaces can be adopted to be controlled remotely and users can observe the related waveform data on the control consoles easily.

Another solutions of waveforms access for some parts of subsystems are using the standalone EPICS embedded data acquisition system and the PCI-Express form factor digitizer. The standalone EPICS embedded data acquisition system is convenient to access data by EPICS PVs via Ethernet interface. The PCI-Express form factor digitizer is plugged into Linux-PC with PCI-Express slot, and the related functional libraries of device driver are necessary for building as an EPICS IOC. The efforts of implementing remote waveforms access supports with EPICS for the TPS and TLS control systems are summarized in the following paragraphs.

SYSTEM ARCHITECTURE OF WAVEFORM SUPPORTS WITH EPICS

Remote waveforms access supports with EPICS mechanism have been developed and applied on both TPS and TLS control systems. The system architecture of waveform supports is illustrated as Fig. 1. During the commissioning phase of TPS control system, the oscilloscopes have been widely adopted to monitor

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP16

variations of subsystems, and also used for adjusting parameters on-site. In order to observe the waveforms data on control consoles for beam operation, the EPICS supports of accessing waveforms are necessary to be implemented.

Standalone fan-less PC-based platform that has a Linux operating system was set up as the dedicated soft-IOC to connect with oscilloscopes via an Ethernet interface. An IOC uses the VXI-11 [5] or TCP/IP protocol to communicate with LXI or Ethernet based oscilloscopes. Control consoles can observe the waveform data by use of specific GUI toolkits through EPICS PV channel access.

At the routine operation phase, a part of subsystems' waveforms variations have to been observed and archived by long time and need not to use oscilloscopes of high sample rate for data acquisition. The commercial standalone data acquisition system with embedded EPICS IOC is suitable to use on this purpose. The operation interfaces based on difference purposes have been implemented and are executed to access their waveform data via EPICS channel access on the control clients.

The digitizer with high sample rate and high bandwidth is under test. The digitizer which is PCI-Express (PCIe) form factor is installed into a Linux-PC with PCIe slot as an EPICS IOC. The IOC can directly access waveform memory of digitizer by use of provided software development kit (SDK). The EPICS support with SDK of digitizer needs to be developed for PVs channel access.

Some waveforms date have been calculated and translated to specific data formats to be easily monitored and archived into database server. Consoles apply toolkits to show trends of observed signals and to retrieve the archived data for long term analysis.

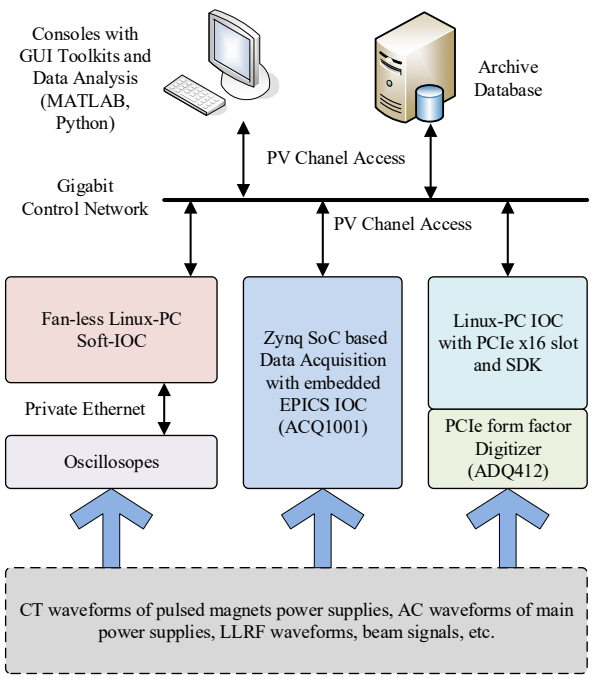


Figure 1: System architecture of waveform support with EPICS for TPS and TLS control systems.

WAVEFORM SUPPORT OF ETHERNET-BASED OSCILLOSCOPES

Remotely access subsystems waveforms from oscilloscopes are set up for eliminating the requirement of long-distance cabling to improve the signal quality. To implement the EPICS supports of LXI/Ethernet-based oscilloscopes, the device supports were built to communicate oscilloscopes with the device driver. The related record supports were created with a link to the device supports. The software block diagram of establishing EPICS support for oscilloscopes is shown as Fig. 2.

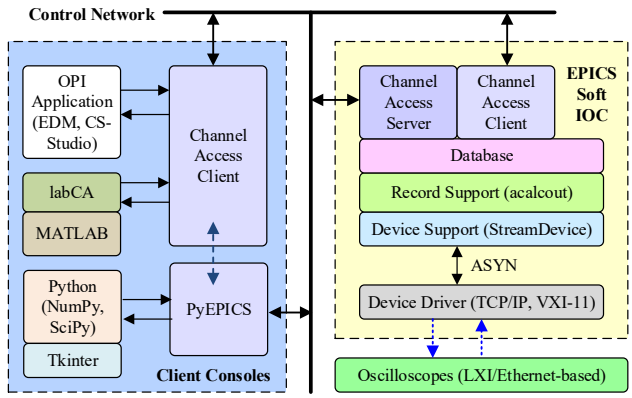


Figure 2: Software block diagram of building EPICS support for acquiring waveforms from LXI/Ethernet-based oscilloscopes.

A user can easily analyse the acquired waveform data using specific toolkits (Python, C/C++, MATLAB, etc.) with the EPICS channel access library. For example, a higher resolution (12-bit) oscilloscope has been used to observe a beam signal from one of BPM (Beam Position Monitor) for fill pattern measurement. To monitor every bunch current, the waveform PV data acquired from oscilloscope are processed online by the MATLAB toolkit and formed to one dedicated waveform PV. The GUI has been created to show the calculated fill pattern as Fig. 3.

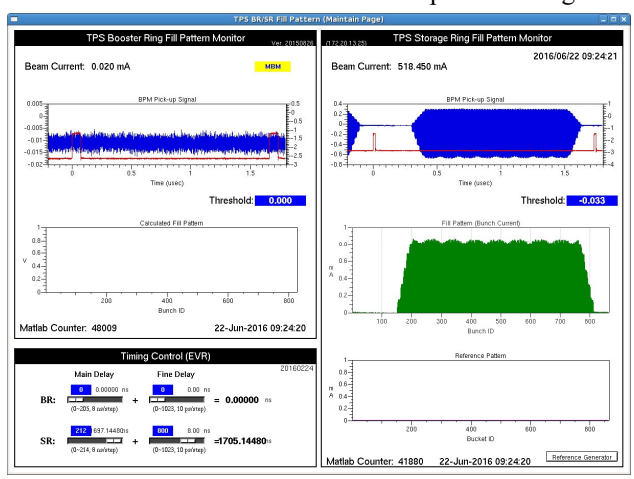


Figure 3: GUI created for monitoring fill pattern of TPS storage ring.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP16

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ZYNQ SOC BASED DATA ACQUISITION WITH EMBEDDED EPICS IOC

A part of subsystems' waveforms need to been continuously observed variations without higher sample rate acquisition during route operation. The ACQ1001 data acquisition [6] is suitable on this purpose and also with low power consumption. The ACQ1001 is a smaller size appliance with external 12V DC power, and supports both provided FMC and ELF modules. It is carried the Zynq System-on-Chip architecture with featuring low power and integral FPGA device. The ACQ1001 supports external clock and trig options, and can be synchronized between units using HDMI cable bus. The outward of ACQ1001 is shown as Fig. 4. Its software environment equips the Linux operation system, and an EPICS IOC is embedded as standard.

Figure 4: Outward of ACQ1001 with FMC module.

The ACQ1001 with ACQ430FMC module supports eight simultaneous channels with 24-bit 128 kSPS. Three ACQ100 data acquisitions have applied on the TLS control system of booster ring. These are used to monitor waveforms of booster main power supplies, beam current, extraction pulsed power supplies CTs for long term. The GUI for monitoring essential waveforms has been created as shown in Fig. 5, and waveforms observation integrates with new ACQ1001 system and EPICS support of existed oscilloscopes. The EPICS applications have been gradually supported and operated on the consoles environment of TLS control system.

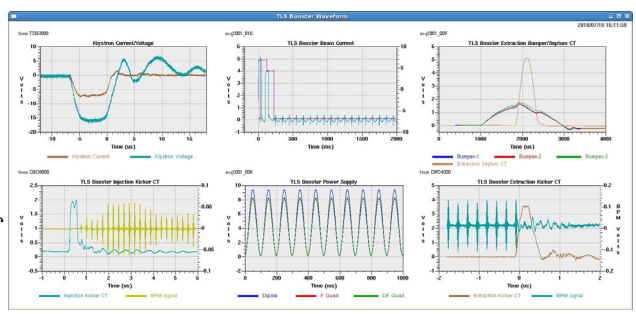


Figure 5: EPICS GUI of monitoring essential waveforms which integrated with new ACQ1001 and existed oscilloscopes for TLS booster ring.

PCI-EXPRESS FORM FACTOR DIGITIZER

The ADQ412 digitizer [7] is equipped up to 4 GSPS sampling rate per channel, up to 2 GHz analog bandwidth and 12-bit vertical resolution. It also supports four input channels, and external clock reference and external trigger input. The ADQ412 digitizer is available in the PCI-Express form factors as shown in Fig. 6, and PCIe for integration in a PC platform. The PC with installing ADQ412 is as an EPICS IOC to be developed the waveform access support with provided SDK. The software framework of EPICS support is under construction and test, and can be verified with actual signals on-site in the future.

Figure 6: Outward of ADQ412 with PCIe form factor.

SUMMARY

To eliminate long distance cabling for improving signal quality, remote waveform access supports are necessary to be implemented. A dedicated EPICS IOC is used to acquire waveform data from Ethernet-based oscilloscopes. In addition, the EPICS IOC embedded data acquisition system which formed by the Zynq System-on-Chip architecture has been also used to monitor waveform during route operation. The EPICS support of PCIe digitizer with high sample rate and high bandwidth under construction and test. The GUI of waveform supports for specific purposes have been created and applied on the TPS and TLS control systems, and enhanced continually.

REFERENCES

[1] G. J. Jan *et al.*, "Computer Control and Instrumentation System at the SRRC", *Nucl. Instrum. Methods Phys. Res.*, Sect. A 352 (1994) 33-39.

[2] EPICS, https://epics.anl.gov/

[3] C. Y. Liao *et al.*, "Commissioning of the TPS Control System", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, paper FRB3O01.

[4] LXI, http://www.lxistandard.org

[5] VXI-11, http://www.lxistandard.org/about/vxi-11-and-lxi.aspx

[6] ACQ1001, http://www.d-tacq.com

[7] ADQ412, https://spdevices.com/products/hardware/12-bit-digitizers/adq412

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP17

INJECTION AND EXTRACTION TIMING CONTROLS AT SuperKEKB DAMPING RING

H. Sugimura*, H. Kaji, F. Miyahara, S. Sasaki, M. Satoh, KEK, Ibaraki, Japan
T. Kudou, S. Kusano, Mitsubishi Electric System & Service Co., Ltd, Ibaraki, Japan
Y. Iitsuka, East Japan Institute of Technology Co., Ltd, Ibaraki, Japan

Abstract

SuperKEKB project aims at the world highest luminosity to 8×10^{35} cm^{-2}s^{-1}. To achieve the luminosity, a lot of equipment was newly constructed or upgraded. Especially, a Damping Ring (DR) was newly constructed at the middle of the injector linac for making lower positron emittance. The DR timing control system was also fabricated as a branch of the main SuperKEKB timing system. The synchronized timing is generated at the main timing system. It is received at DR sub-timing station and is distributed to the end of some equipment, Kicker, Septum, and monitoring devices.

We succeeded to generate not only synchronized timing but also beam control information such as beam gate for trigger inhibit signal and injection and extraction timing "value" via data buffer delivery. By using this method, accelerator operation became more convenient system.

INTRODUCTION

SuperKEKB accelerator is electron-positron collider for high-energy physics experiment [1]. Electron and positron are stored in High Energy Ring (HER) and Low Energy Ring (LER) at the energy of 7 GeV and 4 GeV, respectively. To overcome the luminosity of previous KEKB project, a Damping Ring (DR) has constructed to provide low emittance positron to LER. The specification of DR is listed in the Table 1.

Table 1: Key Parameters of Damping Ring

Parameters	Value	Unit
Energy	1.1	GeV
Repetition frequency	50	Hz
Circumference	135.5	m
RF frequency	508.9	MHz
Harmonic Number	230	
No. of bunch trains	2	
No. of bunches / train	2	

The positron beam is accelerated to 1.1 GeV in the injector linac, and delivered to the DR through beam transport line. The DR stores the positron beam during 40 ms which is an essential damping time to make low-emittance. On the other hand, since the maximum injection repetition is 50 Hz, it is necessary to simultaneously control three kinds of states of injection, extraction, and storage. However, since each state is needed to occupy different bucket, different timing information is needed. Therefore, damping ring timing system has two timing receiver for injection control and extraction control.

* hitoshi.sugimura@kek.jp

TIMING CONTROL AT THE DAMPING RING

Overview

The timing signal is generated and controlled at SuperKEKB main timing station at injector linac [2, 3]. The timing is distributed to all over the SuperKEKB accelerator timing receiver with the fiber-optic cables. The DR timing system is its branch of the main timing system. Figure 1 is the picture of DR timing system. It is fabricated in one VME crate, and consists of a controller, a device monitor (RAS), two timing receivers and a time to digital converter (TDC). We use MVME5500 on-board CPU controller and VxWorks-6.8.3 is working on the board. The RAS monitors power voltage and temperature, and is controlled by serial port connection. As the timing receiver, we use Event Receiver (VME-EVR-230RF) developed by Micro Research Finland [4]. The TDC is originally developed to monitor the timing whether it is generating at an appropriate timing [5].

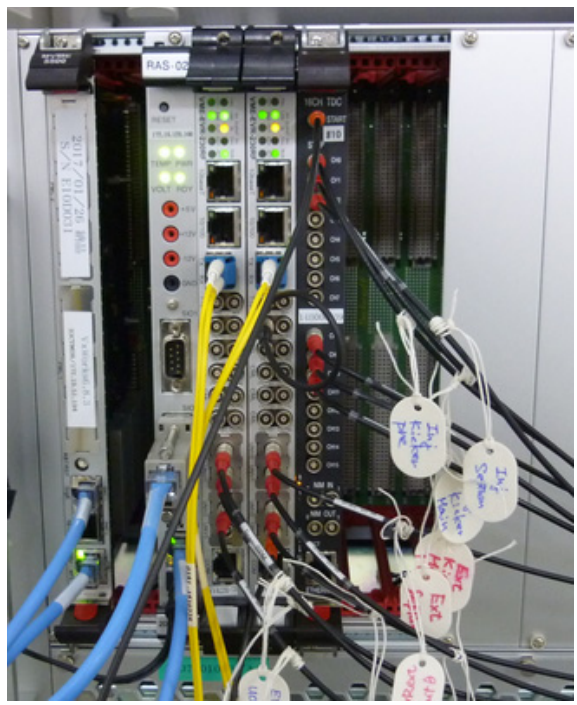


Figure 1: The picture of Damping Ring Timing System.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP17

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

All the generated timings are distributed to TDC and each device.

The timing system receives timing as a digitized 8-bit "event code" with fiber-optic cable, then generate the trigger when suitable event code is received. So, we call the system "event timing system". Application software is developed as EPICS device support [6].

Beam Gate Control

We integrated Beam Gate control system to event timing system by using "distributed bus bit" [7]. The distributed bus bit is one of the function of MRF event timing module and it can be transferred 8 types of bit information simultaneously. By using this function, we can change beam gate status faster than EPICS process and control each device independently. When the beam gate status is changed, the signal is input to Event Generator (EVG). The EVG sends it to EVRs at different timing for each device. EVR change trigger enable or disable status according to distributed bus bit status. This method made it possible to control injection only or extraction only mode. The greatest benefit was that the hardware can be done with no changes.

Pre-Trigger Generation

The kicker magnet of DR is needed to supply not only discharged trigger but also charged trigger, and charging time must be kept at 12 ms. In the operation of 2017, charged trigger for kicker magnet (we call it "pre-trigger") was not able to generate at EVG. Therefore, DR timing system is uniquely set delay timing by receiving appropriate pre-trigger timing as a "delay value" via data buffer. The data buffer function is already equipped in the MRF timing system. It is also used to send bucket number and shot number information and so on. The delay value for pre-trigger is sent before two shot of discharged trigger then set the value. This method could be kept fixed charged time in spite of fluctuated timing of bucket selection. The detail algorithm is described in [8].

Manual Extraction System

In the normal operation mode controlled by bucket selection algorithm, the extraction timing at DR is already calculated before its positron's injection timing. However, after unforeseen abort in the Main Ring happen, extraction cannot be controlled by the bucket selection algorithm. Same situation also happens after dispersion measurement there. To solve this situation, we developed manual extraction system.

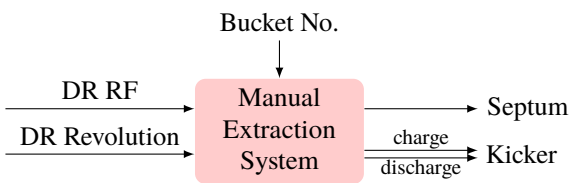


Figure 2: Schematic view of the manual extraction system.

Figure 2 shows schematic view of the manual extraction system. The system counts RF clock of the DR. The frequency is 508.89 MHz which is the same frequency of the Main Ring. Furthermore, the system has input of revolution of the DR. It is the starting point of the bucket number-0. If the charging time is set, the pre-trigger is first issued. Then, after waiting charging time, the main trigger is generated. Since the main trigger timing has to depend on the stored positron bucket number, the system gives delay according to the number. When the extraction bucket number-N is set via epics process, the system counts N clocks from revolution timing, then generate trigger.

SYSTEM OPERATION IN SuperKEKB PHASE-2

SuperKEKB Phase-2 project was started operation from March 2018 and finished July 2018. Before the project, DR standalone operation has been done for a month. We tried to operate beam handling with this timing system. There was no major problem throughout. But to say a little bit, at the beginning of the operation, the charged trigger duration for kicker magnet was fluctuated and sometimes the magnet generates alarm information. This was because the timing calculation is initialized when the operation setting changed. At that time, wrong value of the delay timing was set at EVRs. To solve this problem, we changed output generation to disable mode during operation setting change.

SUMMARY AND OUTLOOK

We developed timing control system at Damping Ring. There are three characteristic control components, the first is we integrated beam gate control to event timing system with distributed bus bit. It became useful function by separating injection and extraction control. Since it became unnecessary additional cable connection, the hardware component became simple structure. Furthermore, beam gate control is able to operate more quickly than EPICS process.

The second is we constructed pre-trigger generation algorithm in the EVRs by using data buffer function of event timing system. The kicker magnet is need 12 ms to charge condenser with precise timing. So, pre-trigger is additionally generated at EVRs. The pre-trigger setting value is sent from EVG, and set delay in previous shot. The system could be kept fixed charging time for Kicker magnet.

The last is we developed manual extraction system. Extraction would be done in the bucket selection calculation in the normal. However, after some optics measurements or occurred irregular abort in Main Ring, manual extraction system manages trigger signal to deliver to each device by setting stored beam bucket number.

The DR operation was successfully finished with no serious event timing problems. Phase-3 project is scheduled to start in spring 2019 to attempt to update the world highest luminosity record and obtain new particle physics events by Belle2 experiment [9]. We will prepare to ensure more stable operation than phase-2 operation.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP17

REFERENCES

[1] Preliminary version of TDR of SuperKEKB, https://kds.kek.jp/indico/event/15914/.

[2] H. Kaji *et al.*, "Upgrade of event timing system at SuperKEKB", in *Proc. ICALEPCS'13*, San Francisco, CA, USA, Oct. 2013, paper THCOCA04, pp. 1453–1456.

[3] H. Kaji *et al.*, "Construction and Commissioning of Event Timing System at SuperKEKB", in *Proc. IPAC'14*, Dresden, Germany, June 2014, pp. 1829–1832. doi:10.18429/JACoW-IPAC2014-TUPRI109

[4] Micro Research Finland (MRF), http://mrf.fi/.

[5] T. Suwada *et al.*, "Wide dynamic range FPGA-based TDC for monitoring a trigger timing distribution system in linear accelerators", *Nucl. Instrum. Meth.* A, vol. 786, pp. 83–90, June 2015. doi:10.1016/j.nima.2015.03.019

[6] MRFIOC2, https://github.com/epics-modules/mrfioc2

[7] H. Kaji *et al.*, "Beam Gate Control System for SuperKEKB", in *Proc. IPAC'18*, Vancouver, BC, Canada, Apr./May 2018, pp. 2124–2127. doi:10.18429/JACoW-IPAC2018-WEPAK015

[8] H. Sugimura *et al.*, "Synchronized Timing and Control System Construction of SuperKEKB Positron Damping Ring", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 229–231. doi:10.18429/JACoW-ICALEPCS2017-TUP094

[9] Belle2, https://www.belle2.org/.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP18

OPERATIONAL EXPERIENCE OF THE DIGITAL LLRF CONTROL SYSTEM AT THE BOOSTER RING OF TAIWAN PHOTON SOURCE

Zong-Kai Liu †, Fu-Yu Chang, Lung-Hai Chang, Mei-Hsia Chang, Shian-Wen Chang, Ling-Jhen Chen, Fu-Tsai Chung, Yi-Da Li, Ming-Chyuan Lin, Chih-Hung Lo, Chaoen Wang, Meng-Shu Yeh and Tsung-Chi Yu, NSRRC, Hsinchu, Taiwan

Abstract

The purpose of a Low-Level Radio Frequency (LLRF) system is to control the accelerating cavity field amplitude and phase. To have better RF field stability, precise control and high noise reduction, a digital LLRF control system based on Field Programmable Gate Arrays (FPGA) was developed at NSRRC. We replaced the analog LLRF system with a digital version for the TPS booster ring at the beginning of 2018. During routine operation of the booster RF, some faults occurred when the digital LLRF operated in the energy savings mode. The performance and operational experience of the digital LLRF for the TPS booster will be presented here.

INTRODUCTION

The Taiwan Photon Source (TPS) at the NSRRC is a third-generation light source operating at 3 GeV. The field in the accelerating cavity is controlled by a Low-Level Radio Frequency (LLRF) system. A digital LLRF (DLLRF) control system based on Field Programmable Gate Arrays (FPGA) is expected to exhibit improved RF field stability, precise control and high noise reduction. The analog LLRF system for the TPS booster ring was replaced by a digital version at the beginning of 2018 [1]. Since the digital tuner loop is not implemented yet, we only replaced the modules in the RF path for the digital system to keep the tuner function and the interlock protection system. The Programmable Logic Controller (PLC), including the analog tuner loop and the function of interlock protection, is still contained in the DLLRF system. Figure 1 shows the architecture and photos of the TPS booster DLLRF system.

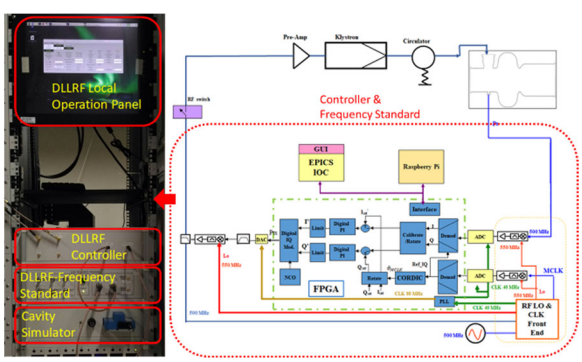


Figure 1: Architecture and photos of the TPS booster DLLRF system.

During full power routine operation with the DLLRF system, the difference between set points and measured values can be controlled to 0.32 ± 0.52 % and 0.01 ± 0.13 degrees for the cavity voltage amplitude and phase, respectively. The sidebands of the 60 Hz noise and its harmonics can be suppressed down to -70 dBc. The details of the design and implementation of the DLLRF controller, the hardware architecture, test results and operation performance of the TPS booster DLLRF system can be found in [1-3].

The operational status can be monitored by the data acquisition system, including the transient data recorder for trip analysis and the archiver for the slow long-term data recorder. Figure 2 shows the hardware structure for the TPS booster RF data acquisition system. All the analog signals are collected on the junction box and distributed to the digitizers. Signal buffers are placed between junction box and digitizers to avoid any impedance mismatch. Four digitizers are used for the EPICS IOC to collect historic data at a slow sampling rate (about 1 kHz) and one digitizer is used for the transient recorder with a sampling rate of up to 250 kHz.

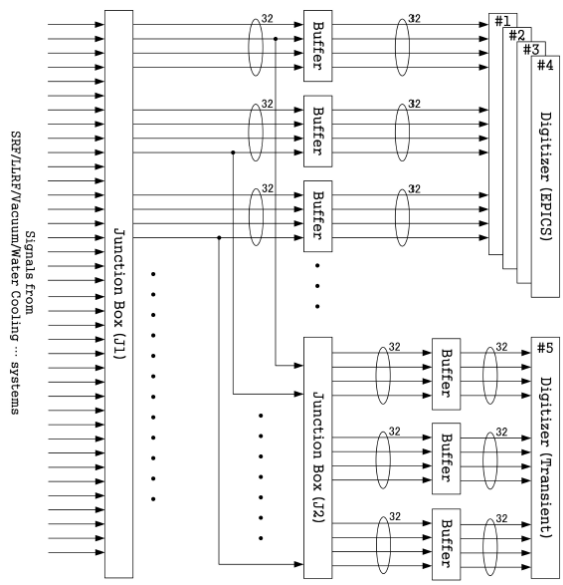


Figure 2: Hardware structure of the data acquisition system.

A function to save energy [4] is implemented in the TPS booster DLLRF by controlling the klystron cathode current and cavity voltage following the injection timing signal [5]. During routine operation, some faults occurred when the DLLRF operated in the energy saving mode. The operational experience of the DLLRF for the TPS booster as well as trip analyses are reported in the following sections.

† liu.zk@nsrrc.org.tw

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP18

OPERATION STATUS

The DLLRF system for the TPS booster was installed on Feburary 2018 and was found to be stable without the function of energy saving. However, when the function of energy saving was turned on, the DLLRF system tripped. The related signals are shown in Fig. 3 for the transient data. The cavity gap voltage was close to saturation at the start of energy ramping. This is due to an unstable oscillation of the cavity gap voltage during the beginning of full klystron power operation. The LLRF status on the PLC would turn to tune mode when the oscillating cavity gap voltage reached its lower limit, and the DLLRF would cut-off the RF forward power due to an inconsistent status.

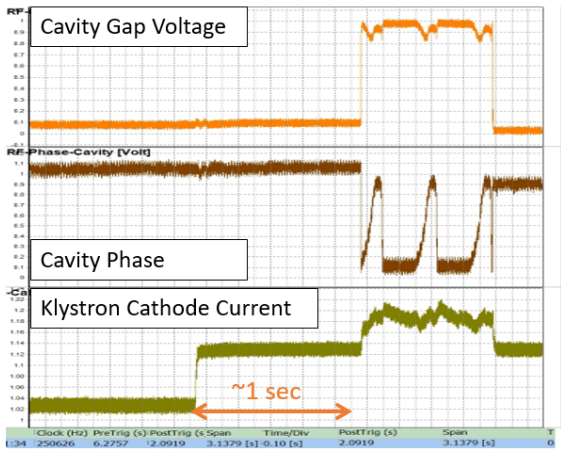


Figure 3: Related signals during the trip event. The y-axis displays the raw reading values in Volt.

This oscillation is due to the phase shift of the RF loop resulting from a changing klystron cathode current, as shown in Fig. 4. The cavity phase shifts by about 60° while the klystron adjusts to full power operation, and back again to 0° after about 6 sec. The PI gain of the digital LLRF controller was optimized for full power operation and not for a phase shift in the loop. After adjusting the PI gain, the TPS booster RF operates stable with the DLLRF system in the energy savings mode. Historical data for stable operation with energy saving are shown in Fig. 5.

Figure 4: Measurement of related signals in tune mode.

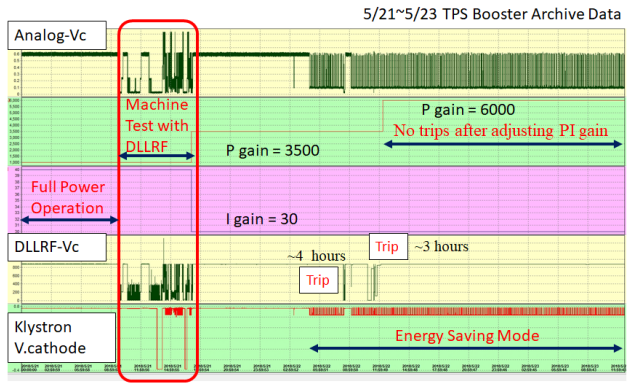


Figure 5: Historical data for stable operation with energy saving. After adjusting the PI gain, no trips occur during energy savings operation.

PI OPTIMIZATION

A simple LLRF system model was developed to optimize the PI gain offline. This simulation program was developed with MATLAB and Fig. 6 shows its block diagram.

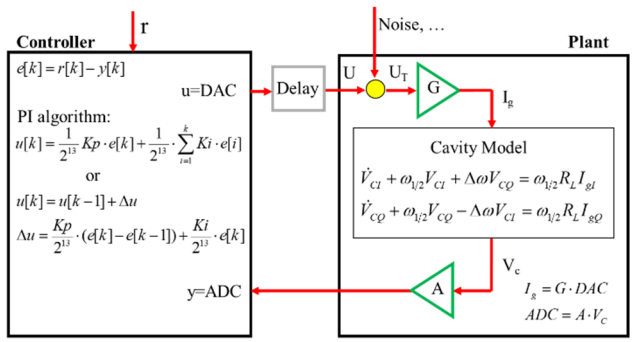


Figure 6: Block diagram of a LLRF system simulation model.

The RF plant model includes the cavity response and the calibration of the DAC to generator current and the cavity voltage to ADC. The equations of the cavity model can be written as follows [6]:

$$\dot{V}_{CI} + \omega_{1/2} V_{CI} + \Delta\omega V_{CQ} = \omega_{1/2} R_L I_{gI}$$
$$\dot{V}_{CQ} + \omega_{1/2} V_{CI} - \Delta\omega V_{CI} = \omega_{1/2} R_L I_{gI} \quad (1)$$

where $\omega_{1/2} = \omega_{cav}/2Q_L$ and $\Delta\omega = \omega_{cav} - \omega_{RF}$ and the calibration coefficients G and A for $I_g = \mathrm{G} \cdot \mathrm{DAC}$ and $\mathrm{ADC} = \mathrm{A} \cdot V_C$ are obtained from measurements. The controller model describes the function of the PI algorithm in the DLLRF controller. For the present design, we use the theoretical form including a 13-bit truncation in the digital implementation, which can be defined as follows:

$$e[k] = r[k] - y[k] \quad (2)$$

$$u[k] = \frac{1}{2^{13}} Kp \cdot e[k] + \frac{1}{2^{13}} \cdot \sum_{i=1}^{k} Ki \cdot e[i] \quad (3)$$

For the older design we use the difference equation algorithm, which is:

$$u[k] = u[k-1] + \frac{Kp}{2^{13}} \cdot (e[k] - e[k-1]) + \frac{Ki}{2^{13}} \cdot e[k] \quad (4)$$

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP18

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

However, errors coming from the 13-bit truncation of the proportional term accumulate in eq. (4) resulting in a poor performance comparison with eq. (3), as shown in Table 1.

Table 1: Analysis of the input signal with different PI algorithms

	Eq. (3)	Eq. (4)
Amplitude SP	905.3 kV	905.3 kV
Amplitude mean	905.09	898.23
Amplitude RMS	± 0.12 %	± 0.19 %
Phase SP	12.1°	12.1°
Phase mean	11.95 °	10.29 °
Phase RMS	± 0.20 °	± 0.21 °

Figure 7 shows the cavity gap voltage signals while the feed forward table was turned on. An additional pulse waveform is added into the RF output. The response of the cavity gap voltage is different with different PI gains and delay time. Therefore, the delay time of the whole RF system can be determined by fitting these waveform signals. The simulation and the measurements are consistent with each other for the 3 μsec delay. The simulation predicts instability for some PI gain values with a 60° loop phase shift, as shown in Fig. 8. This instability is consistent with observation.

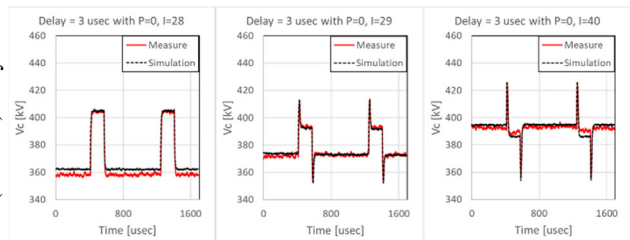


Figure 7: Cavity gap voltage with the feed forward table turned on for different I gain.

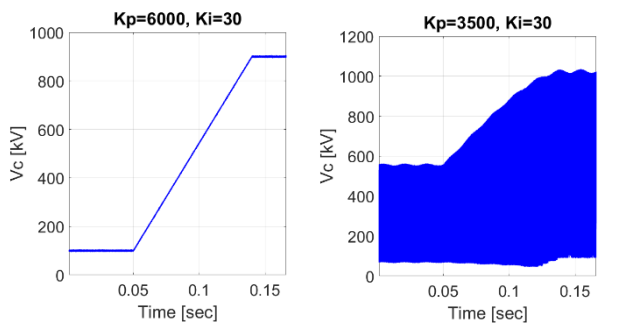


Figure 8: Simulation results for the cavity voltage during ramping with different PI gains. A small P gain may cause the instability.

Figure 9 shows the stable region for PI gains with a 60° loop phase shift obtained from the simulation. The optimized gain is determined by minimizing the RMS errors during ramping in the stable region. The finial gains are set to 7500 and 24 for P and I, respectively. The difference between set points and measured values during ramping with the energy saving function and optimized PI gains can be controlled within 0.05 ± 0.28 % and 0.14 ± 0.21 degrees for the cavity voltage amplitude and phase, respectively. With tuned PI gains, the TPS booster digital LLRF system operates stable with active energy savings function.

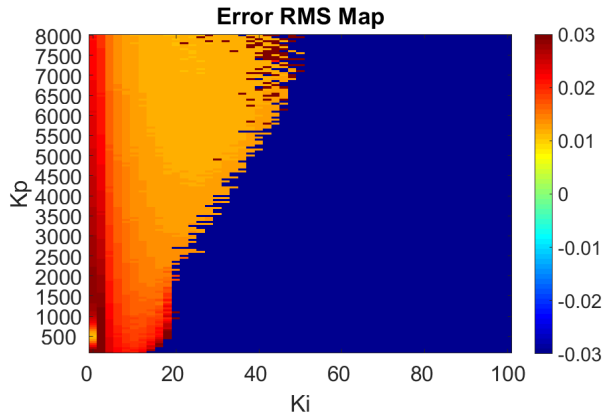


Figure 9: The map of RMS values for the errors between set points and measurements. Negative values indicate an unstable region.

CONCLUSION

The analog LLRF system for the TPS booster ring was replaced by a DLLRF system in February 2018. A function to save energy by controlling the klystron cathode current and cavity voltage with the injection signal, is implemented as well. It may cause instability due to the phase shift in the RF loop resulting from the change of the klystron cathode current. The PI gains need to be re-optimized for the energy savings mode to obtain stable operation. A simple LLRF system model was developed to optimize the PI gains offline. The instability predicted by the simulation is consistent with observations and the optimized PI gains are set to 7500 and 24 for P and I, respectively.

REFERENCES

[1] Z. K. Liu *et al.*, "Digital Low Level Radio Frequency System for the Booster Ring of the Taiwan Photon Source", *in Proc. IPAC'18*, Vancouver, Canada, April 29–May 4, 2018, paper WEPAL054.

[2] F. Y. Chang *et al.*, "Digital Low Level RF Control System for the Taiwan Photon Source", *in Proc. IPAC'17*, Copenhagen, Denmark, May 2017, paper THPAB148.

[3] Z. K. Liu *et al.*, "Input Output Controller of Digital Low Level RF System in NSRRC", *in Proc. IPAC'17*, Copenhagen, Denmark, May 2017, paper THPAB150.

[4] F. T. Chung *et al.*, "The Develop Energy Saving Device for RF System at Taiwan Photon Source,", *in Proc. IPAC'18*, Vancouver, Canada, April 2018, paper THPAB150.

[5] M. S. Yeh *et al.*, "Energy-efficient Operation of a Booster RF System for Taiwan Light Source Operated in Top-Up mode," *Nucl. Instr. Meth A*, vol. 755, pp. 46-53, 2015.

[6] J. Vincent *et al.*, "On active disturbance rejection based control design for superconducting RF cavities," *Nucl. Instr. Meth A*, vol. 643, pp.11-16, 2011.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP19

TIMING SYSTEM FOR MULTIPLE ACCELERATOR RINGS AT KEK e+/e- INJECTOR LINAC

F. Miyahara†, K. Furukawa, H. Kaji, M. Satoh, H. Sugimura, KEK, Ibaraki, Japan
T. Kudou, S. Kusano, Mitsubishi Electric System & Service Co., Ltd, Ibaraki, Japan
H. Saotome, Kanto Information Service, Ibaraki, Japan

Abstract

The KEK e+/e- injector linac is operated in multiple beam modes that can be switched every 20 ms for e+/e- beam injection to five different circular accelerators, SuperKEKB High Energy Ring (HER), Photon Factory (PF) ring, PF-AR, positron damping ring (DR) and SuperKEKB Low Energy Ring (LER). Because those accelerators have different ring energies and radio frequencies, the linac is required to have a timing system which supports multiple modes and synchronization system to the circular accelerator. Those systems to fulfil multiple injections to independent circular accelerators are described in this paper.

INJECTOR LINAC

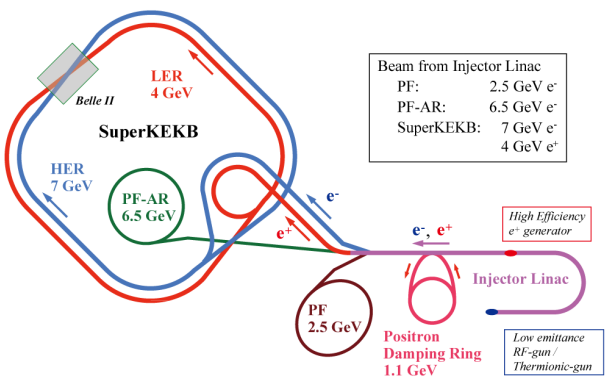


Figure 1: Schematic view of the e+/e- injector linac and accelerator rings at the KEK.

The KEK e+/e- injector linac and five different rings are depicted schematically in Fig. 1. The HER requires low emittance electron beam, thus the photocathode rf-gun [1] is used for HER beam injection. High charge (~10 nC) electron bunches for positron production and electron beams for PF, PF-AR are given by the thermionic gun. The linac consists of 8 sectors (A-C, 1-5) which have 8 high power RF units except for injector (rf-gun, thermionic gun and bunching section) and positron production section. The RF unit provides RF power to 4 accelerating structures. The beam energy at the end of the linac is controlled by RF phase of the unit. In order to set optimum beam optics for each the beam mode, 28 pulsed quadrupoles and 36 steering pulsed magnets were installed in last year [2]. Parameters such as accelerating RF phases and the currents pulsed magnets can be changed every beam pulse by employing the event timing system.

TIMING SYTEM OF THE LINAC

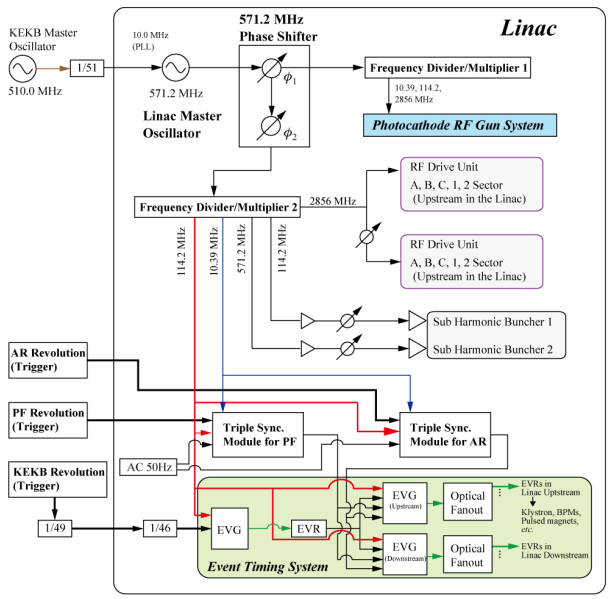


Figure 2: Timing system of the linac.

Figure 2 shows the linac timing system with the fundamental RF signals. The master frequency of 571.2 MHz is generated with a signal generator which is synchronized with the SuperKEKB oscillator via a 10 MHz reference signal [3]. The 571.2 MHz phase shifter set the linac whole RF phase according to the beam mode. The phase of ϕ_1 and ϕ_2 correspond to the HER and the LER injection phase, respectively. In case of the PF and the PF-AR injection, the ϕ_2 is set to zero. Because large phase shift accompanying beam mode change makes the laser system for the rf-gun unstable, the laser system always synchronized with the HER.

The trigger signals to various components are distributed by the MRF event system [4] which consists of event generators (EVG) and event receivers (EVR). The event timing system has three EVGs [5]. The first EVG-EVR set makes a trigger timing signal for the HER/LER injection. The trigger timing signal for the PF/PF-AR injection is sent from a special module to make a synchronized signal with the circular accelerator. The EPICS IOC program for the event system selects a corresponding trigger for 2nd EVGs of the beam mode. Then, the 2nd EVGs distribute an event code which corresponds to the beam

† fusashi.miyahara@kek.jp

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP19

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

mode and a data buffer including some parameters such as a the HER/HER/PDR injection RF bucket number, setting current of pulsed magnets. Because the external RF clock used in the event system depends on the beam mode, the phase of the event system could be varied every pulse.

BEAM INJECTION TO A CIRCULAR ACCELERATOR WITH DIFERENT RF

Synchronization System

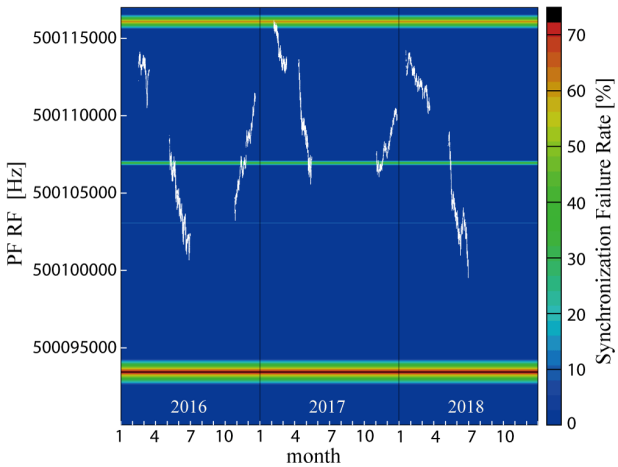


Figure 3: Variation of the PF radio frequency and the synchronization failure rate. White dots show the PF RF in the past 3 years. The colour band represents the synchronization failure rate at the frequency which was estimated by a simulation in a condition of synchronization width of 500 ps and 114.237423 MHz of the linac RF.

Since the event system uses the 114.2 MHz clock which is generated by the linac RF system that uses common sub-harmonic frequency of the 10.39 MHz, the linac beam timing synchronized with HER/LER revolution timing periodically. However, the PF and the PF-AR are not synchronized to the linac because the two accelerators use independent RFs. In that case, to generate the RF for the linac from the RF of the circular accelerator [6] in every pulse is one of the ways. We plan to use the rf-gun for the beam injection to the PF/PF-AR. In that case it is difficult to adopt the way in our system because the laser system requires stable radio frequencies. Thus, we take the way to use chance synchronization of the linac RF and the PF/PF-AR revolution frequency. The triple sync module shown in the Fig. 2 outputs a signal to the event system on a timing when the linac RF (114.2 and 10.39 MHz) is synchronized with a revolution timing of the circular accelerator every 20 ms. The width of synchronization (Γ_{syc}) can be changed. Because the synchronization probability depends on the linac and the circular accelerator RFs, there could be no chance event in a certain period of time. In order to operate the klystron stably, a regular interval of the trigger (20±2 ms in our case) is required. In case of no synchronized event in a waiting time (~0.9 ms) from the 50 Hz trigger input, the module send a trigger output to the event system and a veto signal to the thermionic gun. Figure 3 shows the variation of the PF RF and the synchronization failure rate. The annual variation of the PF RF is about 15 kHz owing to the circumference compensation of the ring. The RFs and variations in a day/year are summarized in Table 1. A very low synchronization probability region is found around 500.093 MHz, but there is no such a region in the operating frequency. We can enlarge the probability of the synchronization by increasing the Γ_{syc}, but it is trade-off for the beam injection phase jitter.

Table1: RF and Its Variation in a Year/Day

Ring Accelerator	Frequency	Annual Variation	Variation in a day
SuperKEKB	508.88 MHz	~300 Hz	~10 Hz
PF	500.11 MHz	~15 KHz	~1 kHz
PF-AR	508.57 MHz	~1.5 kHz	~200 Hz

Resynchronized Delay

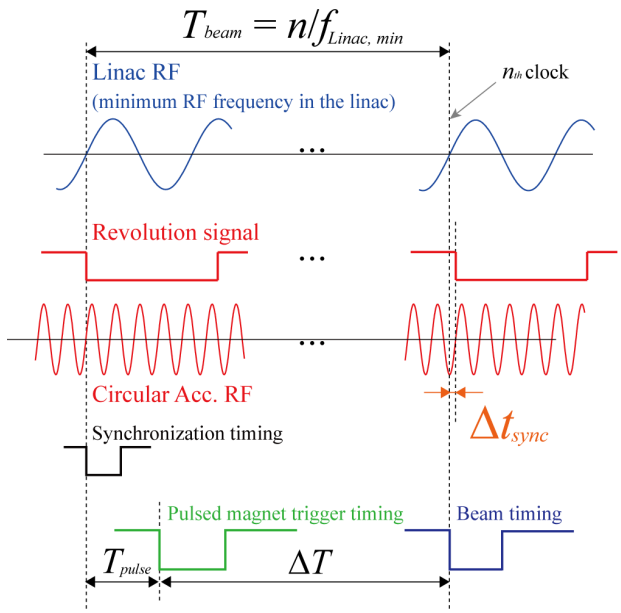


Figure 4: Layout of papers.

To keep reproducibility of the current to the pulsed magnet, the power supply of the magnet requires precise trigger timing which is 3 ms before the beam arrival time. Then, the trigger should be the first timing in the timing system and it is generated by the chance synchronization in case of PF/PF-AR beam injection. Because the RF used in the event system is different from the PF/PF-AR one, a simple 3 ms delayed beam trigger leads to injection to outside the separatrix or an unexpected RF bucket. Thus, we have developed the way finding a synchronized timing after arbitrary delay from the first synchronized timing. Figure 4 shows the relation of trigger signals (the output of the triple sync module, the pulsed magnet, the beam timing), the RF of the linac and the revolution signal from

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP19

the circular accelerator. If we can know the resynchronized delay longer than the required interval (3 ms), the beam timing is synchronized with the revolution timing and the interval can be fixed. The delay T_{beam} which has a jitter less than Δt_{sync} is given by Eq. (1),

$$\left| n\frac{f_{Circ,\mathrm{Rev}}}{f_{Linac,\min}} - \mathrm{int}\left(n\frac{f_{Circ,\mathrm{Rev}}}{f_{Linac,\min}} \right) \right| < f_{Circ,\mathrm{Rev}}\Delta t_{sync} \quad , (1)$$

where, n is the number of clock cycles, int(x) stands for the integer part of the x. The $f_{linac,min}$ and the $f_{Circ,Rev}$ are the minimum fundamental frequency used in the linac and the revolution frequency of the circular accelerator. In the operation, the 114.2 MHz of the linac RF and the 500.11 MHz of the PF RF (508.57 MHz of the PF-AR), are monitored by using the same frequency counter with the resolution of 10 digits per second. The delay and the synchronization probability are calculated every second to follow the changing of the RFs. We can easily find the n, and the timing control is implemented by EPICS. The maximum injection timing jitter is given by $\Gamma syc+ \Delta t_{sync,}$ and the jitter is less than 0.6 ms for the beam injection to the PF. The system using the way has been working since autumn of 2017 without any trouble.

REFERENCES

[1] R. Zhang *et al*., "Yb/Nd Doped Hybrid Solid Laser of RF Gun and Beam Commissioning for Phase-II of SuperKEKB", in *Proc. IPAC'18*, Vancouver, Canada, Apr, 2018, pp.304-306.

[2] Y. Enomoto *et al*., "Pulse-to-pulse Beam Modulation for 4 Storage Rings with 64 Pulsed Magnets", presented at LINAC18, Beijing, China, Sep. 2018, paper MOPO007, unpublished.

[3] T. Matsumoto *et al*., "Low-level RF System for the SuperKEKB injector Linac", in *Proc. IPAC'18*, Vancouver, Canada, Apr, 2018, pp.2131-3133.

[4] Micro-Research Finland Oy, http://www.mrf.fi/

[5] H. Kaji *et al*., "Construction and Commissioning of Event Timing System at SuperKEKB", in *Proc. IPAC'14*, Dresden, Germany, June 2014, pp.1829-1832.

[6] Y. Kawashima, T. Asaka and T. Takashima, "New synchronization method of arbitrary different radio frequencies in accelerators", *Phys. Rev. ST Accel. Beams*, vol. 4, p. 082001, Aug. 2001, doi:10.1103/PhysRevSTAB.4.082001

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP20

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

STUDY OF ENERGY SAVING OPERATION FOR THE TLS BOOSTER POWER SUPPLY

Jenny Chen, Demi Lee, C.Y. Wu, Y. S. Chen, C.Y. Liao, K.H. Hu, K. T. Hsu
National Synchrotron Radiation Research Center, Hsinchu 30076, Taiwan

Abstract

Operating an injector of a synchrotron light source, energy efficiency is an important issue. Dipole and quadrupoles families of the booster synchrotron for Taiwan Light Source (TLS) is resonantly excited by three White circuits at 10 Hz rate. Magnet current cannot response in cycle-by-cycle basic due to resonance nature. The possibility of operation of the booster synchrotron in energy saving mode is explored. Minimizing the duration of magnet excitation without effect of the injected and extracted beam to support top-up operation for the TLS is investigated. Efforts will be presented in this report.

INTRODUCTION

The storage ring of TLS is operated in top-up injection mode at stored beam current 361 mA. Typical beam lifetime is about 6~7 hours, it needs to refill beam at every minute. The booster synchrotron is turned on continually since the top-up injection started from 2005. Energy saving operation of the booster RF system was investigated and activated [1, 2]. The possible operation scenario for the magnets excitation system is studied recently.

There are three families of magnet driven by a White circuit [3, 4, 5, 6, 7]. Each White circuit consists of a DC and an AC power supply in the TLS booster synchrotron. The DC power supply is turned on firstly before AC power supply to protect the polarized electrolyte capacitor. The control sequence is that after DC power supply achieves its nominal value then AC power supply can be enabled. The three families consisted of dipole, focusing quadrupole and defocusing quadrupole magnets. After three families of power supply turned on respectively, the phase regulation algorithm is applied to ensure the relationship between three White circuits to be in phase. The time for turning on DC, AC power supplies and applying phase regulation needs to be ended before beam trigger signal. To proceed the energy-saving of the booster synchrotron, the excitation is removed in reverse order, phase regulation stopped firstly, AC power supply turned off then DC power supply turned off at last. Minimizing the time duration during above cycle will save more electricity.

WHITE CIRCUITS OF THE BOOSTER SYNCHROTRON

There are three families of magnet systems in the booster synchrotron. These are dipole, focusing quadrupole and defocusing quadrupole magnets. Each family includes 12 magnets. The main excitation circuits are configured as three independent White circuits and resonantly excited at 10 Hz. The White circuit comprises two coupled resonant circuits, which are a bypass capacitor and a DC power supply with two resonant LC circuits. Each White circuit has the same configuration; those AC and DC power supplies work independently. Both DC and AC current are controlled within an error less than 5×10^{-4}. The tune variation during energy ramping should be in the order of 0.01 for stable operation which corresponding to 5×10^{-4} focusing error. This implies that the current of DC and AC components for quadrupole magnets must be stabilized within the same order. The amplitude and phase might be drifted which caused by resonance frequency change slightly due to ambient temperature and heating of the capacitor of White circuit. It takes several hours to achieve thermal equilibrium. The tolerable phase drift was estimated to be 0.1^{o} corresponding to 30 μsec for 10 Hz normal operation.

The controls system needs to provide adjustable precision for amplitude and phase of 10 Hz sinusoid reference for AC power supplies of three White circuits. The block diagram of White circuit power supplies interface is illustrated in Fig. 1. The White circuit power supply control interface consists of amplitude and phase detector module, high purity 10 Hz generator and amplitude regulator module, digital delay generator (DDG), time to digital converter (TDC), interlock and protection module, 16 bits DAC and ADC module. The amplitude/phase detection module measures the peak of magnet current for the analog PID amplitude regulator of high purity 10 Hz sine-wave generator.

The major function of the White circuit power supply control includes provide 10 Hz reference signal to drive power supply and regulate amplitude and phase of the AC current of magnets [4]. The amplitude regulation loop purpose is to keep magnet current with constant amplitude. The DAC module sets the amplitude reference for the DC/AC power supply, the high purity 10 Hz sine-wave generator module is based on the amplitude reference to generate 10 Hz sine-wave for AC power supplies to control the amplitude of White circuit. Purposes of the phase regulation loop as shown in Fig. 2 is to keep constant phase difference between two quadruple families with dipole.

The operation sequence of the White circuits, three DDG module sets the phase of a 10 Hz trigger signal for three high purity 10 Hz sine-wave generator modules that produces correlated phase of the three 10 Hz sine-wave output. The DCCT device senses the output amplitude of magnet current for the amplitude/phase detection module that directly detects the output phase of magnet current then feedbacks to TDC module for the software, PID phase regulation, to rectify the phase of DDG module.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP20

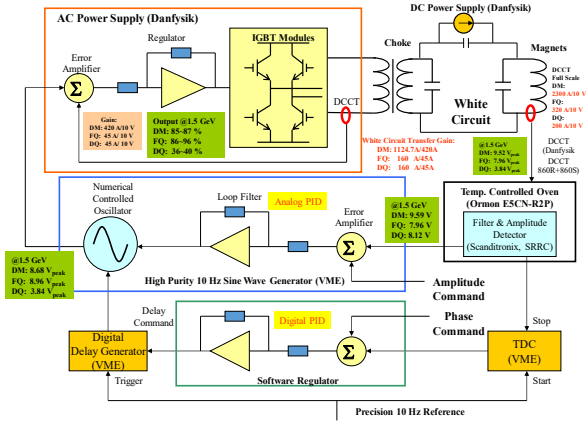


Figure 1: Booster synchrotron AC power supply control interface.

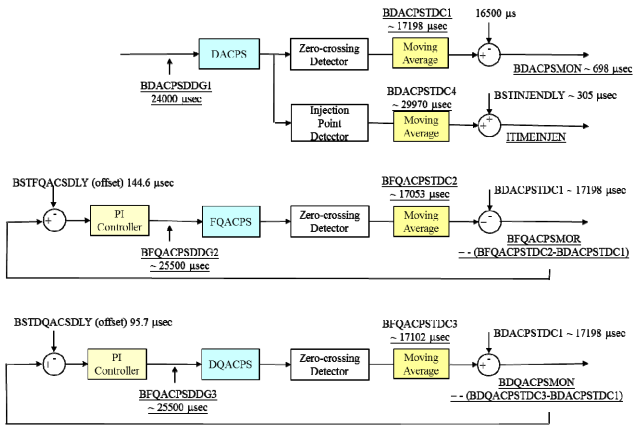
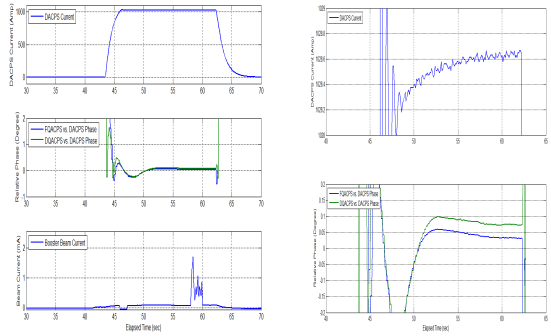


Figure 2: Magnet excitation phasing loops.

In the phase regulation loop, a zero-crossing signals are connected to the time digital converter to measure the zero-crossing timing (phase) of the dipole AC current. The respective zero-crossing timings of the focusing quadrupole (FQ) and defocusing quadrupole (DQ) AC power supply are relative to the dipole current's zero-crossing and are regulated by the phase PI controller that adjusts the phase of 10 Hz driving signal of the FQ and DQ AC power supply. A digital PI regulator is used to keep the phase difference constant between two quadruple families and dipole family. Input of the regulation loop is from TDC and output is to a digital delay generator. Figure 3 (a) shows the amplitude of dipole AC current and the phase difference between FQ/DQ AC current and amplitude and phase regulation loops enabled. Zoom of the amplitude and phase response for this disable/enable operation of power supply is shown in Fig. 3 (b). If the amplitude and phase achieved a certain level the beam can be ramped in booster ring even the amplitude or phase is damped out of the transient response region of PI regulation. The PI regulator minimizes the effect of slow drift to support the working conditions for beam injection and ramping energy in the booster synchrotron. It takes about 10 seconds to regulate the relative phase within 0.1° (~30 µsec) and amplitude within 10^{-4} of the nominal current.



(a) (b)

Figure 3: (a) Dipole AC current amplitude (upper), the respective phase difference of FQ/DQ (middle); booster beam current (bottom). (b) Dipole AC current amplitude and phase difference between dipole and FQ/DQ AC current when regulation loop is in action.

SCHEME FOR ENERGY SAVING OPERATION

Excitation of power supplies is applied while injecting beam and removed after to save energy. The beam is injected every minute when top-up mode is activated. It takes about a few injection cycles (10 cycles/sec) to refill to the storage ring. It takes few seconds after nominal value set of the AC/DC power supplies to ensure amplitude and phase tracking stable enough. The time duration required depends on time constant of loads (White circuits), control rules and small resonance frequency drift due to thermal effect. After injection, power supplies are set to zero for save of electricity. In another words, it needs to enable and disable the current output of AC/DC power supplies to operate in energy-saving mode. Operation flow chart is shown in Fig. 4.

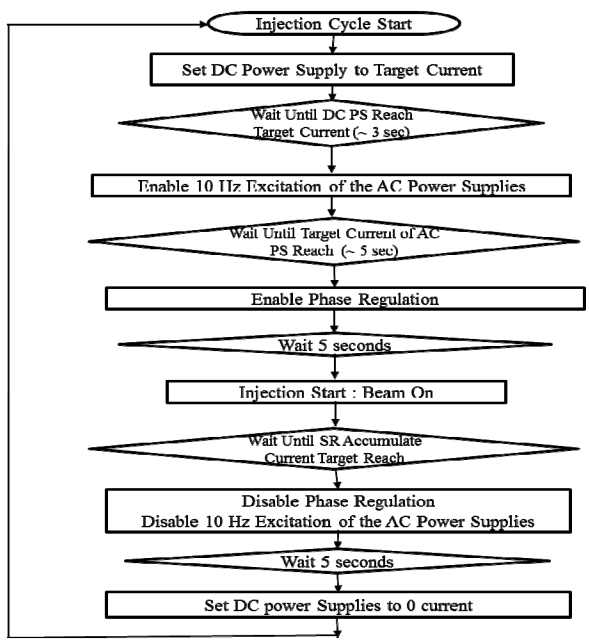


Figure 4: The energy saving operation flow chart.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP20

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

A preliminary test results for On/Off operation of the White circuits with beam is shown in Fig. 5. After optimizing parameters of the regulation loops, the response time is shortened to the minimum period as shown in Fig. 6. Typical continue running in energy saving mode is shown in Fig. 7.

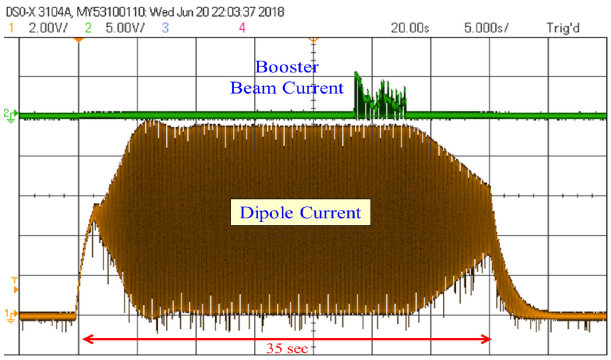


Figure 5: Preliminary test in June for On/Off operation of magnets excitation and booster beam current.

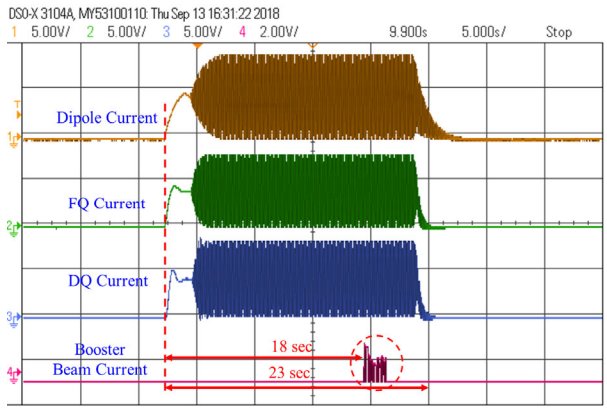


Figure 6: Amplitude and phase regulation loops and beam current after improved in September 2018. All power supplies enabled 18 seconds before beam on for warmup and regulation of amplitude and phase.

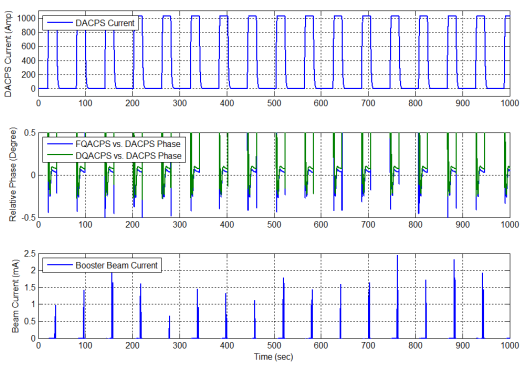


Figure 7: The dipole current, respective FQ/DQ phase difference and booster beam current trend.

Electricity consumption to drive White circuits is 275 kW as shown in Fig. 8. The annually electricity consumption required for the booster is about 1.5125 MWh (275 kW * 5,500 Hours) which is based on 5,500 hours continuous operation. The power supplies operate with 33% duty cycle to support top-up injection every 60 seconds which is achieved at this moment. Saved electricity is more than 1 MWh annually. Estimated cost saving is more than 70 thousand USD per year based on averaged electricity price USD 0.07/kWh in Taiwan.

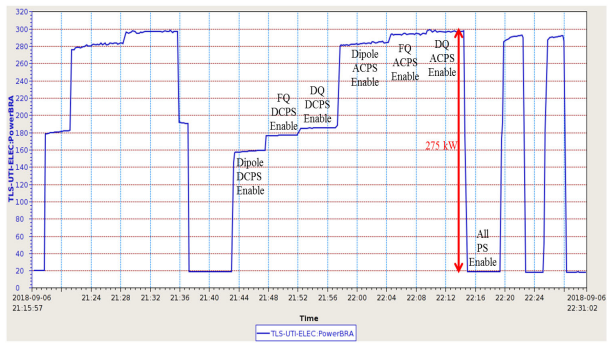


Figure 8: Electricity consumption of the booster main power supplies.

SUMMARY

Energy saving scheme for the booster synchrotron with resonance magnet excitation to support top-up injection for TLS was tested. Further reduce time necessary to reach the conditions for stable beam delivery after power supply enabled need improve response time of the AC power supply amplitude and phase regulation. Various limitation to operate the booster synchrotron in energy saving mode is explored. The magnets excitation can be removed about 40 seconds within a 60-second between consecutive injections. About two-third of the electricity can save in current operation mode without sacrificing its operation performance.

REFERENCES

[1] M.S. Yeh *et al.*, "Energy Saving Controller for the TLS Booster RF System", in *Proc. PAC'09*, Vancouver, BC, Canada, May 2009, paper WE5PFP090.

[2] M.S. Yeh et al., "Energy-efficient operation of a booster RF system for Taiwan light Source operated in top-up mode", *Nucl. Instr. Meth* A775 (2015) 46.

[3] M.G. White *et al.*, CERN-Symposium (1956), 525-529.

[4] K.H. Hu *et al.*, "Control Interface of New White Circuit for SRRC 1.5GeV Booster Synchrotron", in *Proc. EPAC 2000*, Vienna, Austria, Jun. 2000, paper TUP1B19.

[5] C.S. Chen *et al.* "Performance of the New White Circuit in Upgraded 1.5 GeV Booster Synchrotron", in *Proc. EPAC 2000*, Vienna, Austria, Jun. 2000, paper TUP1B18.

[6] C. S. Chen *et al.*, "Study of the White Circuit Tracking Performance in the Booster Synchrotron of SRRC", in *Proc. PAC'01*, Chicago, USA, paper FPAH301.

[7] C.J. Wang *et al.* "Operation Performance of the White Circuit for the Booster Synchrotron in NSRRC", in *Proc. APAC'04*, Gyeongju, Korea, Mar. 2004, paper THP16047.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP21

DEVELOPMENT OF TRIGGERED SCALER TO DETECT MISS-TRIGGER

N. Kamikubota†, K. C. Sato, J-PARC-Center, KEK&JAEA, Tokai-mura, Ibaraki, Japan
Y. Tajima, S. Yoshida, Kanto Information Service, Tsuchiura, Ibaraki, Japan

Abstract

A "triggered scaler" has been developed for J-PARC accelerators. It is a PLC-type scaler with memory-buffers. Number of pulsed signals is counted and stored in a cell of memory-buffer, then, each external trigger (25 Hz) shifts the pointer to the cell. The buffer size (192) is designed to store one machine-cycle (2480 ms or 5200 ms in J-PARC). Demonstrative measurements using a prototype module are reported. In addition, scheme to detect miss-trigger events are discussed.

INTRODUCTION

J-PARC (Japan Proton Accelerator Research Complex) is a high-intensity proton accelerator complex. It consists of three accelerators: a) 400-MeV Linac (LI), b) 3-GeV Rapid Cycling Synchrotron (RCS), c) 30-GeV Main Ring (MR) [1-2]. Since the initial beam in 2006, beam powers for experimental facilities were steadily increased [3].

There are two time cycles in J-PARC. The rapid cycle, 25 Hz, is used at LI and RCS. The slow cycle is used at MR. In 2018, when MR delivers proton beams to Neutrino Facility (Hadron Facility), 2480 ms (5200 ms) is used, respectively. Typical timing scheme is shown in Fig. 1. Since the slow cycle determines the overall time behaviour of accelerators, it is called “Machine Cycle”.

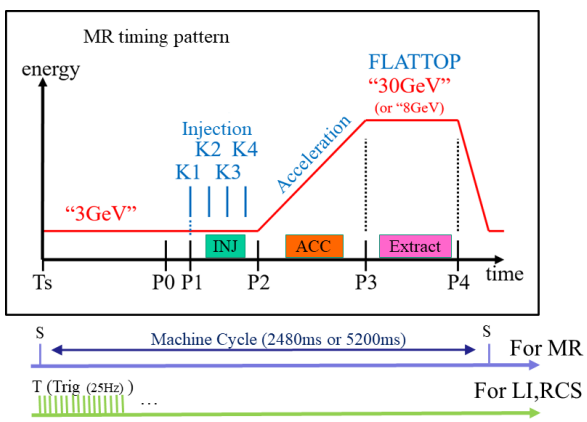


Figure 1: Typical timing scheme of J-PARC accelerators.

TIMING SYSTEM AND MISS-TRIGGERS

Overview of the Timing System

The control system for J-PARC accelerators was developed using the EPICS (Experimental Physics and Industrial Control System) toolkit [4-5]. Addition to it, we have a dedicated timing system [6].

The timing system consists of one send module and several receiver modules. Both modules were developed as home-design VME modules. Event-codes, called “types”, generated by the send module are distributed to receiver modules. A fiber-optic cable network is used for “type” distribution, using several O/E (or E/O) modules. According to the received “type”, each receiver module generates eight independent trigger signals for accelerator components. There are 118/43/45 receiver modules used in LI/RCS/MR, respectively.

Miss-trigger Troubles in J-PARC

During the operation since 2006, we have experienced a few miss-trigger troubles. Here we show three cases.

The first case started as a fault of MR kickers in February, 2014. Soon we found other MR components also miss-behaved occasionally. Finally we found that missing of the 25-Hz trigger-clock occurred in the whole MR area at the rate ~10 times per day at maximum. In May, 2014, we replaced an O/E module, which is located at the root position of MR area.

The second case appeared as a bad quality beam during stable beam delivery to Hadron Facility in November, 2016. Here “bad quality” means that beam size was slightly larger than normal, but still acceptable for experiments. Such beams appeared a few times per month [7]. Some O/E modules were replaced, but not effective. In May, 2017, we found that a receiver module for one of MR steering magnets showed momentary errors, hence, miss-triggers occurred. The timestamps of errors were retrieved from the archive system, which agreed with the timestamps of bad beams. Later survey showed that the errors were caused by external common-mode noises. We added ferrite cores to metal cables connected to the receiver module.

The third case was happen in November to December, 2017. An O/E module, which was used to send the 25-Hz trigger-clock signals from RCS to MR, started to produce fake signals (Fig. 2). Since fake signals affected a critical beam diagnostic system, accelerator operation suspended a few to 10 times in a day. We replaced the O/E module.

In the first and the third cases, it was difficult to find the troublesome module among many suspicious modules. In the second case, miss-trigger events were very rare, and it took six months to find the origin. Such experiences suggested us to develop a new module to detect a miss-trigger.

†norihiko.kamikubota@kek.jp

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP21

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

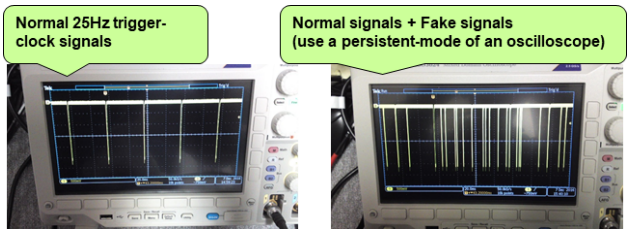


Figure 2: Observed fake signals.

TRIGGERED SCALER

Design of the Module

The conceptual design of the triggered scaler is shown in Fig. 3. Blue part is the module, and external input signals are also shown. The module receives "S" (start of the machine cycle) and "Trig" (start of the rapid cycle, 25 Hz) signals as reference signals, which are generated and distributed by the J-PARC timing system. The module has four input channels, and each of them has dual memory-buffers (16 bit x 192 cells x 2).

In principle, it works as a scaler module. The FPGA logic inside (FPGA_1) writes counts to memory-buffers. When "S" comes, FPGA_1 starts a counting process. It writes a value to the first cell of the first memory-buffer. Each time "Trig" comes, FPGA_1 shifts the pointer to the cell. When next "S" comes, the pointer is moved to the first cell of the second memory-buffer. This scheme allows us to retrieve the counts of input signals during the last machine cycle. In operation with a machine cycle of 2480 ms (5200 ms), 62 (130) of 192 cells are used, respectively. The logic in FPGA_2 reads the memory-buffers, detects a miss-trigger (or verifies no miss-trigger) in them, and sets error flags if necessary.

Development of the triggered scaler was carried out by a company, Hitachi Zosen [8]. By 2016, four prototype pieces were made as Yokogawa PLC-type modules, since it is a standard I/O form in J-PARC MR [9].

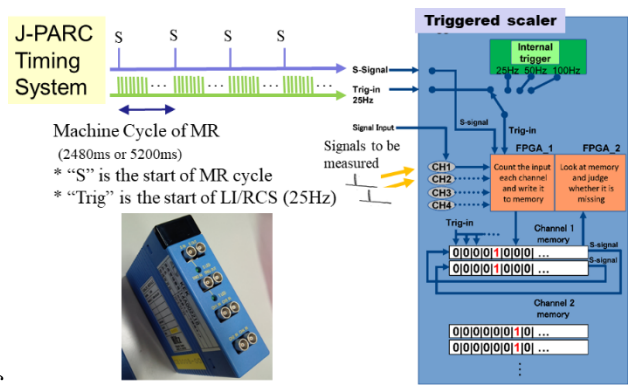


Figure 3: Design of the triggered scaler.

Measurements using a Prototype Module

For the first field test, we prepared a test setup (Fig.4). It consists of a prototype module and a CPU module, where EPICS is running. In January, 2018, we measured a trigger signal for MR injection kickers. In normal operation, there are four successive injections (so-called "K1-K4") in a machine cycle. The observed data showed that four successive "1" values appear in the memory-buffer (Fig. 4).

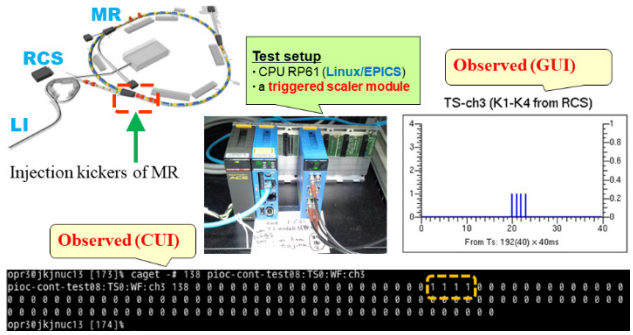


Figure 4: Measurement of a trigger for injection kickers.

As a next demonstration, we measured a RF signal (MR-ring circulation signal), which is generated by an LLRF system. Every machine cycle, 3-GeV protons are injected into MR, then accelerated up to 30-GeV. In Fig. 5, the graph reflects frequency shift of the RF signal, during injection and acceleration periods.

The number of counts in a 40-ms bin were summarized. When beam energy is 3 GeV (8 GeV, 30 GeV), 7429 (7608-7609, 7648) was observed, respectively. Observed counts agree well with the RF parameters (Fig. 6).

Measurements above show that a triggered scaler works as we expected. It is worth noting that a triggered scaler can be a tool to visualize various trigger signals.

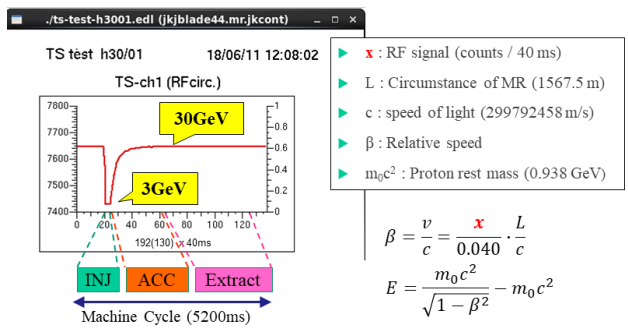


Figure 5: Measurement of a RF signal.

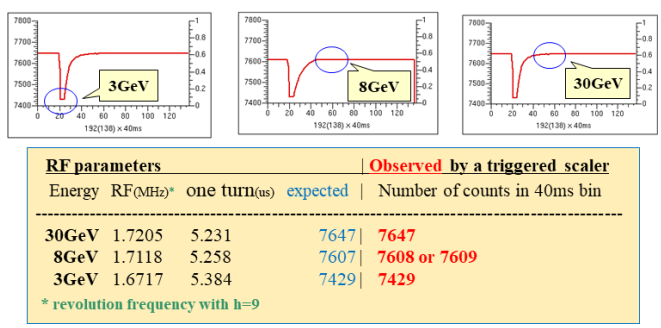


Figure 6: Numbers of counts in 40-ms bin observed at the three different energies.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP21

TO DETECT MISS-TRIGGERS

Three Cases of Miss-triggers

Based on our experiences, there cases of miss-trigger events are considered. Here let us consider the injection kicker signal (K1-K4) as an input signal. The normal view of K1-K4 signal is shown as the reference in Fig. 4. The first case, "miss trigger", is that one trigger (or more) disappears. The second case, "irregular trigger", is that unexpected triggers are overlapped into the original K1-K4 signal. The third one, "double trigger", is that one trigger (or more) is counted double. This could occur when a cable termination is incorrect. Three types of miss-trigger events are shown in Fig. 7.

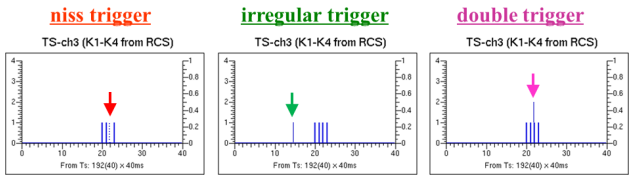


Figure 7: Three cases of miss-trigger events.

EPICS Databases for Triggered Scaler

For easier handling, registers and memory-buffers of the module (hardware layer) were defined in EPICS databases. Part of definitions are given in Fig. 8.

We are developing two more layers: over the hardware layer, (a) operation layers, which describe specific parts for rapid-cycle and for slow-cycle, and over the operation layer, (b) application layers, which correspond to various applications (K1-K4 observation, surveillance of RF signals, miss-trigger detection, etc.). Details are given elsewhere [10].

PV name	Data type	Reg. Address	Description
pageNow	longin	9	Page number, 0 or 1, which point to the active memory buffer
triggerNow	longin	10	Position of a memory buffer, 0 to 191, where a counter value is written
pageSet	longout	11	Switch a page number, 0 or 1, which memory buffer to be read from CPU
errStatus	longin	14	Error status in counting process
triggerInCycle	longin	16	Number of triggers in the previous S-signal period
wf:ch1~ch4	waveform	33-224 289-480 545-736 801-992	Array data of memory buffer (192) for ch1~ch4

Figure 8: Hardware layer definitions of triggered scaler.

Software to Detect Miss-triggers

In stable beam-delivery operations, a miss trigger can occur as a momentary event. There are two issues what we have to discuss.

The first issue is methods to detect a miss-trigger event. For a rapid-cycle mode, an embedded logic in the FPGA_2, which detects a non-zero value in a cell, would be used. For a slow-cycle mode, comparing the latest memory-buffer with a reference is essential. For the latter mode, it often required machine parameters to judge whether a miss-trigger event occurs or not. Thus, combination of EPICS databases and sequencers would be used.

The second issue is what we have to do after a miss-trigger is detected. Studies of routines to save related parameters (shot-number, DCCT waveform, etc.) are in progress.

CONCLUSION

The "triggered scaler" has been developed for J-PARC accelerators. It is a customized module of Yokogawa-type PLC. A prototype module was used to measure accelerator signals, and showed expected functionalities.

Software to detect three cases of miss-triggers is discussed. EPICS databases to access module hardware were developed. Routines to detect miss-trigger events are in progress.

We appreciate the RF team of J-PARC MR, especially Yasuyuki Sugiyama and Keigo Hara, for their helps during test measurements using RF signals. We also thank all accelerator members and companies for their efforts to maintain the control system.

REFERENCES

[1] J-PARC website: http://j-parc.jp/index-e.html

[2] S. Nagamiya, "Introduction to J-PARC", Prog. Theor. Exp. Phys. (2012)02B001.

[3] K. Hasegawa *et al.*, "Performance and Status of the J-PARC Accelerators", in *Proc. IPAC2018*, Vancouver, Canada, May 2018, pp. 1038-1040.

[4] EPICS website: https://epics.anl.gov/index.php

[5] N. Kamikubota *et al.*, "J-PARC Control toward Future Reliable Operation", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, pp. 378-381.

[6] N. Kamikubota *et al.*, "Operation Status of J-PARC Timing System and Future Plan", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, pp. 988-991.

[7] K.C. Sato *et al.*, "Development of Triggered Scaler to Detect Messing Trigger", in *Proc. of 14th Annual Meeting of Particle Accelerator Society of Japan (PASJ'17)*, Sapporo, Japan, Aug. 2017, pp.627-630.

[8] Hitz website: http://www.hitachizosen.co.jp

[9] N. Kamikubota *et al.*, "Operation Experience and Migration of I/O Controllers for J-PARC Main Ring", in *Proc. PCaPAC'16*, Campinas, Brazil, Oct. 2016, pp.101-104.

[10] Y. Tajima *et al.*, "Development of EPICS-based Software of Triggered Scaler", in *Proc. of 15th Annual Meeting of Particle Accelerator Society of Japan (PASJ'18)*, Nagaoka, Japan, Aug. 2018, paper THP094, in press.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP22

GUARANTEEING THE MEASUREMENT ACCURACY IN Em#

X. Serra-Gallifa†, J. Avila-Abellan, M. Broseta, G. Cuni, O. Matilla
ALBA Synchrotron, Cerdanyola del Vallès, Spain

Abstract

ALBA, in collaboration with MAXIV, has developed a four-channel electrometer of 18bit deep with 8 ranges from 1mA to 100pA. The objective of accuracy in the measurements made clear from the beginning the need to compensate the components tolerances and its dependence with temperature. This paper describes the tests performed to characterize the acquisition chain, the automatic calibration developed and the hardware and software implemented to achieve the accuracy target. This implementation has been eased due to the high flexibility given by ALIN and Harmony [1] architectures used in the Em#.

EM# RETROSPECTIVE

Low current measurements are widely extended in Synchrotron light sources, like ALBA. The major need of electrometers in ALBA is for Front-Ends and Beamlines diagnostics. This need was identified during the construction of ALBA and the first version of the electrometer was designed [2]. This version was successfully installed in all ALBA beamlines and Front-Ends, but some beamlines went beyond using the electrometer as a detector. Besides the good precision achieved in the current measurements, this first version had some limitations that affected its functionality: synchronization performance, low ADC resolution, lack of calibration or temperature compensation, are some examples. In 2013, the need of a redesign a new electrometer electrometer was clear. The new instrument had to solve also a problem of obsolescence of the microprocessor board. The same year started the project of the second version of ALBA electrometer, Em# [3]. The first steps were defining an architecture which would give as much flexibility as possible and to avoid future obsolescence basing the design on standards and avoiding brand dependences. On that way, it was decided to use SPEC board from OHWR [4-5] and designed by CERN. In 2015, MaxIV Light source signed a collaboration agreement to join in Em# project. Finally, in 2016, 12 units in ALBA and 50 units in MaxIV were assembled.

ACCURACY OPTIMIZATION

One of the main targets of Em# was the improvement of its accuracy. The tests of first version of ALBA electrometer showed that the achieved precision exceeded expectations. The main reason of that precision was the Current amplifier (CA) low noise, equivalent or better than many state-of-art current amplifiers (Fig. 1). For that reason, the analogue part of electrometer was kept without including major changes. Some minor changes were optimizing the heat generation, enabling the offset correction foreseen in the first design with a DAC and the addition of temperature sensor accessible remotely.

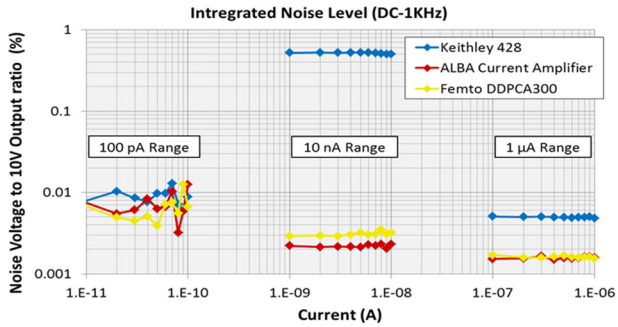


Figure 1: ALBA CA noise comparison with commercial electrometers.

One characterization showed that the noise measured in most of the ranges was so low that the ADC resolution was the limiting factor of the acquisitions. That lead to the decision of using a new the ADC with increased resolution of 18bits.

Another improvement implemented in the new Em# was the complete isolation of measurement ground respect the rest of grounds. This action minimizes the appearance of multiple ground loops which typically increase the noise of the measurements. The enhancement was not done in traditional way; it was decided to isolate the grounds after the ADC, isolating digital signals. A secondary effect of this architectural change was that it allowed measuring currents in environments were grounds needs to be biased. As this change had implications in the cost of the equipment and its auxiliary setup two versions of Em# were developed: one to work safely with HV bias (up to 1kV) and other to avoid the ground loops

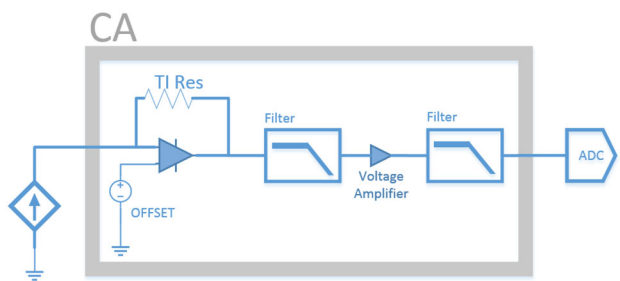


Figure 2: Schema of the CA between the current source and the ADC. The CA consists in a transimpedance amplifier with a gain, fixed by TI resistor and offset adjustment, a voltage amplifier and two low-pass filters.

COMPONENTS TOLERANCES

The first ALBA electrometer minimized the tolerances and temperature dependences in the design using matched pair resistors. However, as requirements in accuracy in-

† xserra@cells.es

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP22

creased the need of implementing a software based calibration inside the instrument appeared as unavoidable.

Before starting the implementation of a software measurement correction the critical components were analysed for a better understanding of the CA dependencies. The most critical components were located close to the first amplification stage: the electrical discharge protections, the transimpedance amplifier, the gain resistors and the offset adjustment circuit. Figure 2 can help to locate the components.

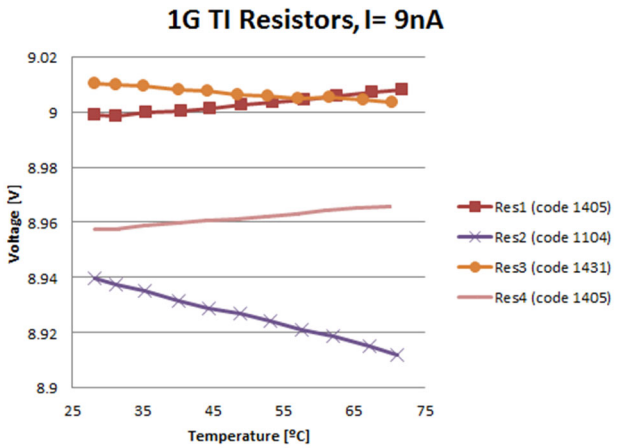


Figure 3: Study of temperature dependency of one of multiple resistors models analysed.

We did an isolated characterization of the first stage gain resistors versus temperature (Fig. 3). We surprisingly discovered that different units of the same resistor model could show negative or positive temperature coefficients even one could see a very similar tendency if the units came from the same production lot.

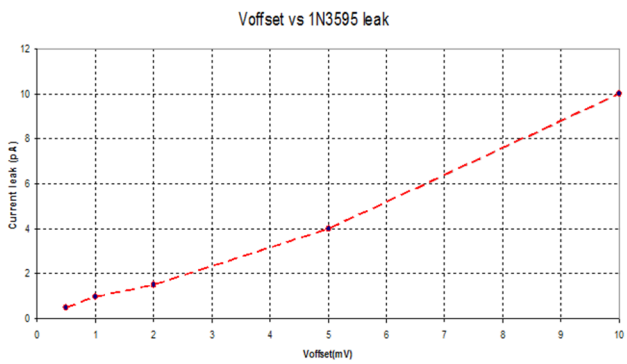


Figure 4: Plot of leakage current on 1N3595 diode versus offset voltage.

The second study was focused in the included discharge protection at each current input. The circuit is based on two small signal diodes 1N3595. These diodes generate a leakage current which depends exponentially on the offset voltage and temperature (see Fig. 4). Both dependences are difficult to separate, because the offset circuit generation also depends on temperature. That is why we did not try to disseminate its contributions since they were quite tangled.

CORRECTION ALGORITHM

The characterization of the current in open loop versus temperature showed that in lower ranges the exponential contribution of the protection diodes dominates whilst in higher ranges the linear contribution of the transimpedance resistors are higher. This can be seen in the following figures (Fig. 5 and 6).

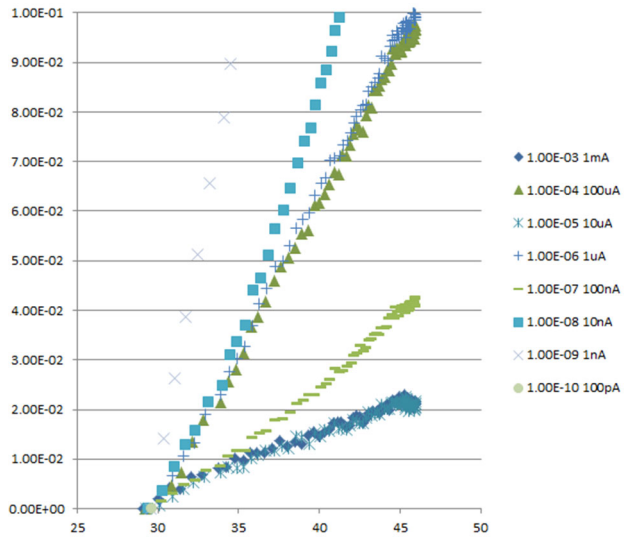


Figure 5: Offset current versus temperature, scaled to full scale.

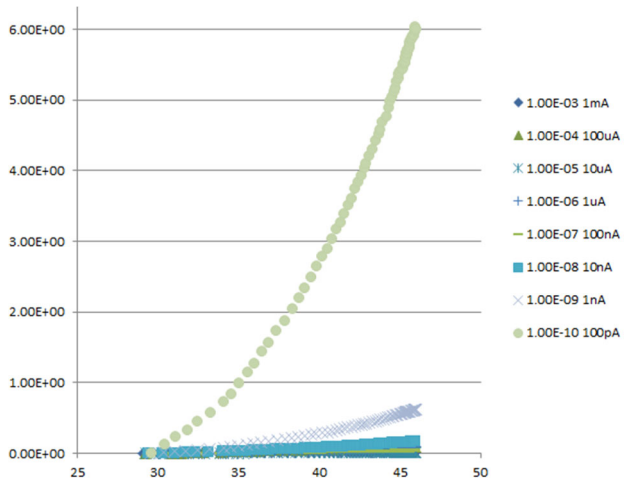


Figure 6: Offset current versus temperature, scaled to full scale. Lower ranges leakage current on diodes dominates.

The decision taken was to implement a correction algorithm of the offset and the gain modelling a linear behaviour with temperature. In case of a high accuracy needed in the lower ranges the protection diodes should be removed.

To implement such correction each CA include a temperature sensor close to this critical elements which reading is used to adjust the offset voltage to be applied. And in parallel a second correction is applied to compensate the gain dependency from temperature.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP22

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

CALIBRATOR

To guarantee operability and accuracy a design of a calibrator started. The calibrator generates a reference voltage using the voltage reference LTZ1000(see Fig. 7). This voltage reference with different resistors allows the generation of 8 different currents from 700µA to 70pA that allows calibrating all Em# ranges. A dependency with temperature bellow 0.1ppm in the range from 20ºC to 40ºC was achieved once the known gain resistor dependence was corrected.

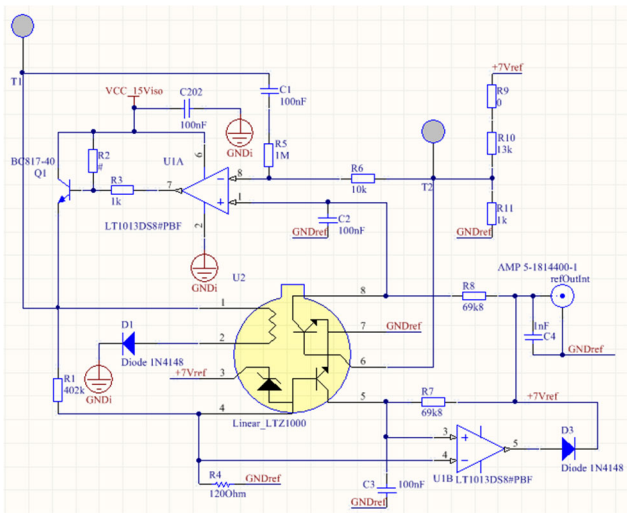


Figure 7: Schematic of voltage reference using the LTZ1000.

Therefore, to guarantee the calibrator accuracy lower than 1ppm requires calibration and correction of the reference currents of calibrator with the temperature. The work done in the CA showed that the resistors value depends linearly with temperature, so a linear correction is calibrated and stored in an internal EEPROM. The calibration is done with a Keysight 34470A multimeter and a Keithley 6517B electrometer. Figure 8 shows how calibrator looks like.

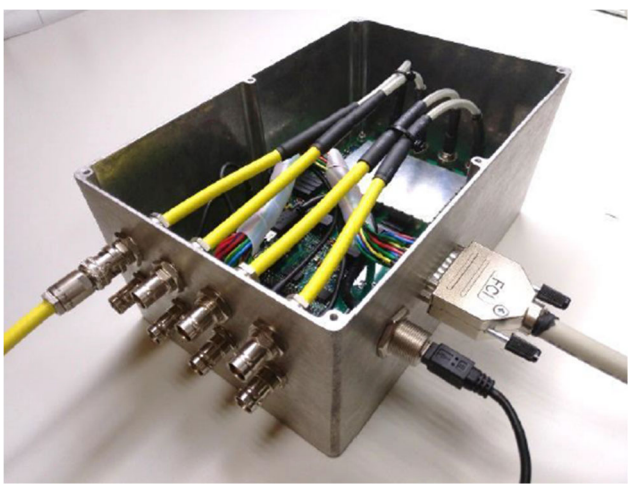


Figure 8: Calibrator.

CALIBRATION METHOD

The correction algorithm is based on a calibration values stored in an integrated EEPROM in each CA. The internal EEPROM allows 2 different calibrations storage: one reserved for a factory calibration and a second to let the users execute further calibration according with their calibration plan.

The implementation first measures temperature and the current in open circuit and adjust the offset voltage until 0 current is achieved. Then it injects a known current in the CA to calibrate its gain. This is done for each gain of transimpedance amplifier with the second stage voltage amplifier fixed to 1. The next step is to calibrate the voltage amplifier gains with the minimum gain on the transimpedance amplifier. This procedure is repeated at different temperature to characterize the temperature dependency.

The Em# includes script for an in-situ calibration. The script do the first measurement at power up and the second measurement when Em# temperature gets stable.

The values of temperature (T) and offset voltage (V_{OF}) in open circuit are used to calculate V_0 (offset voltage at 20ºC) and m (temperature coefficient to compensate offset voltage) in Eq. (1).

$$V_{OF} = V_0 \cdot \left(1 + m \cdot (T - 20)\right) \quad (1)$$

Then the reference current and the voltage read in the ADC are used to calculate the parameters β (tolerance of resistor nominal value R_N) and α (temperature coefficient) of transimpedance gain equation (Eq. (2)).

$$G_{TI} = R_N \cdot (1 + \beta) \cdot \left(1 + \alpha \cdot (T - 20)\right) \quad (2)$$

The temperature dependency of voltage gain is calculated following the same procedure.

CONCLUSION

Designing high accuracy instrumentation is a complex task. This paper summarizes the work needed to achieve that the improved accuracy target of the Em# electrometer could be assured in all cases.

Even concrete actions will necessarilydepends on the concrete instrument to be designed, there are different generic lessons that can be learned: the first one is that the investment of time in the deep characterization of an instrument of high accuracy is comparable with some design phases. The second one is that it is highly recommendable to design an internal architecture of the equipment that allows a flexible implementation of the possible dependencies that are found in the future. This has been the case of the Em# and thanks to its architecture flexibility now Em# is able to measure with requested accuracy comparable to the state-of-the-art electrometers.

REFERENCES

[1] M. Broseta *et al.*, "Present and Future of Harmony Bus, A Real-time High Speed Bus for Data Transfer between FPGA Cores", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017. doi:10.18429/JACoW-ICALEPCS2017-WEAPL01

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP22

[2] J. Lidón-Simon *et al*., "Low Current Measurements at ALBA", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, paper WEPMS025.

[3] J. Avila-Abellan *et al.,* "Em# Electrometer Comes to Light", ALBA-CELLS Synchrotron, in *Proc. ICALEPCS'17,* Barcelona, Spain, Oct. 2017. doi: 10.18429/JACoW-ICALEPCS2017-TUAPL04

[4] OHWR SPEC http://www.ohwr.org/projects/spec

[5] X. Serra, *et al.*, "A Generic Fpga Based Solution for Flexible Feedback Systems", *in Proc. PCaPAC'16*, Campinas, Brazil, Oct. 2016, paper FRFMPLCO06.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP23

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

FURTHER IMPROVEMENTS IN POWER SUPPLY CONTROLLER TRANSIENT RECORDERS FOR POST-MORTEM ANALYSIS OF BPM ORBIT DUMPS AT PETRA-III

P. Bartkiewicz#, C. Gindler, G. K. Sahoo, DESY, Hamburg, Germany

Abstract

PETRA-III is a 3rd generation synchrotron light source dedicated to users with 14 beamlines beginning operations in 2010. The storage ring was modified in 2014 for an additional 12 beamlines in two extensions. It is operated with several filling modes with a total current of 100 mA at electron beam energy of 6 GeV. The horizontal beam emittance is 1.30 nmrad with 1% coupling. During a user run the Machine Protection System (MPS) may trigger an unscheduled beam dump due to high deviations in orbits if transients in the magnet power supply (PS) currents are detected which are above permissible limits. PS controllers provide transient recorder data, showing differences between current set-point and readout values in a time span of several seconds around the moment of a beam loss. We describe automatic management system handling a large number of PSs, performing automatic transient recorder data readout, storing and is available for offline analysis. We discuss hardware implementation of transient recorders and its configuration software, a Java GUI application used to investigate the transient behavior of different PSs, which might have been responsible for emittance growth, orbit fluctuations, or the beam dumps seen in a post-mortem analysis.

INTRODUCTION

PETRA-III [1] is a 3rd generation synchrotron light source commissioned with an electron beam energy of 6 GeV and 100 mA stored current at betatron tune values of 37.12 and 30.28. The horizontal beam emittance is 1.30 nmrad while a coupling of 1% amounts to a vertical emittance of 13 pmrad. The machine was commissioned for experiments at 14 beam lines with 30 end-stations in 2010. The storage ring was further modified at extensions in East and North to incorporate 12 new beam lines including a super luminescence beam line from dipole radiation in 2014. PETRA operates with several filling modes, such as 40, 60, 80, 240, 480 or 960 bunches with a beam current of 100 mA. During the normal user operation, there are unscheduled beam dumps triggered by the Machine Protection System (MPS) [2]. These triggered dumps may occur before or sometimes after the loss of beam. The reasons for beam loss due to the MPS are of course understood. But the loss of beam prior to the beam dump by the MPS or a sudden drop in beam current, are both unexpected events. In these cases the cause remains unidentified or in some cases undetected. However, although the beam is lost, it leaves its signature in its post-mortem data. These post-mortem data are huge and contain a lot of information which can be extracted and analysed in a special Java Web Application Most Effective Orbit Correction (MEOC) [3, 4]. Here we discuss how the Power Supply Controller (PSC) Transient Recorders are used in the post-mortem analysis to pin point the source of disturbance in magnet power supplies, which will help us to avoid or rectify the source of orbit perturbation in machine operation in the future.

MAGNET CONTROL SYSTEM STRUCTURE OVERVIEW

Petra III ring contains 1158 magnets, supplied by 669 power supplies. Each power supply (PS) is digitally controlled by a corresponding intelligent power supply controller (PSC), responsible for switching the PS on/off, setting the output current value in various ramping modes, monitoring the output current values and performing other PS-specific control as well as PS diagnostics functionality as shown in Fig. 1.

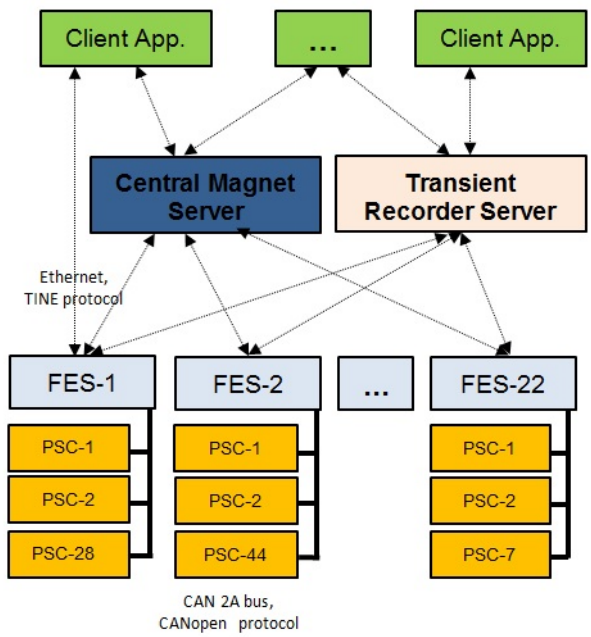


Figure 1: Magnet Control System structure overview.

All PSCs are managed by 22 front-end servers (FES) over 40 CAN buses [5] using CANopen [6-8] protocol. The Central Magnet Server communicates with front-end servers over Ethernet, using TINE [9] protocol and plays a role of the system integration unit and managing PSC group operations. It offers client applications a hardware and bus topology independent view, by hiding device hardware, addressing and fieldbus details.

piotr.bartkiewicz@desy.de

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP23

PSC TRANSIENT RECORDER IMPLEMENTATION

Firmware of the PSCs, besides other diagnostic functionality, implements also a transient recorder (TR) unit consisting of:

- Circular buffer for storing samples of a difference of output current set value (Sv) and a read back value (Rv).
- Configuration parameter set: maximum allowed output current deviation, trigger position, number of samples used for averaging, delay after ramping finish.
- Triggering system which analyses the absolute value of Sv-Rv and compares it to user-defined maximum deviation parameter.

Every 500 microseconds a sample, being a difference of set and read back value is calculated and stored for averaging. Depending on the user-defined settings the averaging can use from 2 to 255 samples. The averaged value is stored in a circular buffer of capacity of 20160 16-bit values. After receiving a start command, if the TR detects an event (that absolute value of the Sv-Rv difference is greater than programmed maximum current deviation), a trigger flag is raised in the PSC status word, the position in the circular buffer of the triggering sample is marked, and the data collection runs until the trigger position matches the requested position in the buffer. The trigger position, similarly to a trigger position in digital oscilloscopes, can be set from 0% (buffer contains only samples after the triggering event) up to 100% (the buffer contains all samples before the triggering event). Due to the impedance of the magnet circuit, the absolute value of Sv-Rv difference can temporally grow both during and just after the ramping process. Therefore the triggering mechanism is disabled during the ramp. A delay-after-ramp-finished parameter let to disable the triggering also for a specified period after the ramp is done. The details are shown in Fig. 2.

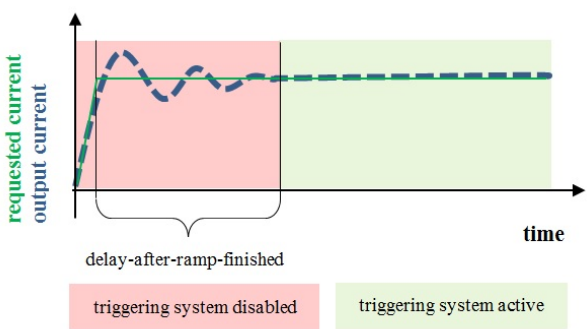


Figure 2: Triggering system disabling conditions.

In addition each sample stored in the circular buffer has a single-bit label showing, whether the sample was collected during the ramp, which helps to interpret the sample value during further analysis.

Completion of the data collection is indicated in the PSC status word and the circular buffer becomes ready for next trigger event readout.

FRONT-END SERVER MANAGEMENT OF TRANSIENT RECORDERS

The FES continuously monitors status of each PSC. For all PSCs whose transient recorders have received the trigger and which have finished collecting samples, a readout procedure is initiated. The entire circular buffer content (20160 samples, 40320 bytes) along with a header containing a PSC status 'snapshot' at that moment is transferred, when the trigger is detected. The snapshot consists of the requested value of the output current, the read back value of the output current, the PSC status word, the number of samples used for averaging and a triggering sample position within the data buffer.

The readout of transient recorder samples buffer is performed as a CANopen segmented SDO [8] transfer. In order to minimize impact of elevated CAN bus load during the transfer on a normal control operation, an additional, dedicated, low priority SDO channel is used. After the transfer of transient recorder data is completed the triggering system becomes reactivated again. The transferred buffer and the header form a record of the transition event, which can be accessed by client applications as well as by the Transient Recorder Server over the TINE interface.

The front-end servers are also responsible also for keeping the PSC transient recorders configuration up-to-date. If the PSC status shows the PSC has been power-cycled or reset, then the FES sends over the CAN bus the recently used TR parameter set.

TRANSIENT RECORDER SERVER

The Transient Recorder Server integrates a huge number of transient recorders and front-end servers into one system, by providing users with methods to access each PSC and thus its transient recorder by its name and hiding all Front-End Servers and CAN buses addressing details.

The most effective TR configuration strongly depends on the corresponding power supply output current range and on electrical properties of the related magnet. The Transient Recorder Server maintains a PSC configuration data base, where configuration of each TR is stored. It also provides users with a possibility of creating PSC configuration groups (e.g. "quadrupoles, 400A") and assigning the same configuration to all group members. One can also issue commands such as "start", "stop", or "test trigger" to all transient recorders belonging to a selected group. It greatly simplifies testing and maintenance of such a complex system.

The Transient Recorder Server collects transient recorder event records from the front-end servers and stores them on the disk. The stored records can be obtained by client applications over the TINE interface for further analysis. There is also an automatic mechanism to delete older records, in order to keep the archive size reasonably small.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP23

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

TRANSIENT RECORDER SYSTEM MAINTENANCE CLIENT APPLICATION

Users can manage the transient recorder system by using a dedicated Magnet PSC Transient Recorder Management Client application (as shown in Fig. 3). The application communicates with the Transient Recorder Server over Ethernet using TINE protocol and reflects the entire functionality of the server. It offers users a comfortable interface to configure and examine a single transient recorder. It also makes it possible to define configuration groups, assign power supply controllers to them, and next to propagate a configuration of a single, selected Transient Recorder to all group members.

There is a feature also to display of status information (such as triggering system active/not active, TR triggered, buffer data transmission in progress) of each of all the transient recorders. The displayer can the present status of transient recorders organized in configuration groups, or also in other, user-definable 'operational' groups, which might serve better, more machine operation specific, views.

The client application obtains a list of the most recently triggered recorders, and the user can examine a transient recorder event for each of them. The time based plot of the transient recorder event, as well as FFT plots are helpful for a rapid data analysis or the case documenting of the specific event.

The possibility of exporting the event in the form of an ASCII file may be used for some further studies.

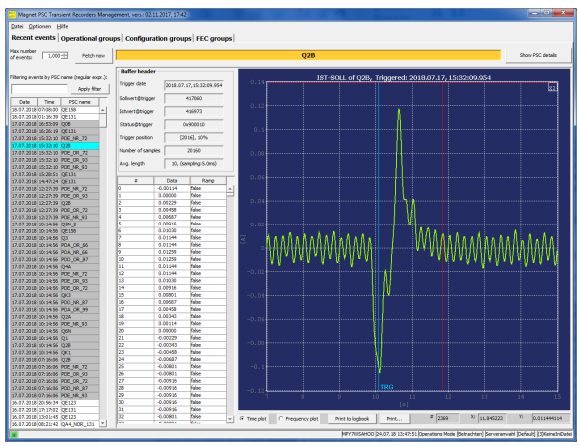


Figure 3: The graphics user interface of Magnet PSC Transient Recorder Management Client application.

RESULTS AND DISCUSSIONS

Using post-mortem beam transverse orbit dump data a proper scrutinization of turn by turn orbits and the frequency spectrum measured at a BPM may improve the understanding of a beam loss and may help to increase the availability of machine operation by eliminating the sources of disturbances. For these purposes, there are 246 by Libera [10] Beam Position Monitors (BPM) distributed in 2303.952m ring to monitor the transverse orbits.

These BPMs are connected with a Ring Buffer where the 16384 latest turns of data for each BPM are continuously stored in each Libera. When the Libera server receives a beam dump signal from the MPS, it dumps its 16384 turns of orbit data for each BPM to an Archive Server with an event time stamp for post-mortem analysis. The MEOC method is applied to identify correctors that might have perturbed the golden orbit leading to violations of the interlock limits at an active BPM. Due to a transient malfunction of a magnet, the orbit will grow and surpass the interlock limits at some special BPMs and the beam will be dumped by the MPS. In a post-mortem analysis this change in orbit can be corrected by a few correctors using MICADO [11] algorithm to investigate the cause of beam loss in transverse plane.

The MEOC is utilized to investigate the suitable corrector that might have perturbed the orbit beyond the interlock limits. For example, the event (Mon Apr 16 15:14:41 CEST 2018) was due to the transients of the vertical corrector magnet PKVSX_WR_140 which was receiving wrong set values due to spikes leading finally to a beam dump via orbit interlocks. You can see from Fig. 4 that the difference orbit was well corrected to zero using the same corrector.

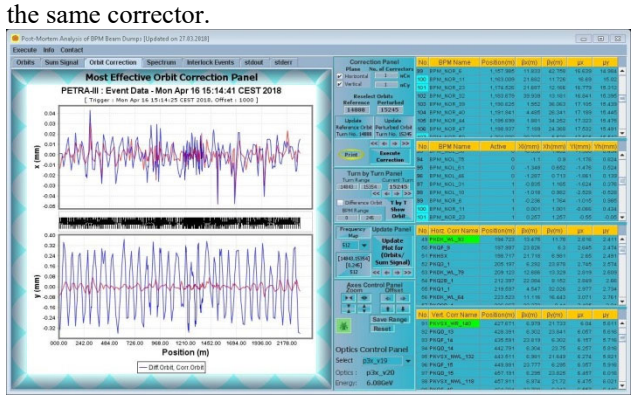


Figure 4: Vertical orbit correction for the Interlock Event on Monday 16 April 2018 at 15:14:41 which indicates that PKVSX_WR_140 vertical corrector as the source of orbit perturbation.

CONCLUSION

The present PSC Transient Recorder is utilized to monitor the transients in 669 PSs of the PETRA III electron storage ring. All the PSs are manually put in active mode which are triggered when the difference in read and set values are larger than the thresholds of respective PSCs. This is used with a post-mortem analysis of BPM beam dumps to find the responsible PS that disturbs stability or causes a loss of beam. We are currently developing better, more user friendly utilities of PSC transient analysis in order to improve stability and reliability of PETRA III operation.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge and would like to thank for the initial contributions by Bernd Pawlowski; and Rainer Wanzenberg & Reinhard Bacher for their constant encouragements and guidance.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-THP23

REFERENCES

[1] K. Balewski, W. Brefeld *et al.*, "PETRA III: A New High Brilliance Synchrotron Radiation Source at DESY", in *Proc. EPAC'04*, Lucerne, Switzerland, Jul. 2004, paper THPKF019, pp. 2302-2304.

[2] T. Lensch, M. Werner, "Machine Protection System for PETRA III", in *Proc. DIPAC'09*, Basel, Switzerland, May 2009, paper TUPD25, pp. 351-353, 2009.

[3] G. K. Sahoo, K. Balewski, A. Kling, "Post-Mortem Analysis of BPM-Interlock Triggered Beam Dumps at PETRA-III", in *Proc. 9th International Workshop on Personal Computers and Particle Accelerator Controls (PCaPAC'12)*, Kolkata, India, Dec. 2012, paper WEPD22, pp. 43-45.

[4] G. K. Sahoo, P. Bartkiewicz, A. Kling, B. Pawlowski, "Power Supplies Transient Recorders for Post-Mortem Analysis of BPM orbit Dumps at PETRA III", in *Proc. 10th International Workshop on Personal Computers and Particle Accelerator Controls (PCaPAC'14)*, Karlsruhe, Germany, October 2014, paper FPO031, pp. 222-224.

[5] Wolfhard Lawrenz, "*CAN System Engineering: From Theory to Practical Applications*", Springer-Verlag, New York Inc., 1997.

[6] Olaf Pfeiffer, Andrew Ayre, Christian Keydel, "*Embedded Networking with CAN and CANopen*", RTC Books, 2003.

[7] CiA CAN in Automation, http://www.can-cia.org/.

[8] CiA Service Data Object, https://www.cancia.org/can-knowledge/canopen/sdo-protocol/.

[9] TINE, http://tine.desy.de/.

[10] Instrumentation Technologies, https://www.i-tech.si/accelerators-instrumentation/libera-brilliance-plus/benefits_1/.

[11] B. Autin and Y. Marti, "*Closed Orbit Correction of Alternating Gradient Machines using a Small Number of Magnets*", CERN/ISR-MA/73-17, CERN, 1973.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCA1

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

NEW COLLABORATIVE APPROACH TO SCIENTIFIC DATA MANAGEMENT WITH NOVA

W. Mexner*, E. Bründermann, M. Caselle, S. Funkner, A. Kopmann,
G. Niehues, N. Tan Jerome, M. Vogelgesang
Karlsruhe Institute of Technology, Karlsruhe, Germany

Abstract

Accelerator physics studies at the storage ring KARA at KIT produce terabytes of diagnostics data per day, which is recorded once and then reused on a long-term basis to answer different research questions at KIT. Finally, raw data and intermediate analysis results should be published along with scientific results. Thus storing from the very beginning the data of all analysis steps and its metadata in a central portal would be very beneficial. Similar requirements exist for synchrotron X-ray micro tomography at the KIT imaging cluster and there is an interest to share the large data analysis effort. By using a new collaborative approach, the NOVA project aims to create tools, to enable an efficient use of valuable beam time. For micro tomography beamlines the project will build up a comprehensive database of various demonstrator organisms for the morphological analysis of animals. The NOVA portal is integrated in the local data handling procedures and the datasets automatically appear in the NOVA portal as they are recorded. For both applications, accelerator diagnostics and X-ray tomography, the NOVA portal will offer new collaborative tools to enable synergetic data analysis.

INTRODUCTION

Scientific data management is becoming increasingly important for large scale physics facilities. The European Commission points out already in 2016 that e-science is essential to meet the challenges of the 21st century in scientific discovery and learning [1]. The emerging question is: Is your data useful for somebody in 10 years? Experiments nowadays produce data at extremely high rates, which are put high demands to the whole data acquisition chain with respect to data analysis, curation, storage and usages. While the first two steps are naturally in the focus of the scientists, the later steps are often not handled with the same effort. However, the basics of the data lifecycle must be considered as early as possible so that curated data sets with high-quality content are stored. The experimental boundary conditions must be described in form of metadata as completely as possible in order to be able to analyze data later and reuse it interdisciplinary. For providing open access to these data, the FAIR data principles, as introduced by Wilkinson et al. [2], are important. Data have to be **F**indable, **A**ccessible, **I**nteroperable and **R**eusable. To be findable, data should have a globally unique and eternally persistent identifier. To be accessible, also (meta)data have to be online. For being accessible, (meta)data have to be retrievable by their identifier using a standardized open, free and platform independent protocol, e.g. a public repository. To be interoperable, (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation. To be reusable, metadata have a plurality of accurate and relevant attributes and should be published together with a clear and accessible data usage license.

Detectors with high temporal and spatial resolution are nowadays available and used in an increasing number of experiments. The resulting data rates are faster growing than data handling technologies. Thus, we move to an era where data becomes to large to copy and more and more scientists in smaller organizations are excluded from latest technologies. During the last ten years several large effort has been taken to find solutions for improved analysis, management and access to large data sets. Synchrotron X-ray tomography has served as an example for imaging applications in general. The integration of online data processing in DAQ systems is essential for high data rate applications. It improves the quality of recorded data and enables advanced experimental control. For the demonstrator application X-ray tomography, a suite of modular software components, called the UFO GPU computing platform, has been designed and implemented. It is intended to execute complex beamline protocols, including real-time data investigation and on-site data processing [3–6]. New methods are added to enhance reconstruction quality and to compute acceptable reconstructions from few data [7, 8]. To manage scientific data at all processing levels, we started the NOVA project. In NOVA, the Network for Online Visualization and Synergistic Analysis, a group of X-ray experts, engineers, computer scientists, mathematicians and biologists teams up to advance analysis tools for tomographic data. The project aims for synergistic data analysis and is building up a comprehensive data portal for morphologic images of small insects [9, 10].

The instrumentation in accelerator physics is currently undergoing a dramatic change. With the new instruments KAPTURE and KALYPSO developed at KIT continuous monitoring of electron beams is possible [11]. These tools uncover phenomena in elctron bunch dynamics and enable advanced beam control [12]. Similar to the imaging applications mentioned before multidimensional datasets with sizes up to the Terabyte level are recorded. Common to all this high rate and high resolution measurements is the need to describe datasets carefully and to provide hierarchical exploration of these datasets. Due to the size of the datasets, storage is costly and access is complex and time consuming.

* wolfgang.mexner@kit.edu

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCA1

Repositories like the European projects EUDAT or ZENODO allow to store datasets with searchable metadata, but currently there are no suitable tools available to browse this kind of datasets and to inspect multi-scale phenomena. In order to share analysis effort in international communities and to provide access to raw data with scientific publications we will adapt the technologies developed with the NOVA web portal and investigate methods to browse in multidimensional datasets. We are convinced, complex datasets need precise textual metadata and visual exploration. Improved visualization is the key to the development of better metadata for multidimensional datasets.

As a first prototype of the NOVA portal for the tomographic datasets has been already developed. In the next step we will adapt the NOVA environment for electron beam diagnostics at KIT. It is intended to explore with a second community the benefits of collaborative data analysis. The system is constantly evolving and integrated with both user-visible front-end components to enhance the view on the data as well as back-end components that connect the system with the actual experiment setup. In the future, the system will enable scientists to structure their data automatically in a hierarchical manner, which is a prerequisite for efficient storage of large data volumes.

NOVA DATA PORTAL

Synchrotron X-ray microtomography offers unique opportunities for the morphological analysis of small animals. Internal structures become observable even in opaque organisms in a non-invasive, three-dimensional way at sub-micron resolution. By using a new collaborative approach, NOVA aims to create new possibilities allowing for a more efficient use of valuable beam time at tomographic synchrotron beamlines. At the same time the project establishes a comprehensive database of various demonstrator organisms and develops the NOVA data portal to archive, analyze, and share these datasets.

The NOVA data portal is split into a core system and third-party services. The core system provides login, authentication, and management facilities as well as a REST API to interact with those facilities remotely. Third-party services can register with the core system to provide additional functionality and features that should or could not run on the same machine as the portal backend itself. Two main services that are currently implemented are the thumbnail service and a 3D visualization service. The thumbnail service is used to generate iconic images to get a quick idea what kind of data is contained in a data set. The service reads the middle slice of a dataset, shrinks it to a user-defined size and maps the grey values to an RGB triple and saves the resulting JPEG in an on-disk cache from which a browser request is served. Instead of using a generic icon, part of the data with a distinguishing color helps to identify a previously seen data quicker (Fig. 1, left). The 3D visualization service generates on-demand slice maps from the raw image data for client-side rendering using a 3D WebGL browser library WAVe. The service receives a description of the desired dataset and its volume origin and region, then reads the required slices, rescales them and re-orders them in a 2D slice map. This process happens in asynchronous fashion, i.e. the client requests the generation and waits for the service to finish processing and get the final slice map URLs. The slice map's filename is determined by a hash computed from the size requirements for efficient on-disk storage and retrieval. On the front-end side the 3D visualization is started as soon as the slice maps for the entire volume is generated. The user can then translate, rotate, and scale the volume and define a bounding box to "zoom in" to load a smaller part of the volume with a higher resolution. This zoom request will then generate a new task for the slice map server with a new origin and a new region. The right-hand side of Fig. 1 shows the front-end after zooming in on a particular area of the volume.

The WAVe Library for 3D Web Visualization

The WAVe library is a web-based 3D visualization application that renders volumetric slice map data as 3D objects. It consists of multiple modules which render the data in different forms, i.e. using surface rendering or direct volume rendering. These modules are implemented as shaders that enable a high flexibility in fine-tuning the GPU usage such as varying the ray-casting steps or performing various interpolation schemes. The library is based on the Three.js framework which simplifies the 3D scene management. Figure 2 shows the architecture of the library. The API layer allows users to integrate the library into any user interface designs.

ADAPTING NOVA FOR ACCELERATOR RESEARCH

The presented data management workflow has been developed from the beginning in a modular and portable way. It is meant as a generic approach for applications with large multi-dimensional datasets. While being developed for organizing tomographic datasets of biological origin, it can easily being adopted to other fields. We have identified a second use-case for accelerator physics data recorded at the KARA light source, which was also used to acquire tomographic data in the ASTOR campaigns. The accelerator research data e.g. Terahertz research, consists of one Terabyte per machine physics day, with the capability of one Terabyte per hour, and has been acquired by different detectors like EO-Laser, Streak camera, HEB detector, KAPTURE and KALYPSO. The development of fast detection methods for comprehensive monitoring of electron bunches is a prerequisite to gain comprehensive control over the synchrontron emission in storage rings with their MHz repetition rate. For example the "KArlsruhe Linear arraY detector for MHz-rePetition rate SpectrOscopy" (KALYPSO) developed at KIT allows to detect longitudinal electron bunch profiles via single-shot,near-field electro-optical sampling at the Karlsruhe Research Accelerator (KARA) [12]. For one analysis,

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCA1

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

100.000 samples with 0,91 Mhz are taken with signal and another set of 100.000 samples as calibration measurements and ar stored structured as tables in the container format HDF5. The organization of the data takes place on file system level with a structuring by date / filling number / detector and measuring station. The metadata is recorded separately in a digital log book (elog) and the accelerator's operating data is automatically logged into a NoSQL Cassandra database.

The interesting use case for NOVA is, that these data has been recorded only once and can be reused on a long-term base for different accelerator physic research questions. Informations like the shift of the bunch profile center in relation to different machine parameters could be automatically calculated and displayed in the portal. Thus, offering all these data enriched with metadata in a central portal would be a big benefit for accelerator research. The core infrastructure is capable of hosting this data, the question that arose was how to ingest it into the system. For the already stored data, some metadata is partially digitally no available (e.g. paper logbooks) or differently structured depending on each experiment type. This means importing can only happen case-by-case. To alleviate this, we identified common metadata structures and began writing an import description format and an import tool that is capable of transforming input data to a common format usable by the NOVA portal. This tool processes an easy extensible job list in JSON format (Fig. 3), which collects all available metadata of the datasets of a certain experiment type and could be added by specific filters collecting data from additional sources like e-logbooks and the Cassandra database.

SUMMARY AND OUTLOOK

Existing repositories like EUDAT or Zenodo allow one to publish digital data with a digital object identifier and make it searchable. The NOVA portal is not only a repository, but allows to browse digital 3D datasets. We are planning to adapt the NOVA collaborative research approach to the

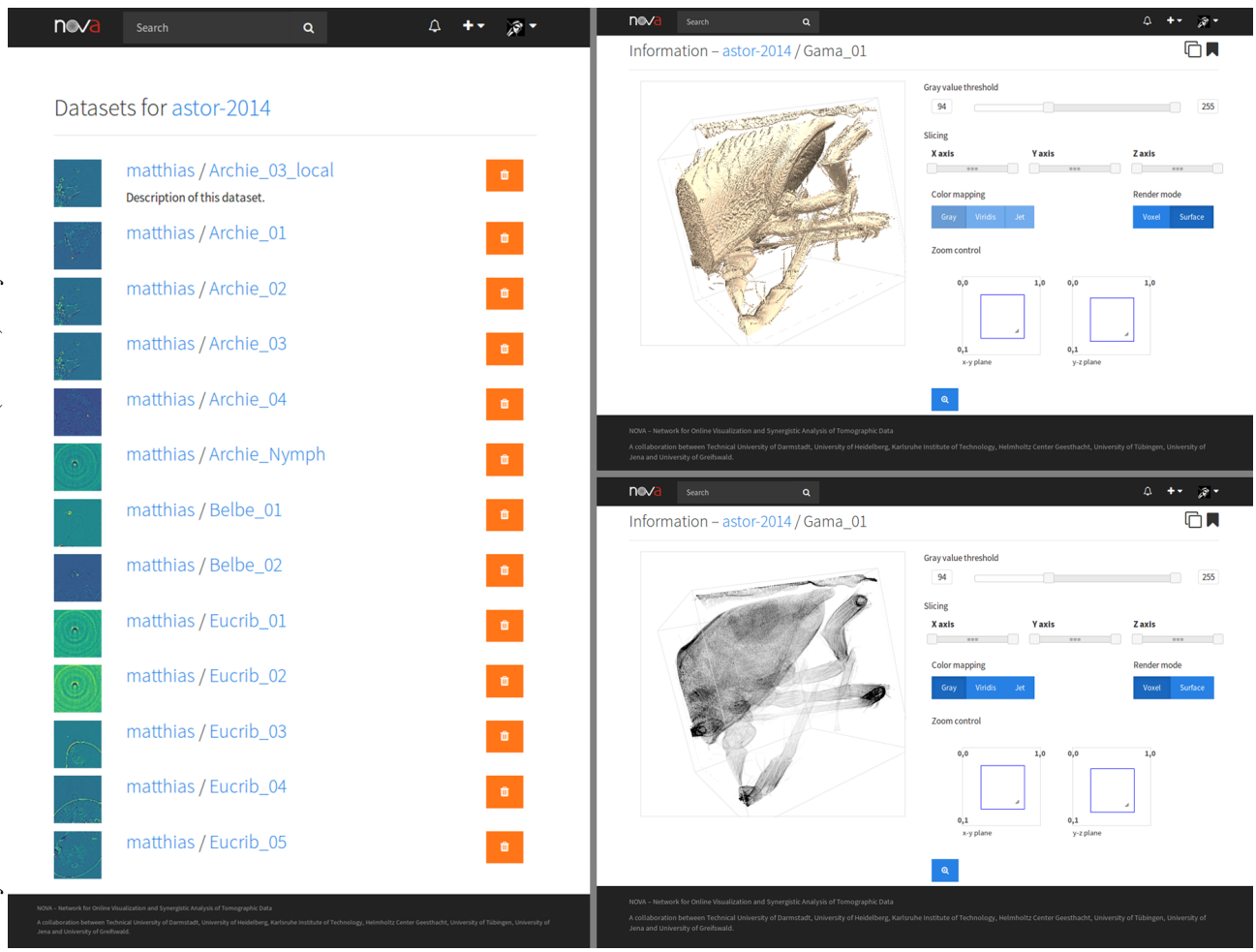


Figure 1: Overview of the NOVA web portal. Left. Thumbnails shown are generated by the thumbnail service allow for easier and quicker distinction between different datasets. Right. Interactive 3D visualization for a selected dataset as surface (upper image) and voxel rendering (lower image). The two smaller blue boxes denote the current zoom area in the x-y- and y-z plane, the blue bounding box in the 3D view shows the current slicing operation.Scheme of the FLUTE accelerator with all installed and planned components.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCA1

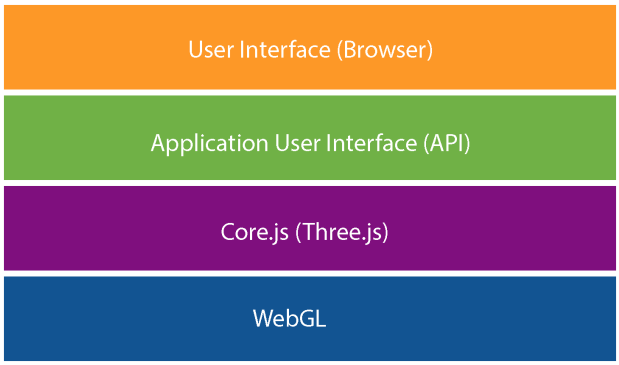


Figure 2: The image represents the high-level architecture of the WAVe library.

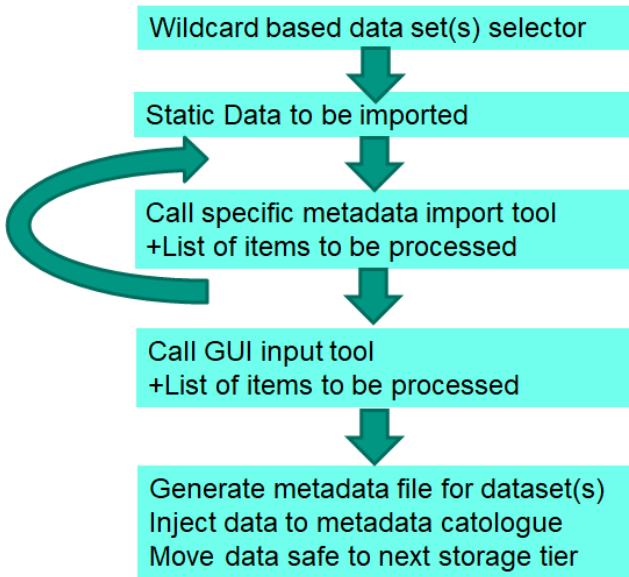


Figure 3: Concept of importing metadata to NOVA portal.

accelerator community for the new generation of detectors for accelerator beam diagnostics.

ACKNOWLEDGMENT

We like to thank the german Federal Ministry of Education and Research (BMBF) supporting the development of the NOVA portal with grant 05K16VBK.

REFERENCES

[1] https://www.naturvardsverket.se/upload/kalendarium/2016/open-data/jiri-pilar-eu-commission.pdf

[2] M. D. Wilkinson, M. Dumontier, *et al.*, "The FAIR Guiding Principles for scientific data management and stewardship", *Scientific Data*, vol. 3, pp. 160018, 2016. doi:10.1038/sdata.2016.18

[3] M. Vogelgesang, T. Farago, T. dos Santos Rolo, A. Kopmann, and T. Baumbach, "When hardware and software work in concert", in *Proc. IPACLEPCS'13*, San Francisco, CA, USA, Oct. 2013, paper TUPPC044, pp. 661–664.

[4] T. Dritschler, S. Chilingaryan, T. Farago, A. Kopmann, and M. Vogelgesang, "InfiniBand interconnects for high-throughput data acquisition in a TANGO environment", in *Proc. PCaPAC'14*, Karlsruhe, Germany, paper FPO001, pp. 161–163.

[5] M. Vogelgesang, T. Farago, T. F. Morgeneyer, L. Helfen, T. dos Santos Rolo, *et al.*, "Real-time image-content-based beamline control for smart 4D X-ray imaging", *Journal of Synchrotron Radiation*, vol. 23, no. 5, pp. 1254–1263, 2016. doi:10.1107/S1600577516010195

[6] M. Vogelgesang, L. Rota, L. E. A. Perez, M. Caselle, S. Chilingaryan, *et al.*, "High-throughput data acquisition and processing for real-time x-ray imaging", in *Proc. SPIE*, San Diego, CA, USA, 2016, vol. 9967. doi.org/10.1117/12.2237611

[7] A. Shkarin, E. Ametova, S. Chilingaryan, T. Dritschler, A. Kopmann, *et al.*, "An Open Source GPU Accelerated Framework for Flexible Algebraic Reconstruction at Synchrotron Light Sources", *Fundamenta Informaticae*, vol. 141, no. 2–3, pp. 259–274, 2015. doi:10.3233/FI-2015-1275

[8] R. Shkarin, E. Ametova, S. Chilingaryan, T. Dritschler, A. Kopmann, *et al.*, "GPU-optimized direct Fourier method for on-line tomography", *Fundamenta Informaticae*, vol. 141, no. 2–3, pp. 245–258, 2015. doi:10.3233/FI-2015-1274

[9] S. Schmelzle, M. Heethoff, V. Heuveline, P. Loesel, J. Becker, *et al.*, "The NOVA project: maximizing beam time efficiency through synergistic analyses of SRμ CT data", in *SPIE 2017*, San Diego, CA, USA, 2017.

[10] NOVA web page, http://ufo.kit.edu/dis/project/nova.php

[11] M. Caselle *et al.*, "A high-speed DAQ framework for future high-level trigger and event building clusters", *JINST*, vol. 12, pp. C03015, 2017. http://iopscience.iop.org/article/10.1088/1748-0221/12/03/C03015

[12] S. Funkner *et al.*, "High throughput data streaming of individual longitudinal electron bunch profiles in a storage ring with single-shot electro-optical sampling", arXiv.org, arXiv:1809.07530v1 [physics.acc-ph], 2018.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCA2

DEVELOPMENT OF A TASK-ORIENTED CHATBOT APPLICATION FOR MONITORING TAIWAN PHOTON SOURCE FRONT-END SYSTEM

Yu-Zheng Lin*, Jyun-Yan Chuang, I-Ching Sheng, Yu Tsun Cheng,
Chin-Chun Chang, Yi-Chen Yang, Hsin-Pai Hsueh, Chih-Hsien Huang
National Synchrotron Radiation Research Center, Hsinchu, Taiwan

Abstract

In this study, we propose a task-oriented chatbot application as an interactive user interface for monitoring Taiwan Photon Source front-end system. This application can get specific information faster and improve the efficiency of the maintenance engineer's definition of fault information when a fault occurs. The chatbot uses LINE's Message API to provide services, and obtains information and status of the front-end system over EPICS protocol, responding to users' needs, like a virtual assistant. At present, the application can obtain the front-end system information already in operation of Taiwan Photon Source, including x-ray beam position monitor, safety interlock system, front-end valve status, vacuum status, etc. However, in addition to the passive provision of information, this program has a fault warning function, and will actively transmit fault information to relevant personnel when a problem occurs in the safety interlock system.

INTRODUCTION

The chatbot application is a dialogue system. Unlike a GUI system that provides a lot of information, the dialogue system is search-based. When the user asks for information, the dialogue system returns the answer. In this study, we propose a chatbot application for monitoring Taiwan Photon Source(TPS) front-end system. This chatbot will make engineer and scientist get TPS front-end status easier and improve work efficiency.

The chatbot application is based on LINE message platform. LINE is a cross-platform communication software that can be easily used on mobile phones, tablets, and personal computers. LINE provides "LINE Message API" [1] with a complete development solution, allowing developers only need to develop once and users can use this application on different platforms. This app has two main functions, one is to reply to the user query and the other is the active fault alert. The reply function is when the server receives the user query, service will use the rule-based engine to filter out the keyword, then uses "Experimental Physics and Industrial Control System"(EPICS) [2] call the associated EPICS process variable to get status information and finally passes the results back to the user. This service also continuously monitors the status of the system. When a failure is detected, it will actively send messages to the responsible personnel of the system.

* lin.yz@nsrrc.org.tw

CHATBOT ARCHITECTURE

The chatbot architecture is shown in Figure 1. We applied to LINE for a channel to implement this chatbot application, this channel like a friend in a social network, users just add this channel as a friend and can use it. The chatbot application is all made by Python, the LINE message API is used to connect to the LINE channel and use the push and reply methods to make fault alerts and reply to the status results queried by the user.

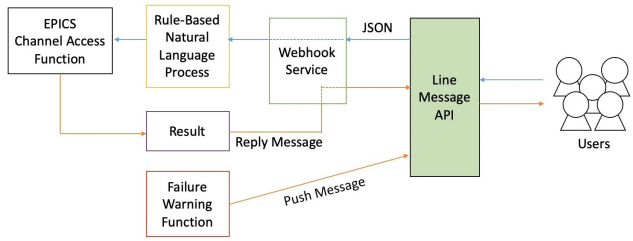


Figure 1: Chatbot architecture.

Reply Function

The Reply function has three parts: Webhook Service, Rule-Based Natural Language Process Engine and EPICS Channel Access Function.

Webhook Service is a HTTP server [3]. To implement this service, python and the flask web framework [4] is used. When a user sends a message to the chatbot, webhook service will receive a JSON file from LINE message API, this JSON file include message information, event source and time. The screenshot of the server operation is shown in the Figure 2.

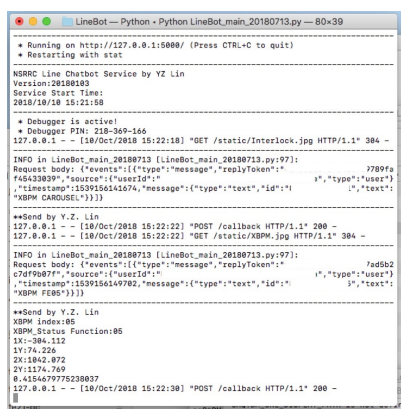


Figure 2: Screen shot of the webhook service operation.

After receiving the message, the Rule-Based Natural Language Process Engine will filter out the keywords in the

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCA2

message. Then use the EPICS Function callback related values according to different keyword rules. Finally, the value is filled in the formatted string of the result and use reply method from LINE message API send back to the user who sent the query.

Figure 3 is an example of how X-Ray beam position monitor information is returned. The rule-based engine will detect if "XBPM" and "FE" is in the text, and filter out the front-end numbers if the condition is true. After filtering out the TPS front-end numbers, this numbers will be filled in "XBPM_Status" subroutine to get X-Ray beam position value. Finally, fill the value into the formatted result string and use LINE message API push method returns the result to the user.

```
if "XBPM" in strupper and "FE" in strupper:
    str=event.message.text
    FE_number=re.sub("\D","", str)#take out FE number
    print('XBPM index:%s'%FE_number)
    if bool(FE_number)==True:
        try:
            #Use EPICS to get XBPM value
            XBPM_1X, XBPM_1Y, XBPM_2X, XBPM_2Y = XBPM_Status(FE_number)
            #Use LINE message API reply method return the result
            line_bot_api.reply_message(event.reply_token,TextSendMessage\
                                       (text="FE%s XBPM Position:\n"%FE_number+\
                                       "• 1X: %s um\n"%XBPM_1X+\
                                       "• 1Y: %s um\n"%XBPM_1Y+\
                                       "• 2X: %s um\n"%XBPM_2X+\
                                       "• 2Y: %s um"%XBPM_2Y))
        except:
            print("No Index")
            return 0
    else:print("No Index")
    end = time.time()
    elapsed=end-start
    print(elapsed)
    return 0
```

Figure 3: Example of how X-Ray beam position monitor information is returned.

Active Fault Alert Function

Active fault alert function is currently used on the TPS front-end safety interlock system. This function monitor ten TPS front-end safety interlock systems status with EPICS. When the safety interlock system fails, this function will send a warning message to the responsible person using LINE message API push method. The message will include possible causes of failure based on the logic of the safety interlock system.

CHATBOT APPLICATION

The user experience of using this chatbot application is like chatting with a friend, very simple and quite easy to understand. The LINE user just needs scans the QR code of the chatbot and adds the chatbot to the friend to start using the chatbot application.

The user only needs to enter the text to be queried, if this string of text matches the rule-based engine definition and the chatbot will return the result. Figure 4 is an example of the user querying X-Ray beam position monitor status and TPS front-end valve, the figure shows that the value of X-Ray beam position monitor and TPS front-end valve status is returned.

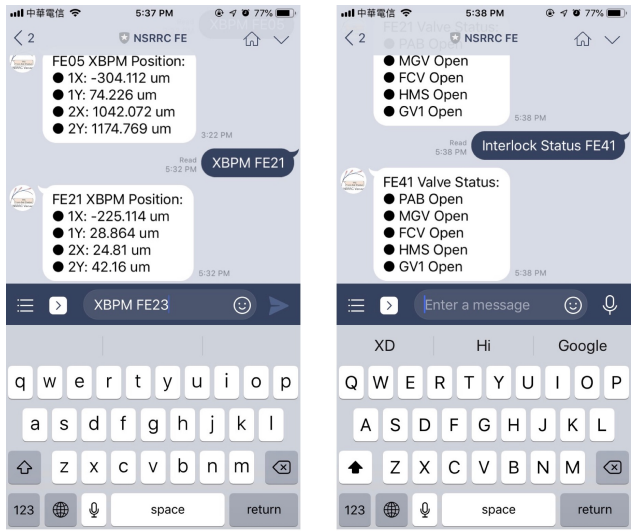


Figure 4: Screen shot of the Chatbot application:
(A) Query X-Ray beam position monitor information,
(B) Query TPS front-end valve status.

CONCLUSION

The chatbot application used to query and provide fault warning information for the front-end system of Taiwan Photon Source has completed the prototype. The application can obtain the front-end system information already in operation of Taiwan Photon Source, including x-ray beam position monitor, safety interlock system, front-end valve status, vacuum status, etc. The future will increase the amount of information that can be provided and enhance the user experience. We also consider providing services on different platforms, such as Facebook, WhatsApp, etc., so that international users of Taiwan Photon Source can use this application more conveniently.

REFERENCES

[1] Messaging API - LINE Developers, https://developers.line.me/en/docs/messaging-api/overview/

[2] EPICS - Experimental Physics and Industrial Control System, https://epics.anl.gov/

[3] Web hooks to revolutionize the web, http://progrium.com/blog/2007/05/03/web-hooks-to-revolutionize-the-web/

[4] Welcome to Flask — Flask 1.0.2 documentation – Pocoo http://flask.pocoo.org/docs/1.0/

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCB1

DATA ACQUISITION SYSTEM FOR ELI BEAMLINES

P. Bastl†, P. Pivonka, B. Ploetzeneder, O. Janda, J. Trdlicka, V. Gaman, J. Sys,
ELI Beamlines/Institute of Physics of the ASCR, Prague, Czech Republic

Abstract

The ELI Beamlines facility is a Petawatt laser facility in the final construction and commissioning phase in Czech Republic. In fully operation phase, four lasers will be used to control beamlines in six experimental halls.

In this paper we describe the ultrafast and distributed data acquisition system of the facility. First, we discuss the requirements and demands on the system and introduce the environment. Then we discuss the structural divide in top- and low-level system. The unique idea of this system is the extensive fibre-based low-latency communication infrastructure, which distributes large RAM buffers and allows to offload computational tasks with high throughput and real-time capability.

INTRODUCTION

ELI Beamlines [1] is an emerging high-energy, high-repetition rate laser facility located in Prague, Czech Republic. Four laser sources will supply six experimental halls, which provide various secondary sources to users. Facility commissioning, and installation work of lasers and experiments is progressing, and first enabling experiments are in progress (2018).

The general architecture of the control / DAQ system is introduced in [2]. This paper will specifically discuss the implementation of the DAQ system.

ENVIRONMENT & INFRASTRUCTURE

The core idea of the facility (multiple laser sources distributed to multiple beamlines over a sophisticated beam distribution system) creates certain conditions for the data acquisition system that are similar, but not equal to classical accelerator facilities:

- The data sources produce bunches of data based on the repetition rate of the driving lasers. Most common are single-shot, 10Hz and 1kHz. There can be more than one laser per experiment, and they are not yet synchronized.
- Characteristic data sources are based on very short pulses (femtoseconds) – requiring high sampling rates. Some lasers are operating with high repetition rates and need to be imaged pulse-by-pulse with cameras and other 2D-datactors.
- The data sources are spatially distributed of 100s of meters and located in various, sometimes challenging or hazardous environments (ionized radiation, EMP, high-vacuum, laser, bio / chemical hazards, cleanrooms up to ISO5).
- Lasers and secondary sources are all developed by different stakeholders (ranging from inhouse research groups, industrial partners – from KMEs to major corporations, national labs, partnering academic labs.). This leads to a heterogeneous environment.
- There are hundreds of cameras, tens of digitizers, and tens of specialized detectors and instruments managed by different stakeholders. While there are efforts to standardize interfaces (for example, for cameras [3]), we still see a wide diversity of interfaces.

The DAQ system has the following tasks:

- Acquisition of data and standardization for further processing and storage.
- Buffering and provision of data for further control tasks (feedback loops etc)
- Data processing (online / offline)

The system also has to provide connections for data storage and integrate the electronic timing system. Facility-wide timestamping and synchronization is a challenge because the “heartbeat” of the facility are multiple, currently unsynchronized laser oscillators using different timing systems. Our solution for electronic timing is based on WhiteRabbit [4] and currently being implemented as described in [2].

STRUCTURAL OVERVIEW

The structure of control and data acquisition system is shown on Fig. 1. This figure identifies the basic room layout relevant to the system, namely main control room, dedicated server room, laser halls, experimental halls, and plant / infrastructure rooms (which are not integrated into the DAQ system, but may receive control feedback).

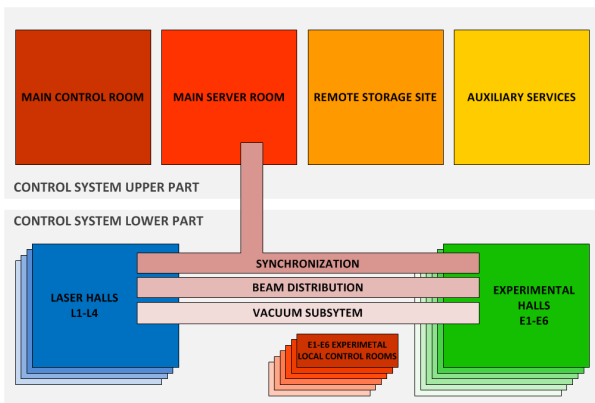


Figure 1: Structural overview.

LOGICAL OVERVIEW

The data acquisition system is divided into two levels, shown in Fig. 2 below:

- **Top level DAQ system**: This system, located in the server room, is responsible for the aggregation and buffering of large amounts of data. It also provides

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCB1

infrastructure for online / offline processing and interfaces for storage

- **Local level DAQ System**: These systems, located in the field (experimental halls / laser halls) provide an interface for the data sources and some capabilities for local processing / feedback loops. They also serve as a real-time interface for the more powerful computational infrastructure of the Top level DAQ system.

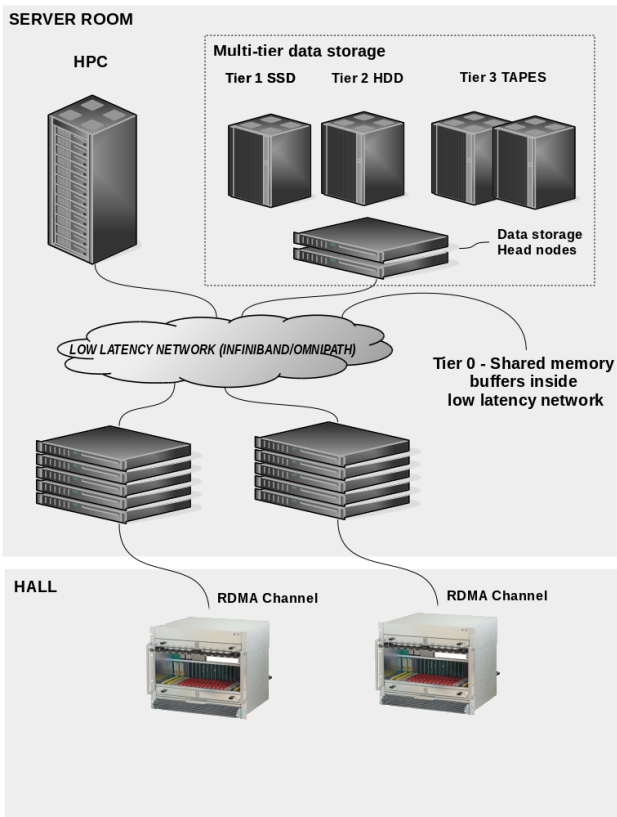


Figure 2: Logical overview of the DAQ system.

TOP LEVEL DAQ SYSTEM

The top level DAQ system is based on 3 data acquisition servers, implemented as a multinode IBM flex system. Each of its seven nodes has the following setup:

- 768GB of RAM (512 GB used for buffer)
- 4 PCIe interfaces for RDMA-enabled NIC cards and FPGA cards
- 24 Cores for further data manipulation and processing
- For each chassis, there is FDR Infiniband infrastructure (2 switches) and Ethernet infrastructure (2 switches, for control and user access).

These servers act as a RAM buffer and provide computational infrastructure.

First Stage: Acquisition and Buffering

Local (Field) data sources are connected to the server point-to-point primarily using the RDMA-enabled NIC cards (10Gb/s).

The data is stored in a pooled RAM buffer (called Tier 0 storage), which is shared using a low-latency network (at the moment: Infiniband 56GB/s, which will be updated to 100GB/s as needed). These low-latency networks cannot work over long distances, but the access via RDMA allows on-demand high-speed distribution of the expensive RAM resources independently of the physical location. This is important because the data load in the facility is highly unbalanced depending on which experiment is running with which lasers.

Second Stage: Acquisition and Processing

In the second stage, data is processed. The CPU cores can be used for this task (trivial standard solution), but we also provide FPGA-based acceleration for online and offline processing.

The field DAQ PCs can access them either using a NIC / RDMA (in this case, a NIC-IP core is implemented on the FPGA of the DAQ server), or using any arbitrary FPGA/FPGA connection. This allows real-time response, and sharing of computational resources (multiple potential inputs per FPGA on the DAQ server, using sx40GB/s interfaces).

Third Stage: Storage

The data then can be saved into data storage which consists of further tiers:

- Tier 0 is the distributed RAM buffer
- Tier1 is set of servers with NVMe in PCIe slots in headnodes of data storage, connected via 100Gb/s Infiniband interface to the DAQ szstem
- Tier 2 is based on conventional harddrives.
- Tier 3 is a tape library for long term data storage.

There are provisions to update this system both horizontally and vertically.

LOW LEVEL DATA ACQUISITION

In the field, we see three types of data sources that are connected to our top-level network:

- Some detectors come with their own inbuilt data aggregation and pre-processing systems
- **PCIe-based DAQ computers** (*Supermicro*) with 128GB of RAM, 24 cores and 10 PICex8 slots. These servers are low-cost, can provide large memory buffers (terrabytes). There is a wide variety of PCIe-cards for different applications.
- **MTCA-based DAQ systems**, which have the advantage of clock support in the backplane and allow card-to-card-connection without involving the CPU.

As already discussed in [2], we want to use the advantages of both Supermicro and MTCA systems (clock support, card-to-card-connections, large memory buffers). Our MTCA-MCH (NAT-MCH-PHYS80) allows to connect its internal PCIe switch through optical cable to PCIe based local DAQ server. The connection setup

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCB1

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

(shown in Fig. 3) provides PCIe x16 interface and can be implemented using PCIe on MTCAs' agnostic backplane and its fat pipes.

Figure 3: Connection between MTCA and PC based system.

OUTLOOK

The technology surrounding DAQ systems is evolving: Low latency (which we use for the shared RAM buffer) is approaching throughputs of 200Gb/s. FPGA technology is accelerating as well, and we see transceivers that are reaching speeds of 30Gb/s.

For such data rates, the PCIe gen.3.0 interface used in our servers will become a bottleneck. Therefore, we anticipate that we will progress to PCIe gen. 4.0, which is unfortunately not yet supported by Intel-based processors – but already supported by current FPGA chips and Open Power processors.

Coherent access to memory (CAPI, OpenCAPI) is also a topic we are actively monitoring and preparing. We have installed a POWER8 server with CAPI enabled interfaces, one FPGA with CAPI support and one Infiniband card for development and evaluation.

This setup seems to be promising as both CAPI and PCIe gen 4.0 are supported by the new Power9 processor generation.

REFERENCES

[1] ELI Beamlines, https://www.eli-beams.eu

[2] P. Bastl *et al*, "Hardware architecture of the ELI Beamlines Control and DAQ System", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017. doi: 10.18429/JACoW-ICALEPCS2017-FRAPL05

[3] B. Plötzeneder, V. Gaman, O. Janda, P. Pivonka, P. Bastl, "Cameras in ELI Beamlines: A standardized approach", in *Proc. ICALEPCS'2017*, Barcelona, Spain, Oct. 2017. doi: 10.18429/JACoW-ICALEPCS2017-THMPL06

[4] The White Rabbit Project, https://www.ohwr.org/projects/white-rabbit

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCB2

DESIGN AND CONSTRUCTION OF THE DATA WAREHOUSE BASED ON HADOOP ECOSYSTEM AT HLS-II*

Y. Song, X. Chen, C. Li, K. Xuan, J. Wang, G. Liu†, NSRL, USTC, Hefei, Anhui 230029, China

Abstract

A data warehouse based on Hadoop ecosystem is designed and constructed for Hefei Light Source II (HLS-II). The ETL program based on Spark migrates data to HDFS from RDB Channel Archiver and the EPICS Archiver Appliance continuously and store them in Parquet format. The distributed data analysis engine based on Impala greatly improves the performance of data retrieval and reduces the response time of queries. In this paper, we will describe our efforts and experience to use various open sources software and tools to effectively manage the big data. We will also report the plans on this data warehouse in the future.

INTRODUCTION

The Hefei Light Source (HLS) at National Synchrotron Radiation Laboratory is the first dedicated synchrotron radiation facility in China, which provides radiations from IR to vacuum ultraviole (VUV) for various user programs. HLS was upgraded from 2010 to 2015 to improve its performance. The upgraded light source is renamed as Hefei Light Source II (HLS-II) [1]. It is comprised of an 800MeV linac, an 800MeV storage ring and a transport line connecting the linac and the storage ring. The HLS-II was fully open to users in January 2016. The control system of HLS-II is a distributed system based on Experimental Physics and Industrial Control System (EPICS) [2].

With the continuous upgrade, a series of data archiving tools has been used in HLS-II supported by the EPICS community, such as Channel Archiver, RDB Channel Archiver and the EPICS Archiver Appliance. As more and more data have been stored, the exciting data archiving tools can not meet our requirements of data query and processing. In order to deal with these problems, we are designing and constructing the data warehouse based on Apache Hadoop ecosystem. In addition, our laboratory is conducting pre-research on Hefei Advanced Light Source (HALS) project [3]. Compared to HLS-II, HALS is a larger and more complex synchrotron facility. The work described in this paper also provides technical verification for the future construction of HALS.

This paper will describe our efforts and experience to use various open sources software and tools to manage the big data effectively. We will also report the plans on the data warehouse in the future.

* Work supported by National Natural Science Foundation of China (No.11375186)
† Corresponding author, gfliu@ustc.edu.cn

HADOOP ECOSYSTEM AND DATA WAREHOUSE

Hadoop Ecosystem

Apache Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to slove problem involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data [4]. As shown in Figure 1, Hadoop core components include HDFS, Yarn and MapReduce, as well as some open source projects based on Hadoop, including Spark and Impala, which provide necessary support for the whole life cycle of big data processing. Our warehouse system mainly uses the following components:

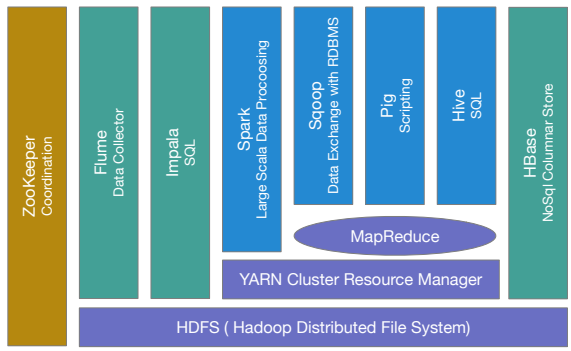


Figure 1: Hadoop Ecosystem.

HDFS HDFS is a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.

Spark Spark has the advantages of Hadoop MapReduce, but unlike MapReduce, its intermediate output can be stored in memory, instead of the HDFS. Tests have shown that Spark is more efficient than MapReduce for the same operation. Spark is widely used in the field of ETL (Extract-Transform-Load) and machine learning [5].

Impala Impala is an open source new query system that can query PB-level big data stored in Hadoop's HDFS and HBase. Compared to MapReduce, Impala provides data query function more efficient and convenient.

Sqoop Sqoop is command-line interface application for transferring data between relational databases and Hadoop.

Hadoop Configuration

Because the Hadoop ecosystem contains a variety of software, it is difficult to handle version compatibility and config-

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCB2

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

uration files, and is also inconvenient to manage and monitor the state of the entire Hadoop cluster. Therefore, Cloudera Distribution Hadoop (CDH) is used as the tool for cluster construction. CDH is an open-source Apache Hadoop distribution provided by Cloudera Inc. and is the most complete, tested, and widely deployed distribution of Apache Hadoop [6].

We built the test system with 5 nodes on the VMware vSphere virtualization platform based on CDH. Figure 2 shows the management of web pages provides by CDH. Through this web page, the cluster administrators can simply complete the management work, such as adding or deleting services, adding computer nodes, and viewing the status of the cluster, without paying attention to the details of the configuration file.

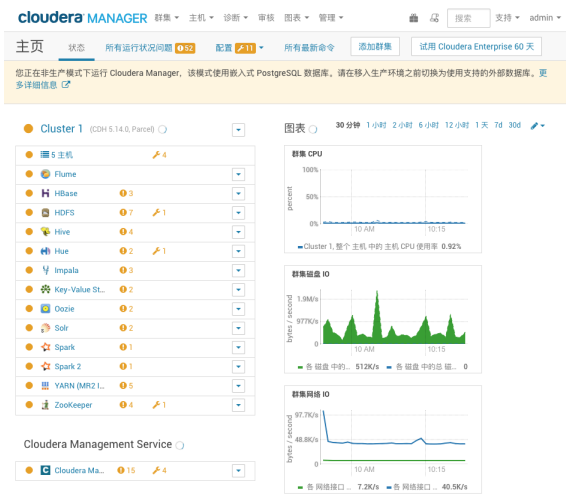


Figure 2: Web pages of cloudera manager provided by CDH.

SOFTWARE ARCHITECTURE

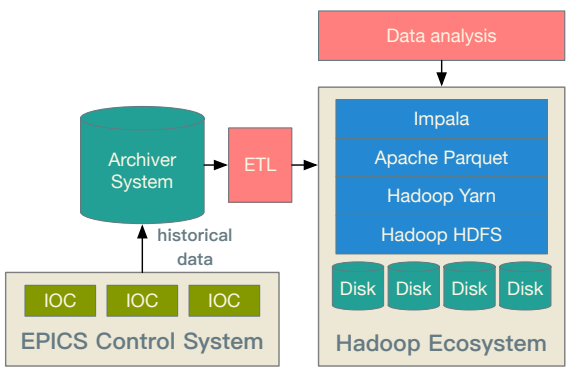


Figure 3: Software structure of the entire data warehouse.

Figure 3 shows the software structure of the entire data warehouse. The whole system is mainly divided into three parts: the control system based on EPICS, the big data platform based on Hadoop Ecosystem and the archiver & ETL program. The Records in EPICS IOC hold the real-time state of the controlled devices. Instead of redeveloping data acquisition and archiving software, we are still using data archiving tools already available in the EPICS community, such as RDB Channel Archiver and Archiver Appliance. These data archiving tools collect historical data and store them in its own storage system in a custom format. ETL program is developed to migrate data from archiving tools to HDFS and store them in Parquet format. Impala provides a convenient data analysis interface for scientists and engineers.

IMPLEMENTATION OF DATA WAREHOUSE

Archiving Tools at HLS-II

At HLS-II, there are two data archiving systems running at the same time: Archiver Appliance and RDB Channel Archiver respectively. The Archiver Appliance collects data from IOC and stores them on the hard disk in Protocal Buffer (PB) format [7]. The Archive Engine of RDB Channel Archiver is also used for collecting data from IOCs of control system and storing them in Oracle database [8]. Both types of data from different data sources need to be migrated to HDFS.

ETL Program

Data migration refers to transferring the original data stored in the archive system to the HDFS file system, storing them in Parquet format, and performing data cleaning and calculation as needed during the transfer process. Usually this process is called Extract-Transform-Load (ETL). The ETL program is the core part of connecting archiving tools and the Hadoop Ecosystem.

The whole process of data migration is shown in Figure 4. Spark's APIs are easy to use and have good support for accessing various databases. Therefore, Spark is very efficient in big data processing and is used to implement the ETL process. The process of data migration is divided into two steps.

In the first step, the original data in the archiving tools are exported to the distributed file system HDFS. For the Archiver Appliance, it provides an HTTP interface for querying data, so a Python program is developed to get data from the Archiver Appliance via HTTP in CSV format and upload the CSV files to the HDFS file system. This process is performed periodically in batch mode. For the RDB Channel Archiver, Sqoop is used to migrate the data from Oracle database directly to the HDFS system.

In the second step, The ETL program written based on Spark is triggered after the data are uploaded. It completes the ETL process, and finally export the Parquet format file into the data warehouse with a predefined data model. In this process, the ETL program also performs a series of data cleaning tasks, including the elimination of abnormal data points and the alignment of sampling time.

In addition, the workflow management program is written to ensure the stable operation of the ETL process, including the workflow task scheduling, metadata acquisition, logging

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCB2

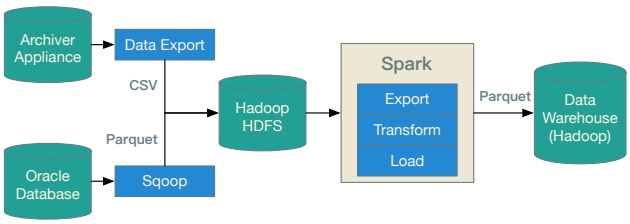


Figure 4: Data migration process.

and notification function. The open source community has released an library Apache Airflow for building complex batch workflows. This system use Python language to develop tthe workflow management program based on Apache Airflow to control the whole workflow.

Data Analysis

After the data are stored in HDFS , the users can conveniently use Impala to perform data query and analysis in the Parquet format. Figure 5 shows the screenshot of the query data through the Impala command line. The user queries the first 10 records of the sample table in the css database through SQL statement and the result is returned within 0.03s.

```
[root@node01 ~]# impala-shell --impalad=192.168.113.42
Starting Impala Shell without Kerberos authentication
Connected to 192.168.113.42:21000
Server version: impalad version 2.11.0-cdh5.14.0 RELEASE (build d68206561bce6b26762d62c01a78e
6cd27aa7690)
***********************************************************************************
Welcome to the Impala shell.
(Impala Shell v2.11.0-cdh5.14.0 (d682065) built on Sat Jan  6 13:27:16 PST 2018)

When pretty-printing is disabled, you can use the '--output_delimiter' flag to set
the delimiter for fields in the same row. The default is ','.
***********************************************************************************
[192.168.113.42:21000] > use css;
Query: use css
[192.168.113.42:21000] > select channel_id, smpl_time, float_val from sample limit 10;
Query: select channel_id, smpl_time, float_val from sample limit 10
Query submitted at: 2018-10-08 09:42:13 (Coordinator: http://node02.cdh.nsrl:25000)
Query progress can be monitored at: http://node02.cdh.nsrl:25000/query_plan?query_id=924247bf
c18d2302:380a2e2f00000000
+------------+---------------+-------------------+
| channel_id | smpl_time     | float_val         |
+------------+---------------+-------------------+
| 1596       | 1525418774966 | 29.74092864990234 |
| 1540       | 1525418773003 | 36.93181228637695 |
| 1088       | 1525418773014 | 30.86883926391602 |
| 984        | 1525418774958 | 29.26275825500488 |
| 1100       | 1525418773027 | 25.61181831359863 |
| 458        | 1525418775031 | 23.49545860290527 |
| 1671       | 1525418774013 | 27.05657958984375 |
| 826        | 1525418799040 | 86.629150390625   |
| 1502       | 1525418834977 | 35.76647186279297 |
| 1596       | 1525418834965 | 29.72908020019531 |
+------------+---------------+-------------------+
Fetched 10 row(s) in 0.03s
```

Figure 5: Query data through the Impala command line.

CURRENT PROGRESS AND NEXT PLAN

At present, we have built the test system on the VMware vSphere virtualization platform and developed the ETL program. In the next step, we will deploy the program to the server in the production environment, and carry out performance testing.

In the future, we plan to complete the following tasks:

- Provide data reporting functions. We will develop a series of programs for calculating data reports based on Impala, such as operating status statistics and integral current calculations. With the efficient computing power of the Hadoop cluster and the optimization of the SQL statement, our goal is to control the response time of the report calculation task within a few seconds.
- Provide more convenient data query entry. Currently, users can only query data by executing SQL statements on the Impala command line. In the future, we plan to provide a web-based data query interface and develop the web applications to facilitate scientists and engineers to query data stored in the data warehouse.
- Carry out research on data mining and Artificial intelligence (AI) algorithm applications. As Spark's machine learning library, MLlib consists of some common learning algorithms and tools, including classification, regression, recommendation, etc.. We plan to use the spark MLlib to mine the characteristics of the data stored in the data warehouse, carry out research on AI algorithm, and improve the intelligence of the entire control system.

CONCLUSION

A data warehouse based on Hadoop ecosystem is designed and constructed for Hefei Light Source II (HLS-II). The ETL program based on Spark and Sqoop migrates data to HDFS from RDB Channel Archiver and the EPICS Archiver Appliance continuously and store them in Parquet format. Such a technology plan combines various open sources tools under Hadoop framework and the archiving tools in EPICS community. It provides a sulotion to solve the big data problem for the large-scale scientific facility.

REFERENCES

[1] J. Y. Li, G. Huang, W. Wei, *et al.*, "Operational Status of HLS-II", in *Proc. IPAC'16*, Busan, Korea, May 2016, pp. 4155–4158. doi:10.18429/JACoW-IPAC2016-THPOY028

[2] C. Li, J. Wang, K. Xuan, *et al.*, "Event-driven timing system based on MRF cPCI hardware for HLS-II", *Nuclear Science and Techniques*, vol. 26, pp. 060401, 2015. doi:10.13538/j.1001-8042/nst.26.060401

[3] L. Wang, Z. Bai, N. Hu, *et al.*, "Hefei Advanced Light Source: A Future Soft X-Ray Diffraction-Limited Storage Ring at NSRL", in *Proc. IPAC'18*, Vancouver, BC, Canada, Apr./May 2018, pp. 4598–4600. doi:10.18429/JACoW-IPAC2018-THPMK120

[4] Hadoop home page, http://hadoop.apache.org/.

[5] Spark home page, https://spark.apache.org/.

[6] Cloudera CDH home page, https://www.cloudera.com/products/open-source/apache-hadoop/key-cdh-components.html

[7] Y. F. Song, C. Li, K. Xuan, *et al.*, "Automatic data archiving and visualization at HLS-II", *Nuclear Science and Techniques*, vol. 29(9), pp. 129, 2018. doi:10.1007/s41365-018-0461-6

[8] J. Qiang, G. Liu and Jie Qiang, "The data archive system and preliminary applications of Hefei light source based on Oracle", *Nuclear Techniques*, vol. 36, pp. 1–5, 2013.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCB4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

THE APPLICATION FOR FAULT DIAGNOSIS AND PREDICTION OF POWER SYPPLY CONTROL DEVICE ON BEPCII

Jia Liu, Di Wang, Jincan Wang, XiaoLi Wang
Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China

Abstract

With the widely adoption of complex electronic devices and microcircuits in accelerator system, the probability of system failure and functional failure will be enlarged. For example, the fault of the magnet power supply front-end electronics devices may cause accelerator energy instability and even lead to beam loss. Therefore, it is very necessary to diagnose and locate the device fault accurately and rapidly, that will induce the high cost of the accelerator operation. Faults diagnosis and prediction can not only improve the safety and reliability of the equipment, but also effectively reduce the equipment's cycle costing. We applied the FMECA and testability modelling method for the PSI device, which using in BEPCII power supply control system, and evaluated the remaining life of the PSI under certain temperature and humidity condition based on the reliability model and accelerated life test.

INTRODUCTION

The complex electronic devices and microcircuits used in accelerator system is often the first part of the system to fail and lead to shorter service life of equipment or higher cost of system maintenance and operation steadily. In the Beijing Electron Positron Collider (BEPCII), hundreds of magnet power supplies are installed to provide power for various types of electromagnets, and the current of the power supply can be adjusted to change the magnetic field and thus control the orbit of particles. During the operation of the accelerator, failure of massive electronic devices may cause the accelerator energy to be unstable and even cause beam loss.

The traditional method to solve these problems is the fault diagnosis technology. However, fault diagnosis is generally only applied to the current fault status of the equipment. That is, the type and location of the fault are analyzed and diagnosed after the equipment fails, and cannot be pre-warned before the failure occurs. Besides, the cause of the failure of the electronic system is usually the combination of multiple failure modes and environmental stress. In actual, the issue of how to monitor the working status of electronic equipment in real time and provide early warning of possible faults is of great importance to improve the stable and reliable operation of the entire accelerator system.

TESTBILITY MODELLING

The testability modelling is a method to analyze the failure related information of equipment, which include two part at least. One is the Failure modes, effects, and criticality analysis(FMECA), that is a methodology to identify and analyze a product or process in order to determine all possible failure modes of various components in the system, how these failures affect the system, and what measures can be taken to avoid failures occurrence or reduction of the impact of a fault on the system [1]. In general, the method is to identify early in the product or process design all manner of failures so they can be eliminated or their impact reduced through redesign at the earliest possible time.

The other is testability modelling based on multi-signal flow graph [2], which combines the available test point configurations by analyzing the failure modes of each component in the system and the causes and effects of each failure mode. The Dependency Matrix $D=[d_{ij}]$ ($1 \le i \le m, 1 \le j \le n$, where m and n denotes the totality of the source of failure and the set of testing respectively) of whole device was established, and the Fault diagnosis is to find the candidate set of faults [3] based on the structure of multi-signal model. The testability modelling report is shown in Fig. 1. Because of the reliability of the constituent units is not same in practice system, it is necessary to ament the fault probability. There are two methods that can be used when correcting, Probabilistic Equalization Method and Probabilistic Priority Method. The Probabilistic Priority method is including the failure mode more comprehensively, and the obtained analysis result is better than the Probabilistic Equalization Method.

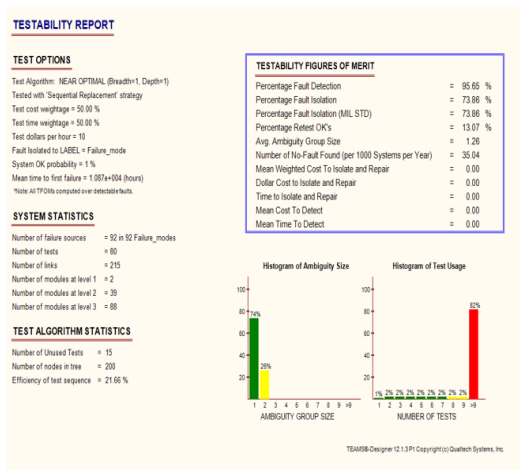


Figure 1: The testability report of power supply interface.

FAILURE PHYSICAL SIMULATION

Reliability Simulation, which based on Physics of Failure [4], is a reliability theory for establishing the relationship between failure and environmental stress from mechanism. Because of Temperature and vibration are the main factors that affect the reliability of electronic products, so

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCB4

vibration modeling and thermal modeling were performed for PSI. The result of simulation is shown in Fig. 2 and Fig. 3. The simulation analysis illustrate that the failure mechanism of PSI is the slight cracking of solder caused by the heating of connector components. The main function of this connector is to connect the voltage output from the power module to the interface board module, and 60% of the failure of PSI are related to this connector during the actual operation of BEPCII.

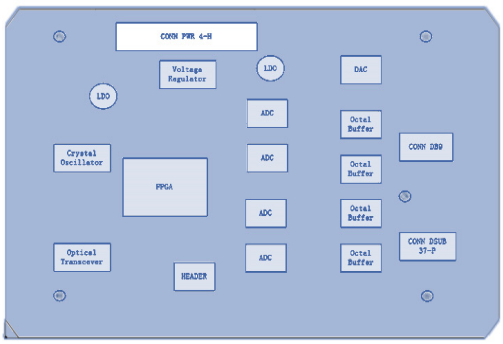


Figure 2: The vibration modelling of power supply interface.

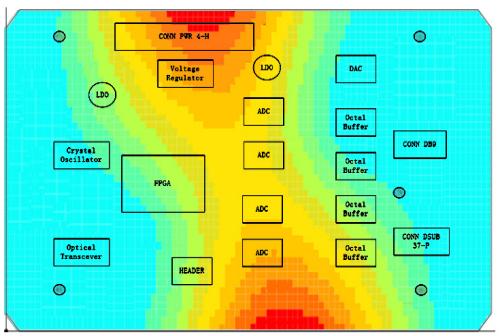


Figure 3: The thermal modelling of power supply interface.

REMAINING LIFE PREDICATE

The connector is located as the weak link of the whole PSI through the simulation, and then the Accelerated Degradation Test(ADT) is used to test the degradation process by increasing environmental stress without changing the mechanism of failure. ADT was carried out in a temperature and humidity alternating test box, several groups of temperature and humidity conditions were set up [5]. An oscilloscope is used to acquire and store the voltage that can characterize the connector. The degradation model was used to analyze the data obtained by the ADT. Based on the physical statistical model, the failure life can be associated with acceleration variables and the life or degradation rate of the equipment under normal operation can be estimated [6]. Usually, there are three degradation models for electronic equipment as Eq. (1)~(3),

$$y = \alpha_i + \beta_i t, \quad (1)$$

$$\log(y_i) = \alpha_i + \beta_i t, \quad (2)$$

$$log(y_i) = \alpha_i + \beta_i \log(t), \quad (3)$$

We chose the third model to fit the voltage experimental data of the PSI connector and obtain the degradation curve of equipment performance. The model under multiple-stress that considers the temperature and humidity conditions [7] is consistent with the logarithmic normal distribution:

$$\varepsilon = \beta_0 + \frac{\beta_1}{t+273.15} + \frac{\beta_2}{H} \quad (4)$$

The remaining life of PSI can be predicted by the reliability distribution curve, which is shown in Fig. 4.

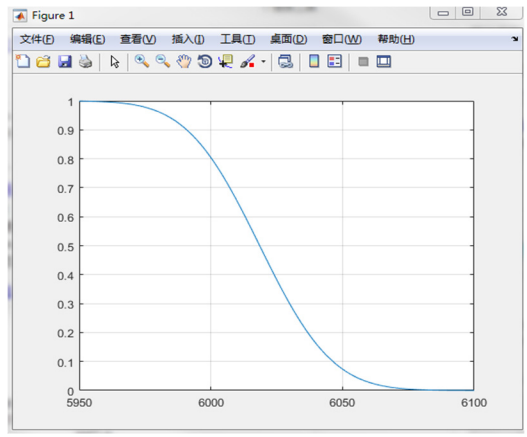


Figure 4: The algorithm normal distribution reliability of power supply interface.

SUMMARY

This paper introduces the basic workflow for fault diagnosis and prediction based on the PSI, the front-end electronic device used in BEPCII. It is used to obtain information about the faults of each component of the system through testability modelling analysis. And then perform the physical simulation of the system to determine the weak link of equipment. Based on the accelerated degradation test and reliability model, we can predict the remaining life of the device. However, the structure and function of PSI are relatively simple and the sample size of experimental data is too small, the large number of online data should be obtained to verify the validity of the prediction model.

REFERENCES

[1] Charles E. Ebeling, An Introduction to Reliability and Maintainability Engineering, Waveland Pr Inc; 2 Har/Cdr edition, 2009.

[2] S. Deb, K. R. Pattipati, V. Raghavan *et al.*, "Multi-signal flow graphs: a novel approach for system testability analysis and fault diagnosis", *Aerospace & Electronic Systems Magazine IEEE*, vol. 10, no. 5, pp.14-25, 1995.

[3] Di Wang, Jia Liu, Xiaoli Wang, "Research on Fault Diagnosis of Power Supply Control System on BEPCII", in *Proc. ICALEPCS'17*, Barcelona, Spain, paper TUPHA001.

[4] Liang Qiao, Chuanri Li, Longtao Liu, "Application of Reliability Simulation Test Based on Fault Physics", *Equipment Environmental Engineering*, vol. 09, no.2, pp. 7-11, 2012.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCB4

[5] Kaitai Fang, Orthogonal and Uniform Experimental Design, Science Press, 2001.

[6] V. Crk, "Reliability assessment from degradation data", *in proc. of Reliability and Maintainability Symposium*, Los Angeles, USA, 2000.

[7] Zhanqiang Jia, Jinyan Cai, Yuying Liang, *et al*. "Research on Fault Prediction of Electronic Equipment Based on Integrated Environmental Accelerated Life Test", *Journal of Electronics*, vol. 37, no. 6, pp. 1277-82, 2009.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC1

FPGA BASED IMAGE PROCESSING SYSTEM FOR ELECTRON BEAM WELDING FACILITY*

M. M. Sizov†, A. A. Starostenko, K. A. Blokhina, A. M. Medvedev
Budker Institute of Nuclear Physics of the Siberian Branch of the RAS, Novosibirsk, Russia

Abstract

Automated electron beam welding (EBW) facilities require real-time control systems. EBW facility at BINP has recently renewed its control unit. High-level part of control system runs as Linux user-space application written in Erlang, while real-time subsystem runs on Field Programmable Gate Array (FPGA) for signal processing and welding control. Time-sensitive algorithms such as joint finder algorithm and pipelined data filters implemented in VHDL and dataflow DSL language Caph. DSL eases implementation and testing of algorithms and translates code efficiently to HDL. Control system filters noise and calculates a joint location for weld autocorrection within 2 µs.

INTRODUCTION

Budker Institute of Nuclear Physics has production of electron beam welding facilities (EBW) for low series production (such as manufacturing of vacuum chambers for FAIR project [1]) and for materials experiments such making hafnium carbide [2]. The largest of all facilities built has cylindrical vacuum chamber 3.5 m long and 0.98 m diameter, it has a coordinate table inside with achie-vable area of 1970×300 mm. Electron gun's acceleration voltage is 60 kV and power of its beam is up to 60 kW. It has two fore-vacuum pumps and three turbomolecular pumps. The emitter is directly heated tantalum cathode and the magnetic coordinate system has two pairs of correctors, that can deflect the beam in two dimensions. Secondary emission electrode is located below electron gun and could be used for constructing of a height map of sample welded or as a joint finder.

Foundation of the control system is a commercially available single board computer DE1-SoC. On board is Intel Cyclon V system on chip (SoC) with dual-core ARM processor and field gate programmable array (FPGA), 1 Gb RAM, ADC, Ethernet, USB. Having standard ARM processor on board allows to easily reuse thousands of open source software packages ported to ARM architecture including operating system, programming languages build tools, message brokers.

FPGA can process data in parallel – different subsystems run simultaneously, each of subsystems constructs a pipeline. Thus, the time needed for data processing (and react to inputs) is the largest of pipeline latencies, which is fixed and could be easily calculated. FPGA's generally have 20-300 K of logic elements, 1-12 Mb of memory blocks, 25-300 DSP blocks which altogether allow implementation of complex real-time algorithms.

* work supported by Russian Science Foundation (project N14-50-00080)
† email address: sizov.m.m@gmail.com

High-level control system part is written in Erlang functional language that has a number of features that ease implementation of the control system [3]. It has language concepts for binary data decomposition. And actor-model parallelism allows to write loosely coupled finite state machines. Erlang ships OTP templates library, that features patterns such as FSM and process monitors.

Although DE1-SoC is great in many aspects, it has limitations too. It doesn't have SATA connectors, so only possible options for storage are microSD and USB drives. It has VGA port, but its connected using FPGA fabric, so having user interface on system would take extra FPGA space. Memory and calculation intensive algorithms are not suited well for CPU processors as well as FPGA. To mitigate these limitations, we implemented "thin" client on regular desktop machine, that connects to control system that is able to store data and build surface heights approximation for the operator. Interface is built with ElectronJS framework and uses React for UI widgets and logic.

IMAGE ACQUISITION

EBW facility is able to weld long titanium, aluminum or steel samples. Electronic gun is stationary so coordinate table moves the sample. Having surface height map, operator enters sequence of coordinates and parameter values at key points such as current of heater, corrector coils or anode. To weld a joint operator needs to set two or more points on sample. Then control system sets high current on cathode heater, starts to move sample according to entered sequence and in case of long sample welding corrects deviation of joint position from program.

Figure 1 shows a block scheme of hardware setup. Between electronic gun and coordinate table we have a magnetic system, that is able to deflect the beam. To construct the surface map of the sample under the electron gun, user turns on electron beam with low heater current, then Erlang control system programs FPGA to repeatedly write periodic waveform signal to magnetic system. Magnetic system has corrector coils that deflect the beam in two dimensions. Electrons hit the surface and scatter. Fraction of reflected electrons hit secondary emission electrode. FPGA system synchronizes data from coordinate table, secondary emission electrode and deflection angle that was written into magnetic system. The principle is the same as in electron microscopes (but with much higher beam intensity and subsequently noises). FPGA preprocesses image scan lines with pipelined algorithms to reduce a noise. Prepared data is passed into joint finder FPGA [4] submodule that calculates joint position in the same units as magnetic deflection angle.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC1

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

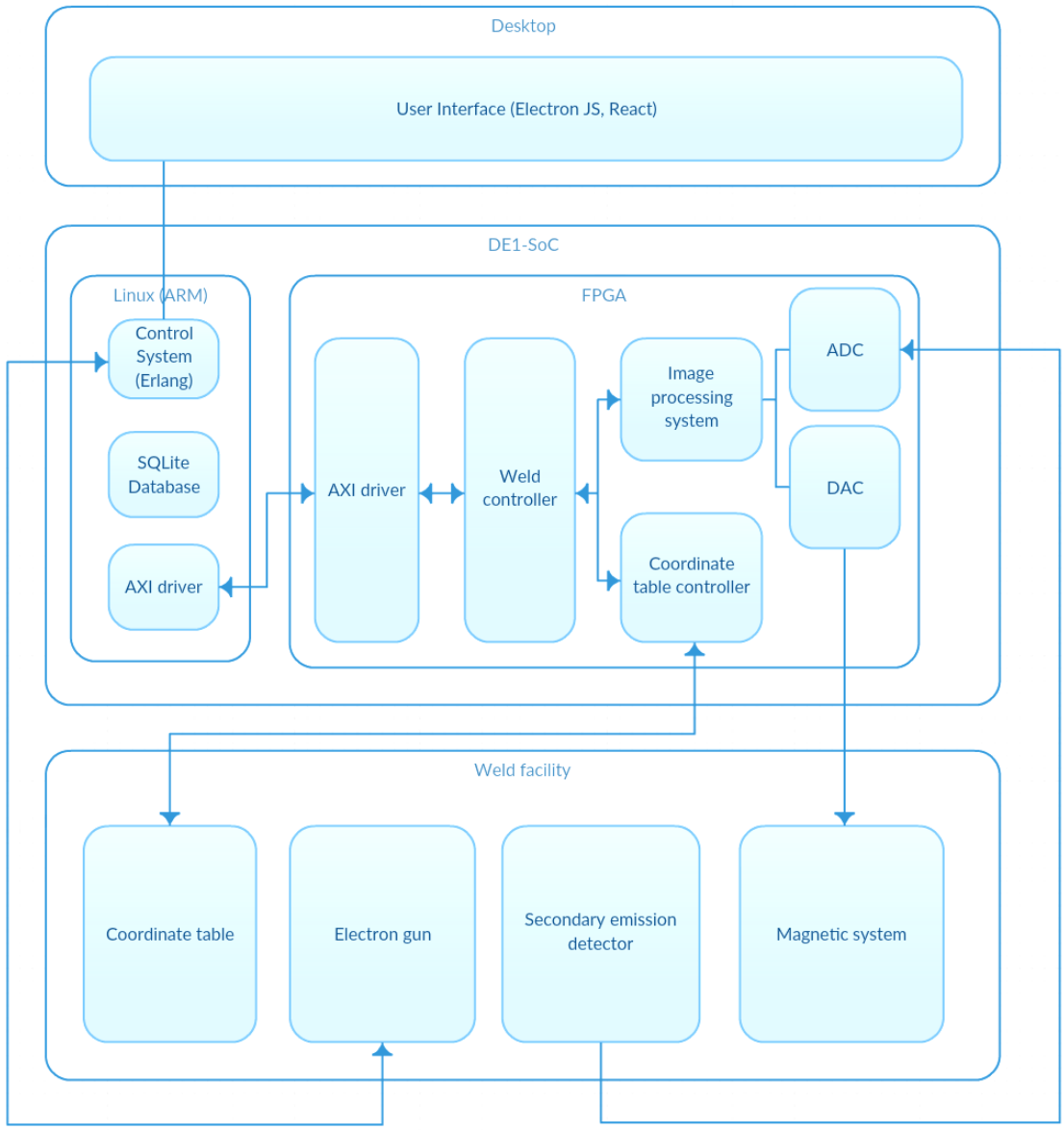


Figure 1: Block scheme of control system data flows.

Passing this position in a negative feedback loop to magnetic system compensates operator precision errors, vibrations and heating deformations. Preprocessed image line is sent to Erlang code, that stores data in DB and passes it through into ElectronJS interface.

Interface part accumulates image lines and binds height information to table coordinates. Current FPGA system has up to 500 of pixels per line with resolution from 60 µm per pixel up to 2 µm per pixel, although the minimal achievable beam spot is approximately 1 mm in diameter. 2 meters with 2 µm per pixel resolution would be a million of pixels which is not reasonable to work with. We artificially limit spatial resolution to 0.1 mm per pixel in long axis direction that makes working with picture feasible. As we have JavaScript as UI base, we had a need to speed up manipulations with image data which can be up to 16000×500 pixels. We have implemented WebGL shader for performance improvement. Shaders are parallel and executed in GPU processing units, algorithms implemented include zoom, pan and data filters such as FFT, median filter, and special filter that equalizes intensity of different rows between each other.

FPGA BEAM DIAGNOSTICS SYSTEM

FPGA is also used for the beam diagnostics system is based on pinhole probing method. The size of the pinhole is very small compared to the minimal beam diameter and to capture the beam characteristics, the beam is raster scanned over the hole. The beam current is collected by Faraday cup under the pinhole and through the resistor the ADC signals for each beam cross-section are obtained. In order to get better resolution the sets of signals are processed by filter method. This system is being developed for obtaining the beam power density distribution, which is the most important parameter of the beam used in welding applications.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC1

The power density distribution maps are built from at least two beam projections. This scanning method allows obtaining a number of beam cross-sections in different projecting planes. The method of processing of data to be used for mapping the power density distribution is at the testing stage. The transversal cross-sections of the beam are shown on Figure 2. This scanning method with high accuracy allows characterizing the beam shape and transversal size. The measured beam size is 1.3 mm.

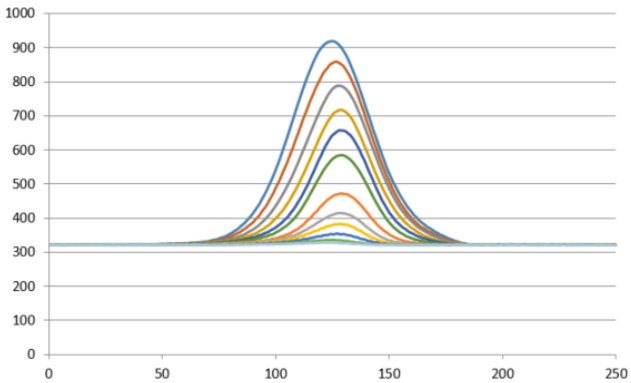


Figure 2: Beam cross-sections received by scanning with 0.015 mm per pixel (x axis corresponds to index in image line, and y axis shows the signal in mV).

FPGA IMAGE PROCESSING SYSTEM

Caph [5] is a high-level domain specific language which translates to VHDL. Dataflow paradigm utilized in this language exploits true parallelism of FPGA and ability to construct pipelines. Processing of sample heights starts with reducing of electromagnetic noise from 60 kV power source. Median filter removes spikes and a low-pass filter reduces high frequencies. Algorithms written on Caph describe algorithm as network of actors with inputs and outputs. On Figure 3 example of such graph is shown for median filter. ADC measurements are abstracted as stream of structured data that is either start of line, valid value or end of line. Values are consumed at actor inputs. Each actor is independent state machine and executes algorithms in parallel (as this code runs in FPGA), computed outputs are distributed to each input connected.

This computational model is declarative (as well as hardware description languages are) so it has a little overhead compared to hand-written code. As this language is functional, developer can execute and functionally test code and compute latencies on desktop PC without having to simulate hardware on register lever or compile FPGA firmware.

CONCLUSION

In this project, data is processed in each layer – FPGA, on Erlang control system and in user interface. Control system features ability to perform beam diagnostics, set arbitrary waveforms and provides relative heights of welded sample. Operator uses data acquired during scanlines to properly set welding points and weld samples. Arbitrary scanlines

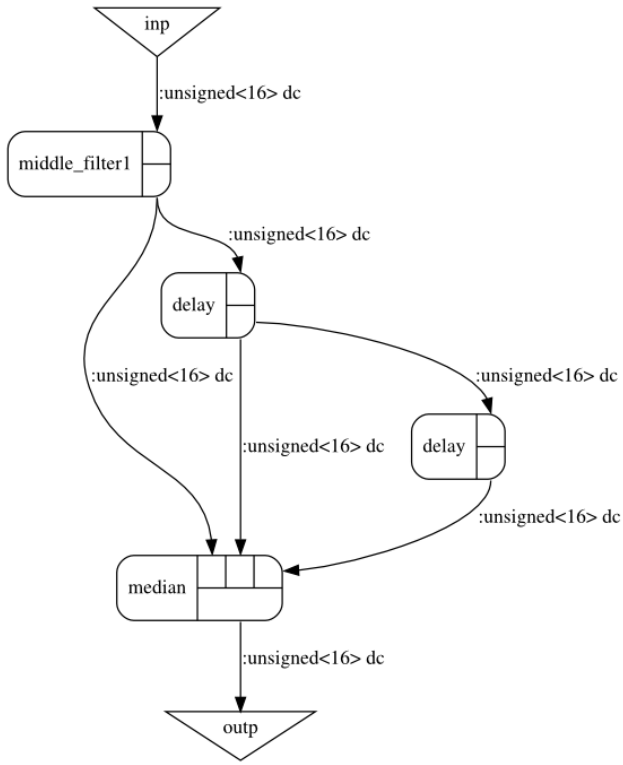


Figure 3: Calculation graph for median filter written in Caph.

support various experiments on quality of seams. In addition, joint-finder module was developed using domain-specific language Caph. Module compensates difference between joint position and center of welding scanline.

REFERENCES

[1] A. I. Ancharov, N. F. Grigoreva, P. V. Logachev, Iu. I. Semenov, A. A. Starostenko, B. P. Tolochko, "Possibility of application of hafnium and tantalum carbides as materials for additive manufacturing," The international seminar on interdisciplinary problems in additive technologies. 6-9 December, 2016, Tomsk.

[2] A. M. Medvedev, A. M. Semenov, Y. I. Semenov, A. A. Starostenko, M. M. Sizov, A. S. Tsygunov, "EBW application for the manufacture of HEBT dipole vacuum chambers, FAIR," *"E+E"*, 2016, p. 148–152.

[3] K. Kruger and A. Basson. "Erlang-based control implementation for a holonic manufacturing cell," *International Journal of Computer Integrated Manufacturing*, vol. 30.6, pp. 641–652, 2017.

[4] M. M. Sizov, A. A. Starostenko, A. S. Tsygunov, A. M. Medvedev, "Real time FPGA-based seam finder for EBW facility," *Electrotechnica & Electronica E+E*, The Union of Electronics, Electrical Engineering and Telecommunications /CEEC/, 2018, vol. 53, no. 5–6, pp. 145–148.

[5] J. Sérot and F. Berry, "High-Level Dataflow Programming for Reconfigurable Computing," *International Symposiumon Computer Architecture and High Performance Computing Workshop (SBAC-PADW)*, 2014.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC2

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

CONTINUOUS BEAM SCANNING INTENSITY CONTROL OF A MEDICAL PROTON ACCELERATOR USING A SIMULINK GENERATED FPGA GAIN SCHEDULED CONTROLLER

P. Fernandez Carmona, V. Minnig, G. Klimpki, D. Meer, M. Eichin, C. Bula, D. C. Weber, S. Psoroulas, Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

Abstract

At the Center for Proton Therapy at the Paul Scherrer Institute cancer patients are treated with a fixed beamline and in two gantries for ocular and non-ocular malignancies. The gantries use a step-and-shoot technique to deliver dose covering the treatment volume with a grid of weighted proton bunches with different energies. Dose delivery for tumours moving under respiration is however challenging and not routinely performed, because of the interplay between patient and beam motions. At the Gantry 2 treatment unit, we are implementing a novel continuous beam modulation concept called line scanning, aiming at realizing a faster dose delivery to allow for effective organ motion mitigation techniques such as rescanning, gating or breath-hold. The beam current should stabilise within 100 µs, which is challenging due to the non-linearity of the system and latency of the dose monitors. In this work we implemented a gain scheduled controller and a predictor by modelling the accelerator in Simulink and developing a controller using the frequency domain robust method. We used Mathwork's HDL Coder functionality to generate VHDL code that was implemented in an FPGA in the gantry control system. Latency, overshoot and dosimetry performance improved considerably compared to a classic PID controller.

INTRODUCTION

At the center for Proton Therapy in PSI we are treating cancer patients using spot scanning techniques since 1996 [1], which opened the possibility for advanced tumour treatments with protons [2, 3]. In order to push technology forward and open the possibility to treat additional sites (moving tumours like liver and lung), we are developing in our Gantry 2 a new beam delivery modality: continuous line scanning of tumours [4]. Such a technique requires controlling the intensity of the proton beam with a high precision at the 1% level, especially during fast changes, expected to happen on a sub-ms time scale.

The facility generates the proton beam with a 230 MeV superconducting cyclotron, jointly developed between PSI and the company Accel [5].The intensity of the proton beam can be modulated near at the center of the cyclotron using two vertical deflector plates (VD) that can select how much of the beam can pass through a set of collimator slits by applying a transversal electric field [6]. In this way the radiation activation of the accelerator is minimized, as only the desired amount of protons gets accelerated to the nominal energy.

The electric field of the VD is driven by a high voltage power supply. The treatment delivery system (TDS) of the gantry can control the beam intensity by sending the appropriate voltage set point to this power supply. The intensity of the beam that is delivered to the patient is measured with an ionization chamber monitor located at the nozzle of the gantry. Figure 1 shows the accelerator and the irradiation of a patient in Gantry 2.

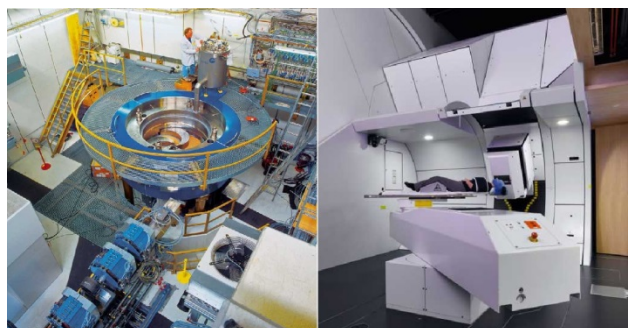


Figure 1: Cyclotron accelerator and Gantry 2, where the patients get treated with the proton beam.

For traditional spot scanning, being dose driven, the requirements for beam intensity control are not strict. However, the beam settling time and stability are crucial for a time driven delivery. In Table 1 we present the requirements for the vertical deflector voltage that were derived from the dosimetry quality required for a line scanning patient dose delivery.

Table 1 Requirements for VD Driving Voltage

Requirement	Value
Reach set point	50 µs
Remain stable (within 10V) of set point	100 µs
Voltage ripple	+-5 V
Maximum overshoot	10 V
Full range	0-5000V

A first attempt of using a traditional PID controller in the TDS to regulate the beam intensity was used as a proof of concept [7] but precision and reproducibility were still far from clinically usable. System latency, power supply hardware limitations and variability of the accelerator behaviour are the main challenges which we had to address with a novel design.

In this article we will present how we overcame the limitations of the beam current regulation system of the accelerator by characterizing the plant over more than a year, modelling it with Matlab and generating a gain scheduled controller using Simulink's HDL coder that was then implemented in the TDS on an FPGA.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC2

BEAM DELIVERY CHALLENGES

The proton beam extracted at the accelerator is directed through a beam line up to the Gantry 2, where patients are irradiated. The TDS is in charge of reading the prescribed patient irradiation fields and requesting the right intensity, among other parameters, from the accelerator.

Hardware Constraints

Due to the harsh radiation environment and limited space available in the accelerator bunker, the VD power supply is located outside the concrete shielding, one floor above. The cable leading the high voltage to the VD is about 35 meters long and has a capacity of 3.5 nF, the plates themselves 100 pF and together with the internal cabling adds up to 5 nF and 125 mΩ. This is an almost purely capacitive load in a typical working range of 0 to 2500 V.

To match the stringent requirements in Table 1, the supply is internally built using switched stages instead of as a monolithic device. The internal switching results in slight overshoots when the set point crosses an internal boundary, as seen in Fig. 2.

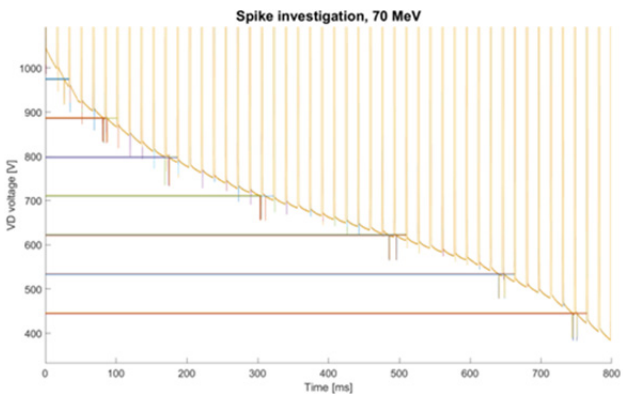


Figure 2: Sequence of consecutive lines covering a typical clinical working range, measured at the output of the VD power supply, showing voltage spikes mostly grouped at regular intervals.

A priori it is not possible to predict when the power supply will cross an internal boundary, which results in apparently random beam current overshoots. In Fig. 3 we can see how an exact same pattern applied several times results in different step responses.

Accelerator Variability

Our particle accelerator is a superconductive machine, whose magnetic field is extremely sensitive to environmental conditions. This results in delivering different beam currents on different days for the same vertical deflector voltage. In addition, the beam current extracted is related to the voltage applied at the vertical deflector in the shape of a Gaussian bell, which means that a given voltage increment results in different current changes depending on its location in the curve. Furthermore the losses of beam are different for different energies. The convolution of all these effects makes the plant highly non-linear, and control is challenging (see Fig. 4).

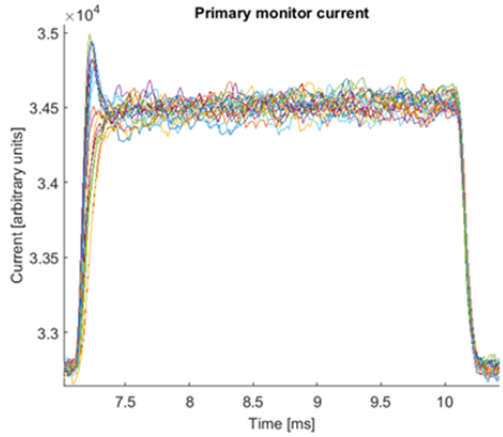


Figure 3: Superposition of 20 sequential executions of the same voltage square pattern in open loop.

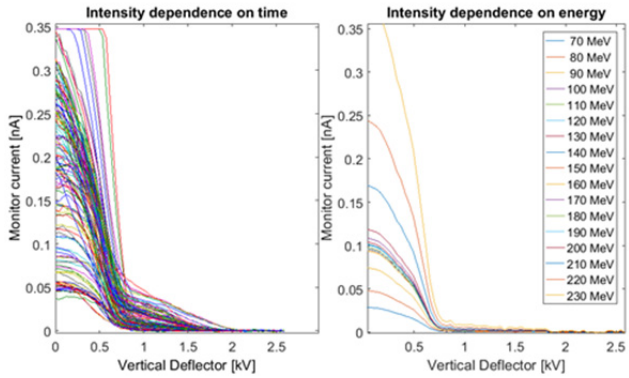


Figure 4: Beam current transfer function variation for a given vertical deflector voltage over a year for a fix energy (left) and for different energies the same day (right).

Latency

The time between the TDS software deciding a set point and the power supply output reaching it is about 80 µs. The time between the protons passing through the vertical deflector (modulation point) and the extraction point into the beamline is 20 µs. There is a negligible transport time in the beam line, but the collection time of the ionization monitor is approximately 90 µs. To this, a readout time of 10 µs of the monitor electronics is added, which computes to a total latency in the order of 200 µs between requesting a value and measuring its effect. Latency makes intensity regulation challenging, as the reaction time is longer than most of the dynamic effects observed. As reference, in clinical practice we intend to apply lines as short as 300 µs long.

FACILITY MODELLING

We acquired data along a year in order to get a good idea of the variability of the beam arriving at the Gantry 2. We sampled the high voltage at the output of the power supply, the set point from the control system and the beam current at the Gantry in a synchronous way with a 10 µs time resolution. We applied random jumps between two voltage levels within the working range of the vertical deflector and classified the data according to two

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC2

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

variables: Overshoot of the beam current in open loop, and size of the voltage jump applied.

As shown in Fig. 5, there is a high variability in the behaviour of the facility, as the unpredictability of the power supply is amplified by the changes in the beam transfer function. For a requested intensity change, depending on the energy settings and accelerator conditions of the day, a different voltage jump will be applied. This can itself take different internal switching paths and lead to very different overshoots before reaching a stable value. It is to be noted that the smaller the jump, the higher the variability is. We divided the jumps in three areas, of low, medium and high potential overshoot.

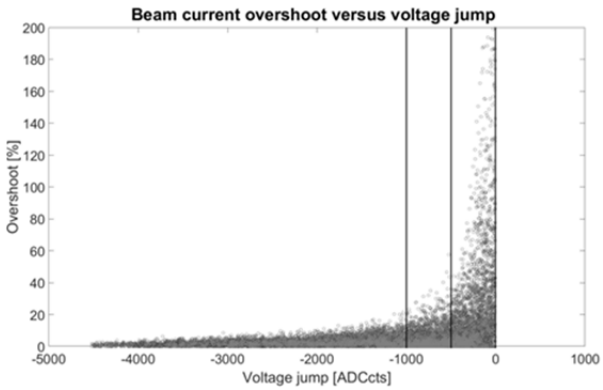


Figure 5: Random voltage jumps classified by size and beam overshoot measured [8].

The response of the accelerator, beamline and beam monitor to a given voltage jump (see measured step response in Fig. 6) can be assimilated to a second order filter and a delay.

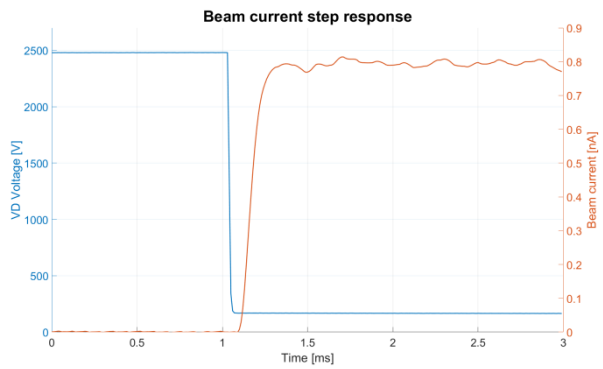


Figure 6: Accelerator control signal and dose monitor response to a voltage step.

Based on the data acquired, we created a Simulink model that would represent the facility from voltage set point to beam current detected at the ionization chamber (see Fig. 7). For each of the areas of varying potential overshoot (high, medium and low), we selected a set of evenly distributed data points (Fig. 5), and identified a set of linear models from these data points. These models were then used for the controller design. The generation of these models was automated using a Matlab GUI that takes as input a set of random voltage jumps and the resulting proton current read-out at the beam monitor. In this way, it is possible to re-generate the model whenever the conditions of the facility change, as in the case of a parts exchange.

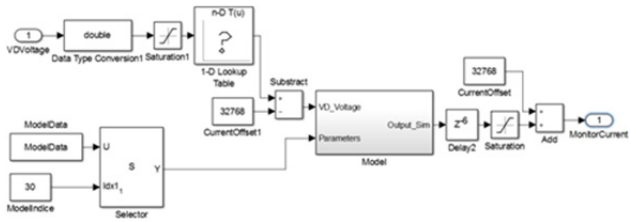


Figure 7: Main blocks of the plant modelled with Simulink.

CONTROLLER DESIGN

We took the strategy of developing a gain scheduled controller using the frequency-domain robust control (FDRC) [9] toolbox in Simulink. In this method, we define three criteria that the closed-loop system must meet: the settling time, the maximum instantaneous current overshoot and the integral dose difference. These criteria are derived from dosimetry considerations on line scanning plans, and they are the basis for calculating a desired open-loop transfer function that the optimization algorithm of the FDRC toolbox will try to approach.

For each of the three areas of varying potential overshoot discussed in the previous section, we designed a separate controller, adjusting the robustness (via the modulus margin) depending on the variability in the area. Thus, the area for the small jumps, with the largest overshoot variability, will have the most robust controller. An example of desired open-loop transfer function and robustness (modulus margin) can be seen in Fig. 8.The controllers that have been designed were of order 3. Since all of the 3 controllers (one for each area of varying potential overshoot) have the same structure, we can simply choose the parameters of the controller to be used based on the requested voltage jump.

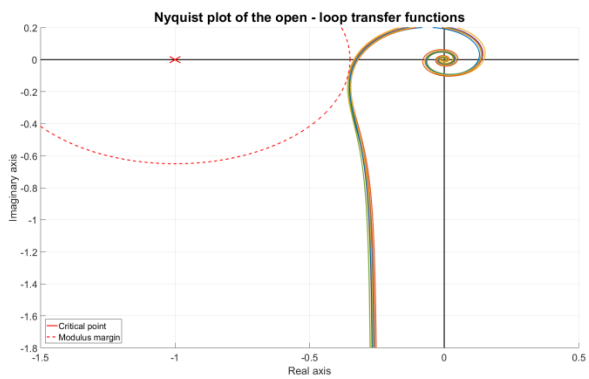


Figure 8: Definition of the stability margin of the controller on a Nyquist plot.

To cope with latency we introduced a Smith predictor in the controller, to react on the expected behaviour of the plant and to compensate for the 200 µs delay. We chose three predictor models for our controller (one for each area of varying potential overshoot). The model chosen

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC2

for each area presents an average overshoot of that area. The full diagram is depicted in Fig. 9 and the Simulink design is shown in the Appendix. We then used the HDL coder functionality to generate the 3209 lines of VHDL code that were instantiated in the control system's FPGA, replacing our older PID controller.

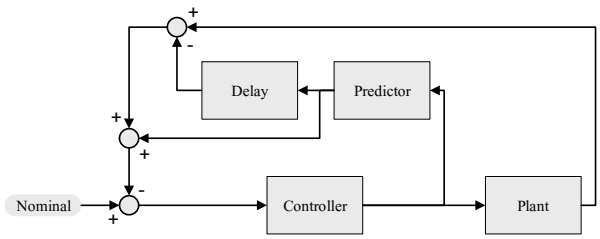


Figure 9: Structure of the new controller designed.

RESULTS

We tested the new controller, with and without latency compensation, in Gantry 2 with beam and compared it to the older PID implementation controller. In Fig. 10 the performance is compared between the new controller, with the predictor delay compensation disabled, and the older PID controller. Preliminary results in the facility show that a precise control of the beam intensity in line scanning irradiation can be achieved both in terms of instantaneous current and measured dose distribution. The line shown in Fig. 11 is a realistic example from a patient dose field, applied using the gain scheduled controller with delay compensation.

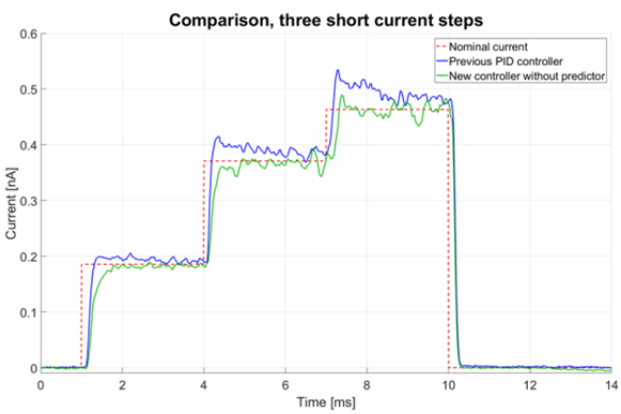


Figure 10: Comparison of PID and gain scheduled controller instantaneous beam intensity control of a line containing three consecutive steps.

.

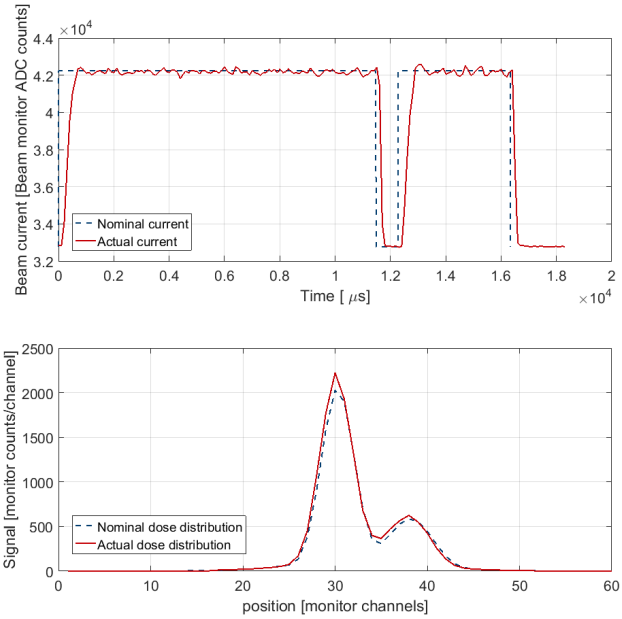


Figure 11: Experimental results with beam. Instantaneous monitor current (above) and dose profile measured at the nozzle strip chamber (below).

DISCUSSION AND CONCLUSION

We have presented the design process of a precise beam intensity controller designed for line scanning in our Gantry 2 treatment area. We modelled the facility in Simulink, developing an optimised gain scheduled controller and automatically generated VHDL code using the HDL coder tool. Designing a controller for the complex and highly non-linear facility was a challenge that required a good collaboration of physicists, controls and electricals engineers. Preliminary results show that the new controller demonstrates better performance than the standard PID in terms of settling time and overshoot.

Simulink proved to be a powerful tool to develop and optimise the controller. However, one should not underestimate the effort of adjusting the timing and synchronization elements to obtain realistic code that would match the particularities of the target FPGA. In our experience, Simulink offers great help but an understanding of the underlying hardware is still essential.

The current status of the development is testing, as reproducibility is still the biggest challenge. In the coming months we will continue adjusting the controller parameters and designing a stable solution that can be used clinically, adapting to day to day variations with minimal manual intervention.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC2

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

REFERENCES

[1] S. Lin *et al.*, "More than 10 years experience of beam monitoring with the Gantry 1 spot scanning proton therapy facility at PSI", *Medical Physics*, vol. 36, no. 11, pp. 5331-5340, 2009.

[2] D. C. Weber *et al.*, "Spot Scanning–based Proton Therapy for Intracranial Meningioma: Long-term Results from the Paul Scherrer Institute", *International Journal of Radiation Oncology*Biology*Physics,* vol. 83, no.3, pp. 865-871, 2012.

[3] A. Lomax *et al.*, "Treatment planning and verification of proton therapy using spot scanning: Initial experiences", *Medical Physics*, vol. 31, no. 11, pp. 3150-3157, 2004.

[4] A. Schätti, D. Meer, and A. J. Lomax,"First experimental results of motion mitigation by continuous line scanning of protons", *Physics in Medicine and Biology*, vol. 59 no. 19, pp.5707–3723, 2014.

[5] J. M. Schippers *et al.*, "The SC cyclotron and beam lines of PSI's new proton therapy facility PROSCAN", N*uclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 261, nos. 1–2, pp. 773–776, 2007.

[6] J. M. Schippers, D. Meer, and E. Pedroni, "Fast scanning techniques for cancer therapy with hadrons – a domain of cyclotrons", in *Proc. Cyclotrons'10*, Lanzhou, China, Sep. 2010, pp. 410–415.

[7] S. Psoroulas, P. Fernandez Carmona , G. Klimpki, C. Bula, D. Meer, and C. D. Weber, "Challenges in fast beam current control inside the cyclotron for fast beam delivery in proton therapy", in *Proc. Cyclotrons'16*, Zurich, Switzerland, Sep. 2016, pp. 126-129.

[8] V. Minnig, "A high-precision gain-scheduled controller for the beam intensity regulation in proton therapy treatments", Master thesis, Ecole Polytechnique Federal de Lausanne, Switzerland, 2018.

[9] A. Karimi, "Frequency-domain robust controller design: A toolbox for Matlab", Automatic Control Laboratory, Ecole Polytechnique Federal de Lausanne, Switzerland, 2012.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC2

APPENDIX

Figure 12 shows the simulink controlle design. The logic functionality is in grey and the colour blocks are different synchronization elements to take into account the particularities of the TDS hardware, to allow different clock domains and pipelining.

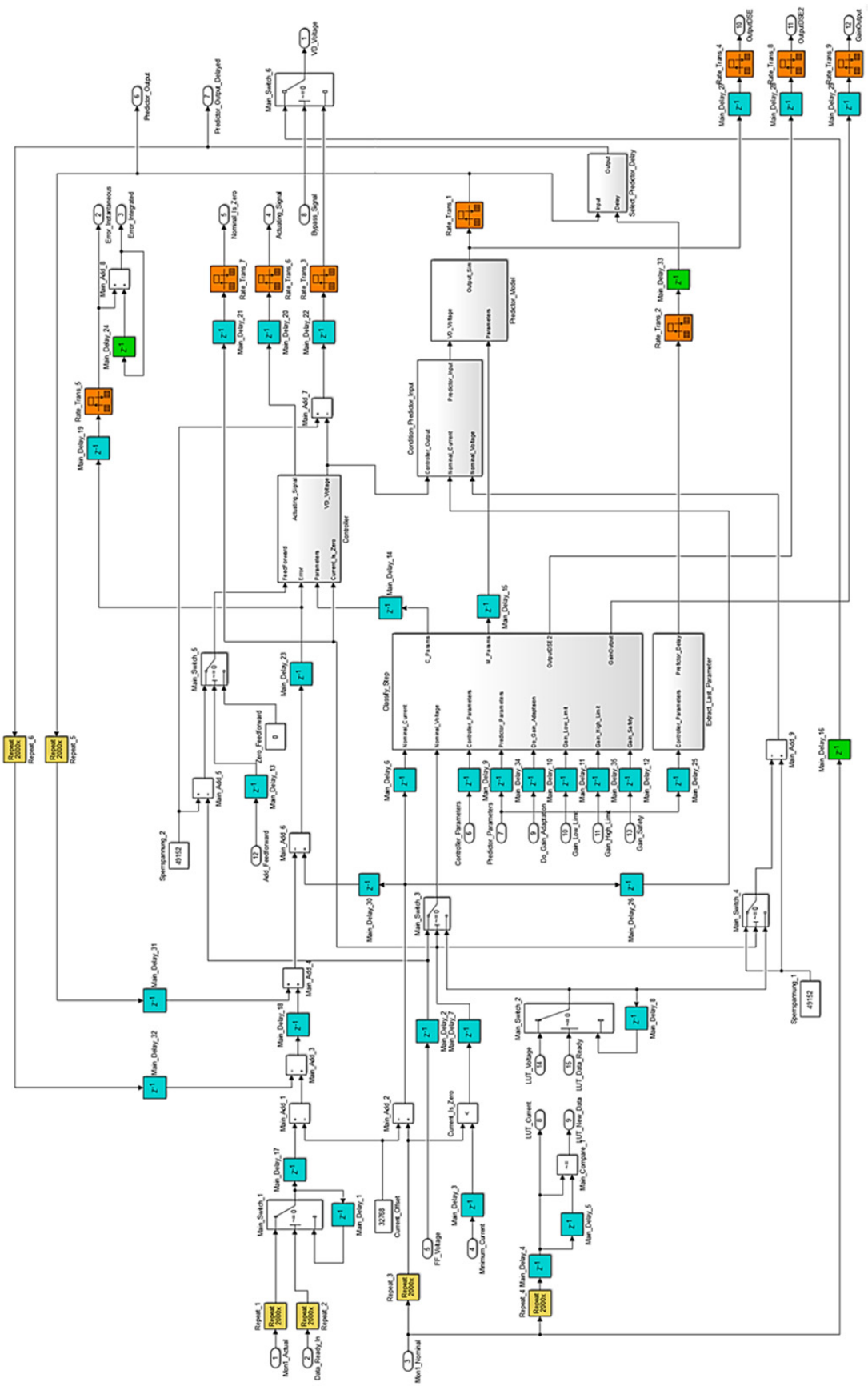


Figure 12: Simulink controller design.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC3

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

CERN SUPERVISION, CONTROL AND DATA ACQUISITION SYSTEM FOR RADIATION AND ENVIRONMENTAL PROTECTION

Adrien Ledeul, Gustavo Segura Millan,
Alexandru Savulescu, Bartlomiej Styczen, Daniel Vasques Ribeira
CERN, Geneva, Switzerland

Abstract

The CERN Health, Safety and Environment Unit is mandated to provide a Radiation and Environment Supervision, Control and Data Acquisition system for all CERN accelerators, experiments as well as the environment.

The operation and maintenance of the previous CERN radiation and environment supervisory systems showed some limitations in terms of flexibility and scalability.

In order to face the increasing demand for radiation protection and continuously assess both conventional and radiological impacts on the environment, CERN developed and deployed a new supervisory system, called REMUS - Radiation and Environment Monitoring Unified Supervision.

REMUS design and development focused on these desired features. REMUS interfaces with 75 device types, providing about 3,000 measurement channels (approximately 600,000 tags) at the time of writing.

This paper describes the architecture of the system, as well as the innovative design that was adopted in order to face the challenges of heterogeneous equipment interfacing, diversity of end users and continuous operation.

INTRODUCTION

CERN HSE Unit is in charge of providing Radiation Protection and Environmental Impact Monitoring to CERN facilities and immediate surroundings. This constant surveillance ensures:

- Workplace safety for CERN personnel and external visitors.
- Quick and appropriate response to any unplanned release of potential pollutants or ionizing radiation.
- Reporting to the authorities the nature and the quantities of emitted ionizing radiation.
- Reporting to the authorities of any release of potential pollutants.

In order to fulfil these missions, CERN has set up numerous and diverse monitoring equipment across the organization and its surroundings.

In addition, CERN HSE Unit provides a Supervision, Control And Data Acquisition (SCADA) system able to gather remotely all the relevant data in the domains of Radiation Protection and Environmental Impact Monitoring.

Between 2005 and 2016, HSE operated a supervisory system named RAMSES (RAdiation Monitoring System for the Environment and Safety) [1], which was in charge of the data acquisition, control/command and supervision of 50 device types, allowing the remote supervision of 1,500 channels.

This system allowed, for the first time at CERN, to supervise the majority of HSE relevant equipment in a consistent manner. However, it had some limitations in terms of scalability and evolutivity, making it unsuitable for longer-term use. For instance, some monitoring equipment were not remotely supervised, due to the high cost of interfacing them through the supervision. Some others where remotely controlled by proprietary supervisors, such as Berthold MEVIS, which constrained final users to use different software to operate them.

In 2012, HSE started the development of a new supervisory system, REMUS (Radiation and Environment Monitoring Unified Supervision) that would tackle those limitations. In particular, this system would:

- Interface with all the equipment that were interfaced with RAMSES system.
- Allow quick interfacing of new type of monitoring equipment, in order to unify the supervision and integrate stand-alone devices, most of them being COTS (Commercial Off-The-Shelf) products.
- Provide a reliable, scalable and cost-effective system.
- Use common CERN software, WinCC OA (WinCC Open Architecture) [2], and JCOP framework (Joint COntrols Project) [3].
- Reduce the delay and the cost of declaring and displaying new devices of known types in the supervision.
- Provide light and fast clients, tailored to the needs of diverse end-users.
- Reduce the overall maintenance and user support effort necessary to maintain the system in operation.

REMUS founded its evaluation of scalability and performance on the study performed by CERN engineering department [4]. The technologies used are commonly employed in large SCADA systems supervising the LHC (Large Hadron Collider) such as the LHC Quench Protection System Supervision or CERN Electrical Network Supervision.

For its development, REMUS applies the same International Standard IEC 61508 [5] that was set up during the development of RAMSES.

CORE FUNCTIONALITIES

Project Objectives

The REMUS project aims at developing a universal software for supervision, control and data acquisition for

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC3

the entire suite of monitoring stations covering all radiological and environmental parameters that can potentially be affected by the operation of the facilities of CERN.

Functionalities

The main functions of the system (See Fig. 1) are:

- Data acquisition of measurements and events coming from the instrumentation.
- Display of near real-time measurements, alarms and operational states of instrumentation through customizable user interfaces composed of synoptic, widgets and alarm screens.
- Remote sending of commands and operational parameters to the instrumentation.
- Display of archived and near real-time measurements and events coming from the instrumentation, through a data visualisation tool, ERGO (Environment and Radiation Graphic Observer).
- Publishing of alarms and faults to the CERN central alarm system, LASER (Lhc Alarm SERvice) [6].
- Publishing of measurements to other systems via the CERN Data Exchange system, DIP (Data Interchange Protocol), built on the top of DIM (Distributed Information Management) [7].
- Publishing of measurements to CERN shared MDB (Measurement DataBase) [8].

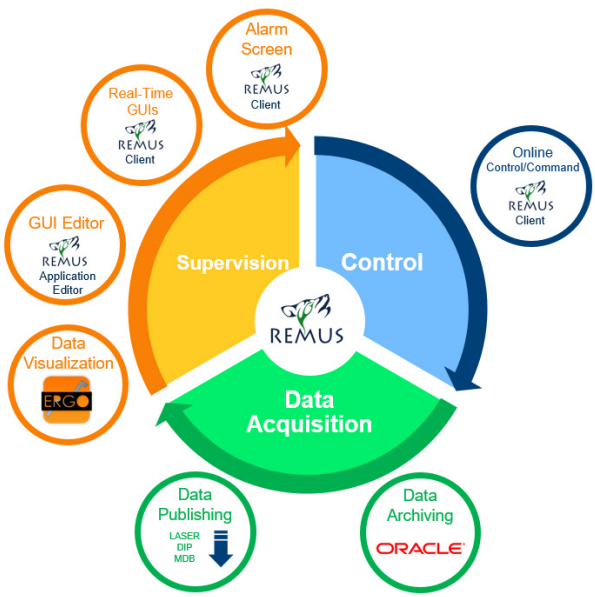


Figure 1: REMUS functional diagram.

The system shall monitor the instrumentation 365 days a year, 24/7 and ensure the data acquisition of measurements and system events in case of network outages.

HETEROGENEOUS DEVICE TYPES INTEGRATION

Problem Statement

CERN has set up, over the years, numerous and diverse monitoring equipment across the organization and its surroundings, in order to efficiently monitor ionizing radiation and environmental impact.

The first driver of this heterogeneity is the wide scope of measurements to acquire, ranging from weather conditions, useful to predict the dispersion of pollutants in the environment, to gamma ionizing radiation.

Another driver is the long lifetime and consequent cost of such monitoring equipment. As a result, monitors are not necessarily replaced all at once when a newer generation of the same type of monitor is available on the market and selected by CERN for installation. They are gradually replaced over the year, duplicating the number of different monitors to interface with the supervisory system. This heterogeneity creates challenges on the SCADA side.

The first challenge is the obsolescence of the firmware and protocol used by such devices, which generally happens before the obsolescence of the electronics. The use of obsolete technologies and protocols make the integration with modern network infrastructure and SCADA troublesome.

Another challenge is the increased complexity of the system as a whole. The SCADA shall make abstraction of heterogeneity for the end user, whose attention should be targeted on the data produced by the instrumentation rather that its technical specificities.

Abstraction by Modelling

REMUS took the approach of making abstraction of the instrumentation by modelling it with simple and generic concepts:

- Channel: An abstract entity representing a geographic point of measure, or a functional position (e.g. “Valve pressure of equipment X”). It allows REMUS to identify every data streams. It is associated to a device.
- Device: A piece of equipment of a determined type (Electronic boards, PLCs, Ionization chambers …) holding its own parameters, possibly connected to other devices and channels.
- Monitoring Station (MS): An encapsulation of a set of devices and associated channels, connected to the system through the network infrastructure.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC3

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

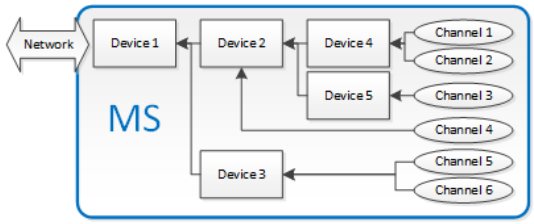


Figure 2: REMUS Instrumentation Model.

Each type of Monitoring Station to interface is modelled in a relational database, using those concepts (See Fig. 2). Usage of abstract concepts homogenizes the software, makes most of source code reusable and simplifies the development process, from analysis to tests and validation.

In addition, it provides REMUS users with what appears to be a uniform instrumentation, easier to comprehend and operate.

DIVERSITY OF USERS

Problem Statement

Radiation Protection and Environmental Monitoring data are of interest for many different type of users at CERN. REMUS shall provide interfaces for operators of the different CERN accelerators and experiments, radiation protection engineers, environmental engineers, physicists, maintenance teams and the CERN fire brigade. The user interfaces shall be easily editable, and as close as possible to the users' needs in term of data and its presentation.

Editable Graphical User Interfaces

REMUS allow trained users to design their own Graphical User Interfaces (GUI), encapsulating all the widgets and synoptic they may need for them and their teams. The GUIs are called *REMUS Applications*, and consist solely of a descriptive xml file, that can be uploaded to the REMUS Server (See Fig. 3). Applications can be created and updated through a user friendly interface, the *Application Editor*. The applications have specific access control, and a dedicated alarm screen. All the applications can be updated at run-time, and the updates are propagated to all the clients simultaneously.

Today, REMUS has 18 distinct Applications, such as Large Hadron Collider Radiation overview, used by CERN Control Center (CCC), or Water conventional monitoring, used by the CERN Fire Brigade.

The advantages of this strategy are multiple:

- Users can focus on devices they are interested in (only a sub-set of the total instrumentation).
- The performance of the clients is improved, as only the variables displayed in the current Application are fetched from the servers.
- The maintenance effort is distributed among several *Application Administrators*, close to the needs of the users.

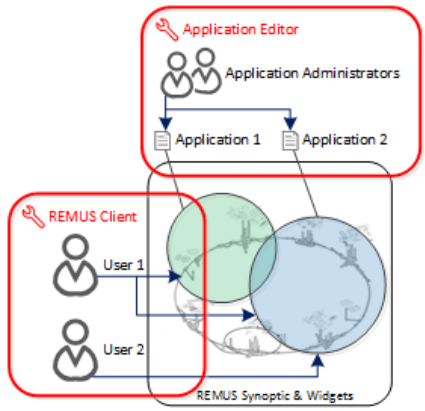


Figure 3: REMUS Application definition and usage.

ARCHITECTURE

Interfacing with Instrumentation

REMUS currently implements five different ways to interface the instrumentation, depending on the protocols implemented by the device firmware:

- WinCC OA native driver (Modbus, S7)
- WinCC OA CERN driver (Controls MiddleWare)
- In-house driver developed on top of WinCC OA API
- File exchange system
- OPC

Data Acquisition & Archiving

The instrumentation connected to REMUS provide about 1,000 measurements to archive every second.

All the devices produce homogeneously formatted measurement files, at the lowest possible level (Firmware, Driver or Supervision), that are later injected into REMUS Oracle database through Oracle SQL*Loader.

This mechanism allows capable devices and intermediate file servers to buffer the measurement data in case of network outages in order to ensure the availability and recovery of the data.

Real time Animation Graphical User Interfaces

The drivers connected to instrumentation send about 1,800 data point updates requests to the central servers every second. The redundant central servers then proceed with the update, and replicate their respective internal databases. The cross-platform Client GUI (remote UI in WinCC OA terminology), running on Linux or Windows, requests all the data needed to animate the tailor-made GUIs either by systematic polling at defined intervals or driven by notifications of updates.

Animation of the alarm screens are working in a similar fashion, with the exception that the alarms meta information is pre-computed on the server side in order to minimize the data exchange between the clients and the redundant servers.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC3

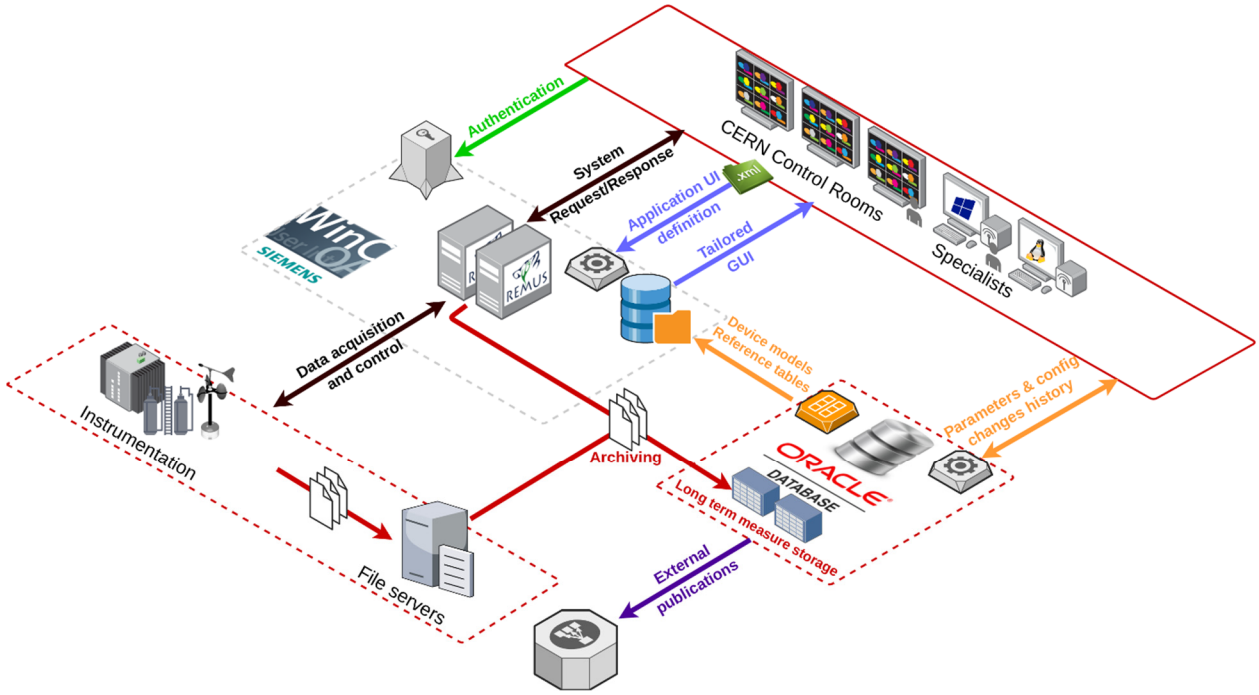


Figure 4: REMUS architecture.

Control/Command

Dedicated UI panels are available from the client GUIs in order to send commands to the instrumentation. Commands and change of parameters requests are encapsulated into formatted messages, *System Requests*, to the central servers, which forward them to the selected device. Once the command has been acknowledged by the device, the change of parameters is recorded in the REMUS Oracle database for future reference and audit trailing.

CONTINUOUS OPERATION

Radiation and Environmental impact monitoring are equally important during operation or shutdown of the accelerators. While the run phases of the accelerators shall naturally be closely monitored, the shutdown periods are typically a time when underground access is more frequent, and when the personnel have more exposure risks. As a result, REMUS needs to be available 24/7, 365 days a year.

REMUS architecture (See Fig. 4) has been designed to be as resilient as possible, in order to ensure its continuous operation.

In addition, all maintenance and support operations have been designed to be executable at run-time, making a 0% downtime objective nearly achievable. Between 2013 and 2018, the availability of the system surpassed 99.998% (20mn of planned down time due to servers Operating System change and 15mn unplanned downtime due to power or network cuts).

This high availability figure has been made possible by implementing the following features.

Run-time Declaration of New Devices

REMUS offers the possibility to users having the necessary access rights to declare new devices to interface directly from the client GUI.

Users select a known device type from the list of available models, a driver and an address. The supervision will add all the internal variables and necessary connections.

Combined with the possibilities of tailoring the GUIs, a new device can be added and have its data shared to all the concerned users in minutes. All this without any interruption of the supervision.

Server Redundancy

REMUS core consists of two redundant synchronized servers, playing interchangeably the roles of *Active* and *Hot Standby* (See Fig. 5). This redundancy has two major advantages:

- The *Hot Standby* server can take over the operation of the core in case of network or power outage of the *Active* server.
- All maintenance operations can be carried on the *Hot Standby* server, transparently for the users.

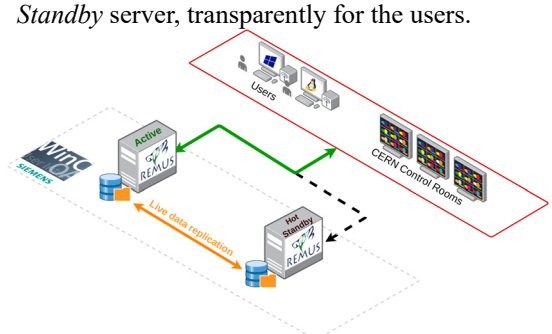


Figure 5: REMUS redundancy.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC3

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

CONSIDERED EVOLUTIONS

REMUS succeeded in all its initial objectives, and is now the unique supervision of more than 3,000 measurement channels, monitoring 75 types of monitoring equipment in charge of the surveillance of Radioactivity levels and environmental impact.

Since REMUS is in production, in 2013, the amount of single measurements archived by HSE supervisory systems jumped from 15 billion to 30 billion a year, and the number of monitored channels from 1,700 to 3,200. This dramatic increase generated new needs from Environmental and Radiation protection specialists. New tools and features are required in order to allow efficient data analysis.

High Resolution Measurement Storage

In order to guarantee low latency when accessing data for a long period, archived REMUS data are reduced after a defined time, following algorithms defined for each type of measurements. In most common cases, the measurements are reduced to one minute resolution after 15 days, and with the best available resolution around specific events (i.e. exceeded thresholds).

However, radiation protection specialists are interested, in some cases, to be able to keep the best available resolution for a longer period of time. NoSQL (Not Only SQL) ecosystems could offer such high resolution measurement service, at the cost of a higher latency on data retrieval.

Both services could be made available from the data visualisation tool, in order to be able to switch from a low latency/low resolution service to a high latency/high resolution one (See Fig. 6).

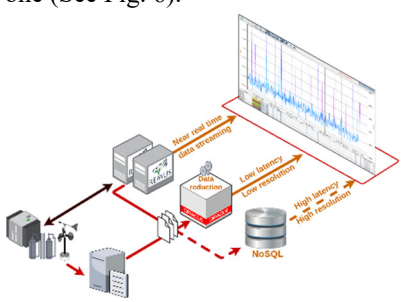


Figure 6: REMUS future data service architecture.

Data Streaming

REMUS users are gaining interest on accessing REMUS real time data on mobile terminals, for on-site operations.

The emergence of big data technologies has brought new possibilities regarding reliable and scalable data streaming services. At the time of writing, REMUS relies on a CERN-developed protocol for streaming data in near real time to its data visualisation tool.

A modernisation of REMUS data streaming service could be an opportunity to both increase its efficiency and allow the development of web and mobile applications.

CONCLUSION

The CERN Health, Safety and Environment Unit succeeded in the development and deployment of a new Radiation and Environment Supervision, Control and Data Acquisition system.

This supervisory system, called REMUS - Radiation and Environment Monitoring Unified Supervision - is now the unique supervision for all radiation and environmental monitoring equipment at CERN.

Its cost-effectiveness, reliability and scalability makes it suitable for long term use at CERN, and opened the possibility of collaboration with other laboratories and industrial partners.

ACKNOWLEDGMENT

We would like to thank all the members, past and present, of the REMUS team as well as colleagues from Radiation Protection, Environment, Engineering, Information Technology and Control departments for their fundamental contribution to the success of the REMUS project.

REFERENCES

[1] G. Segura Millan, D. Perrin, L. Scibile, “RAMSES: The LHC Radiation Monitoring System for the Environment and Safety”, in *Proc. ICALEPCS’05*, Geneva, Switzerland, Oct. 2005, paper TH3B.1-3O.

[2] ETM.at website: http:// http://www.etm.at

[3] O. Holme, M. Gonzalez Berges, P. Golonka, S. Schmeling, “The JCOP Framework”, in *Proc. ICALEPCS’05*, Geneva, Switzerland, Oct. 2005, paper WE2.1-6O.

[4] P.C.Burkimsher, “Scaling up PVSS”, in *Proc. ICALEPCS’05*, Geneva, Switzerland, Oct. 2005, paper PO1.056-6.

[5] International Standard IEC 61508 “Functional safety of electrical / electronic / programmable electronic safety-related systems” (E/E/PE) system.

[6] F. Calderini, B. Pawlowski, N. Stapley, M.W. Tyrrell, “Moving towards a common alarm service for the LHC era”, in Proc. ICALEPCS’03, Gyeongju, Korea, Oct. 2003, paper TH512.

[7] C. Gaspar, M. Dönszelmann, Ph. Charpentier,“DIM, a portable, light weight package for information publishing, data transfer and inter-process communication” *Computer Physics Communications*, vol. 140, pp. 102– 109, 2001.

[8] C. Roderick, R. Billen, M. Gourber-Pace, N. Hoibian, M. Peryt, “The CERN Accelerator Measurement Database: On the Road to Federation”, in *Proc. ICALEPCS’11*, Grenoble, France, Oct. 2011, paper MOPKN009.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC4

MAINTENANCE AND OPTIMIZATION OF INSERTION DEVICES AT NSLS-II USING MOTION CONTROLS

Christopher Guerrero, Richard Farnsworth, Dean Hidas, Yuke Tian, John Escallier
Brookhaven National Laboratory, 11793 Upton, NY USA

Abstract

The purpose of this project is to demonstrate the effective improvements on insertion device performance via upgrades on the motion control software. The insertion devices installed inside the National Synchrotron Light Source II (NSLS-II) storage ring are currently operating at sub/micron resolution with slow speeds, which can limit the scope of user experimentation preferences. We can manipulate the devices with adaptive tuning algorithms to compensate for varying electromagnetic forces throughout motion scans. By correcting positional feedback with encoder compensation and redefining motion programs, we can safely increase the speed to run the same motion trajectories in less time.

INSERTION DEVICES

Introduction

An insertion device (ID) is used at NSLS-II as a photon source for beamlines. These devices include in-vacuum undulators (IVU), elliptically-polarizing undulators (EPU), damping wigglers (DW), and three-pole wigglers (3PW). With the exception of the 3PWwhich has a fixed gap between the three magnetic poles, all insertion devices adjusts the distance between parallel arrays of magnets. This distance is called the ID gap, and EPUs have additional degrees of freedom that can be adjusted horizontally, which controls the polarization of photons produced. This type of adjustment is called the ID phase.

Motion Control System

The motion control system used for the insertion devices comprises of both hardware and software. The hardware used for IDs at NSLS-II are: 64-bit Linux computers; Delta Tau Programmable Multi Axis Controller (PMAC-2) modules [1]; Delta Tau/in-house built power supply modules; Stepper/servo motors (and brakes where needed); Renishaw optical glass linear positional encoders [2]; embedded rotary positional encoders (for some IDs); Ethernet, motor, and encoder cables. The ID software is mainly divided into three parts: configuration on the PMAC, software Input Ouput Controller (IOC), and user interface via Experimental Physics and Industrial Controls System (EPICS) communication [3]. IOC has EPICS applications to channel access interface with motor controllers. The user interfaces for IDs are operator interface (OPI) screens on Control System Studio (CSS) software used facility-wide [4]. These OPIs can set and monitor kinematic parameters of the motion control system. In summary, the major control system components (seen on Fig. 1) are: the drive assembly connected by motors, the motor controller and driver, the IOC, and the OPI.

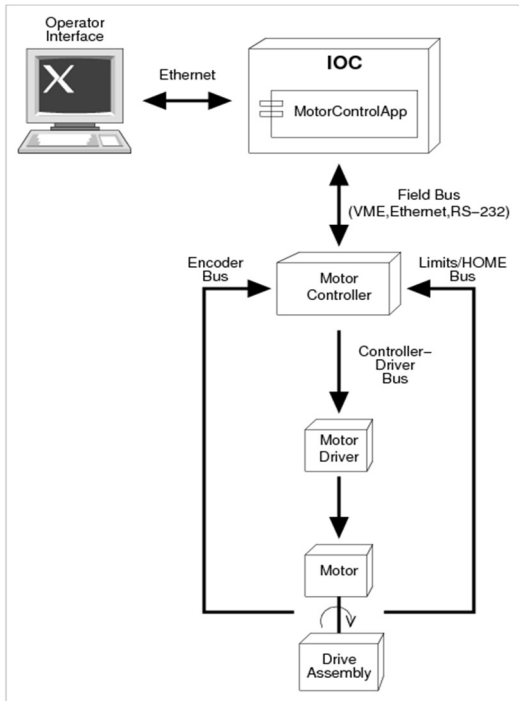


Figure 1: Motion control system components.

3PW/DW Motion Control Setup

The simplest motion control system for an ID at NSLS-II is that of a 3PW since it only involves one motor. A 3-phase stepper motor runs at 2 Amps to rotate the drive assembly. This assembly contains a motor, a gearbox, and a ballscrew, which moves the actual magnetic device horizontally on double supported rails. The direction of this motion in reference to the accelerator beam is inboard (positive direction) and outboard (negative direction). We set the fully outboard position to zero, and the 3PW reaches the fully inboard position when it is aligned with the beam passing through a flattened beampipe chamber section.

For the 3PW, we use a 50-nanometer resolution Renishaw absolute linear encoder for positional feedback. This is directly read on the PMAC on power on, so we always recover the last known position. The software does not require coordinated definitions on the PMAC configuration, so we directly change the position to the axis using jog commands. These move commands are transferred via setpoint and go fields in the OPI seen on Fig. 2 below. We can also enter the desired position using the move in/out command buttons to fully insert/retract the device.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

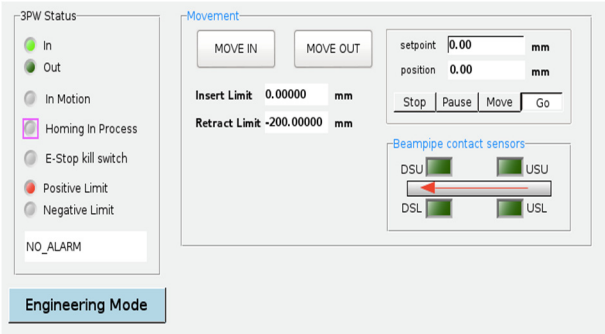


Figure 2: 3PW OPI.

The 3PW OPI can also show status of limit switches, inboard/outboard position, proximity switches to the beampipe chamber, soft limits, and position readbacks.

We are currently using 7 3PWs at cells 4, 6, 7, 11, 12, 17, and 22 of the NSLS-II storage ring. This means we create a motion control sub-system for each 3PW, create an IOC for each 3PW, and run all softiocs within the same IOC server.

IVU/EPU Motion Control Setup

The motion control systems for undulators are more complex and require solid knowledge of coordinated motion to correctly adjust the gap size. For our IVUs of magnetic assemblies that are 1.5 meters or 2.8 meters long, the control system is designed to drive four shafts using ballscrew supported by fixed bearings. The 3 meters long IVU and EPUs have a control system to drive a one piece shaft supported by fixed-free bearings. The shaft assemblies are interconnected with motor assemblies through worm reducers, as seen on Fig. 3 below.

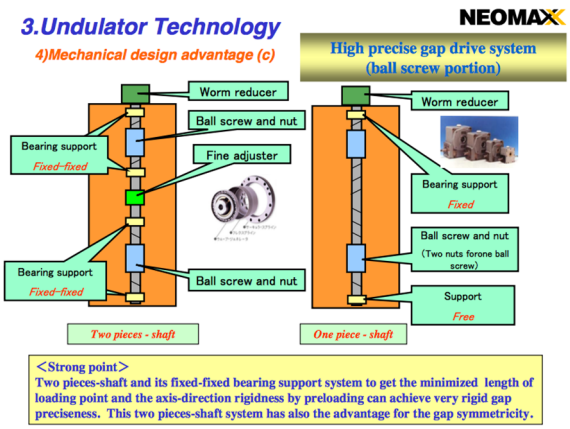


Figure 3: Neomax Undulator Mechanical Design.

The IVU devices are configured to simultaneously drive one motor on the upstream position and another motor on the downstream position in reference to the beam direction. The other IVUs have controls that allow us to move two (upper and lower) motors on each side. By moving the motors together, we can incrementally open/close the gap. These undulators require two encoders for each motor: a linear absolute feedback mounted externally of the shaft alignment, and a rotary incremental feedback directly coupled to the motors. In total, the undulators' controls systems provide four linear feedbacks for gap position, and four rotary feedback for motion commands and status. The OPI for IVUs' motion controls seen on Fig. 4 allow operators to set the gap by entering the value and starting a motion program that executes the synchronous move and updates the actual to the new setpoint. There is a fine positional adjustment program that can be enabled for IVU control systems, this is labelled as the correction function on right of the OPI.

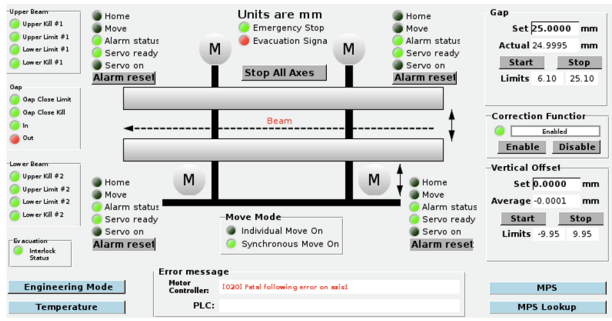


Figure 4: IVU OPI.

INSERTION DEVICE ISSUES

Issues with 3PW

The main issue with controlling motion of the 3PWs was due to small PID gain, motor driving current, and low operational speed values. These parameters are configured internally in the Delta Tau software. Most 3PW devices were running at 0.2 millimeters per second, which would take about 15 to 20 minutes to complete full range of motion. From fully inserted to fully retracted and vice-versa, this range of motion is approximately 190 millimeters.

After re-evaluation of the mechanical assembly of the motor, gearbox, and ballscrew, we found there was slack between the nut and setscrew that connected the coupler with the ballscrew. The loose assembly caused poor control conditions and possible mechanical hazard if the motor shaft disconnects.

Issues with Undulators

The main issues from undulator controls are caused by:

- Linear encoder feedback not matching rotary encoder feedback
- Mechanical gearbox and springs on IVUs
- Virtual IOC server too slow for processes and commands containing all ID softiocs
- Operational error on CSS

The rotary encoders are mounted directly on the motor axes, whereas the linear encoders are mounted externally to the outer ends of the girder. This can cause geometrical issue on the cantilever motion for systems with two motors. For example, if the motors open the gap and the downstream encoders report a gap distance greater than the distance between upstream encoders, the positions report a tilt effect. To compensate for this tilt, the Delta Tau gains adjust the position of the motors to eliminate the imbalance. However, the cross coupled gantry configured on Delta Tau's does not take into account that the

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC4

linear encoders are not inline with the gap positions from the leadscrew axes that we wanted to monitor and drive.

The mechanical setbacks encountered with undulators are: bent shafts or angular misalignment in the drive train, and the meshing of spur and bevel gears internal to gearboxes.

RESULTS

3PW Motion Controls Update and Mechanical Corrections

3PW motor performance is evaluated using the Delta Tau PewinPro PID tuning software. This software allows a motion trajectory to be exercised for a certain distance and period with input PID gain values to optimize motor performance. This test was exercised for all 3PW devices before any changes to the mechanical or control setup. The plot below (Fig. 5) is an example of poor performance of the 3PW motor due to issues.

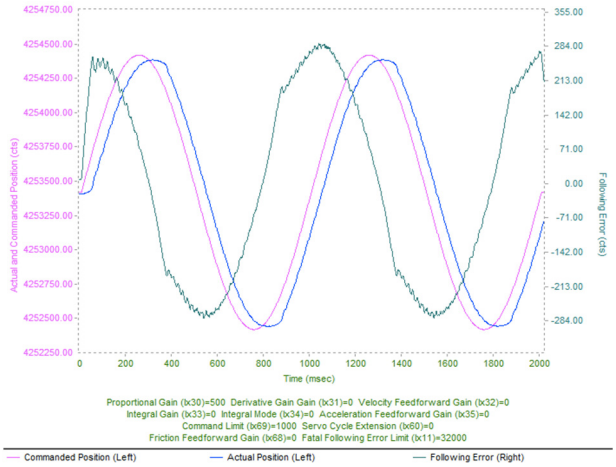


Figure 5: 3PW Sinusoidal Motion Plot with Loose Ballscrew.

The sinusoidal motion trajectory can easily display how far off the (blue) actual position trajectory was from the (pink) commanded position trajectory. The 3PW controls were updated to drive the motors with approximately 2 Amps, proportional gain was doubled, and the new operational speed set to 2 millimeters/second. This was applied to every 3PW controller after the setscrews and nuts were securely fastened to the ballscrews. With the updated configuration, we can move the devices ten times faster and complete full range of travel in less than 2 minutes.

Undulator Motion Controls Update and Mechanical Corrections

Repair of IVU controls and hardware had the highest priority due to scheduled maintenance and shutdown periods of NSLS-II. We ran diagnostic software to evaluate motor performance for all IVUs and pinpoint issues on each device. The diagnostic software plots contain DAC outputs from the Delta Tau controller. Using the plots from DAC output data, we can observe torque ripple 3illimetr. The ripple is an oscillatory modulation of the torque required for motion with constant speed.

Diagnostics of IVU Motor Performance

To repeat the motion plot for multiple IDs, we need software that can automatically generate move commands on the EPICS level and run data gather on the Delta Tau software buffer. The software to interactively tune and record motion trajectories has been internal to the Delta Tau software. This can only be done one motor at a time, for a predefined duration (distance in counts and gather period). To run the data gather more efficiently, we collect motor position data while repeating move commands on higher level of software. Using the Asyn interface via EPICS, data gather and collection can be defined simultaneously and remotely. A python shell script is executed to generate scripts for each gap move commands. For IVUs, the script changes gap setpoints for a full range motion scan. These scans can be arranged for 3illimetre increments or longer. Each scan executes a Python script that runs a buffer, saves linear and rotary encoder readbacks (as well as DAC outputs of each motor) to registers in the controller memory, reformats and processes the data, and saves this in a text file. The file is then plotted using Matplotlib (a Python package) to generate comparisons between linear and rotary encoder feedbacks. As proof of concept, Fig. 6 below demonstrates partial gap motion of IVUs from four linear encoder feedbacks and repeatability of this software.

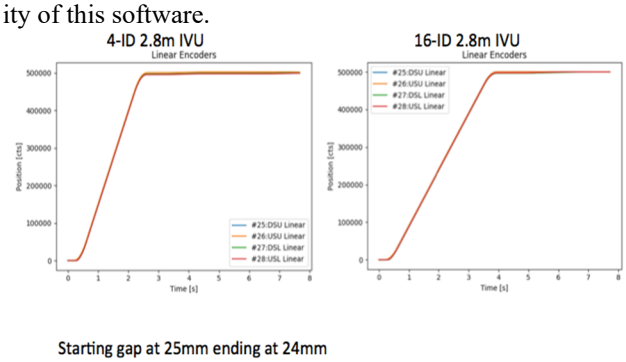


Figure 6: Gap Motion Plots of IVUs.

It would take a few hours to collect and plot this data with Delta Tau's PewinPro software. Using Python scripts, this data is recorded and plots are generated within minutes. Therefore, this software allows for a more rapid plotting and analysis of motor performance than that of the PewinPro PID tuning software. The software is maintained at: https://github.com/dhidas/IVUTests.

CONCLUSION

The corrections to motion control software have:

- Allowed 3PWs to move faster to satisfy operational requirements
- Diagnosed mechanical issues on IVUs by exercising motion and recording encoder feedback and torque command values
- Eliminated hang-ups on communication between IOC server and ID softiocs
- Identified future solutions to the non-linear forces acting on IDs

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4 doi:10.18429/JACoW-PCaPAC2018-FRCC4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

The vendors of the devices provided the last updated version of IVU motion controls in 2008-2010. This latest version of software is out-dated due to the NSLS-II controls programming standards [5]. To keep motion control software running, we will have to re-write most of the configurations, test changes, and verify operational readiness of code.

FUTURE WORK

EPU Phasing Correction

When the phase of an EPU is moving past full repulsive forces, the directions of the forces flip. If the tuning of the axis is "soft", there will be significant axis movement. Integral windup has to compensate the force direction change. The Delta Tau controllers do not understand the non-linear nature of the external forces. Therefore, the PID tuning has to be "softened" to keep stability throughout the range of motion. We are correcting the phase errors by using the rotary encoders to drive and rigidly control motor positions with compensation tables that add to the desired positions and reduce the actual position errors. Another possible solution to the phasing issue is to write an "adaptive" tuning algorithm that can adjust the PID gain values as a function of gap position and direction of motion.

Cantilever Geometry Correction

In addition, correction of the cantilever geometry from encoder feedback will:

- Give true ballscrew locations
- Remove positional error from cross couple gantry algorithm
- Allow higher proportional gain to reduce following error
- Eliminate need to install encoders directly on ballscrews

Motion Control Software Version Control

In continuation with this project, the motion controls of insertion devices will undergo configuration revisions. Version control is documented on Gitlab [6].

ACKNOWLEDGMENTS

I would like to acknowledge the contributions of all members in the Insertion Devices Task Force at NSLS-II. This project was funded by Brookhaven Science Associates.

REFERENCES

[1] Delta Tau Turbo PMAC, http://www.deltatau.com/DT_Products/ProductDetailDescription.aspx?CatID=100-BRICK%20CONTROLLER

[2] Renishaw RESOLUTE, http://www.renishaw.com/en/resolute-encoder-series--37823

[3] EPICS, https://epics.anl.gov

[4] Control System Studio, http://controlsystemstudio.org

[5] NSLS-II PMAC Programming and Configuration Standards, https://ps.bnl.gov/acc/controls/Shared%20Documents/MotionSystems/Standards/PS-R-XFD-STD-BL-MC-002_Rev1.docx

[6] NSLS-II ID Gitlab, https://gitlab.nsls2.bnl.gov/ID

ISBN: 978-3-95450-200-4

List of Authors

Bold papercodes indicate primary authors; ~~crossed out~~ papercodes indicate 'no submission'

— A —

Andre, J.M. WEP17
Anicic, D. WEP09, **THCA1**
Arroyo Garcia, J.A.R. ~~WEPI4~~, WEP04
Arsov, V.R. WEP09
Avila-Abellan, J.A. THP22

— B —

Bacher, R. **THCB3**
Balzer, B.M. ~~FRCB3~~
Barbour, A. M. ~~WEPI3~~, WEP03
Bartkiewicz, P.K. **THP23**
Bastl, P. THCA5, **FRCB1**
Behrens, U. WEP17
Bengulescu, M. ~~WEPI4~~, WEP04
Beteva, A. ~~WEPI2~~, WEP02
Biedron, S. **THCB5**
Bischof, G. ~~**WEPI3**~~, **WEP03**
Bisegni, C. WEC5, ~~WEPI8~~, WEP08
Blokhina, K.A. WEP22, FRCC1
Blomley, E. WEP10
Bobnar, J. WEP19, THCB1
Boeckmann, T. WEC1
Boonpornprasert, P. WEP24
Branson, J. WEP17
Broseta, M. THP22
Brummer, P. WEP17
Brunet, M. ~~FRCB3~~
Bründermann, E. WEP10, FRCA1
Bueno, M. ~~THPI3~~, THP03
Bula, C. FRCC2

— C —

Cao, H. WEP31, **WEP32**, WEP33
Cara, P. ~~WEP20~~
Caselle, M. FRCA1, ~~FRCB3~~
Catani, L. WEC5
Chan, C.K. WEP14
Chang, C.-C. WEP14, THP13, FRCA2
Chang, F.Y. THP18
Chang, L.-H. THP18
Chang, M.H. THP18
Chang, S.W. THP18
Chang, Y.-T. WEC2
Chen, G.H. WEC3
Chen, J. WEC2, **WEP21**, ~~THPI6~~, THP06, **THP20**
Chen, J.F. WEC3
Chen, L.J. THP18
Chen, T. ~~WEKT1~~, ~~**FRKT1**~~
Chen, W.-C. ~~**WEKT1**~~, ~~FRKT1~~
Chen, X. ~~WEPI6~~, WEP06, FRCB2
Chen, Y. WEP24
Chen, Y.X. WEP31, WEP32, **WEP33**
Cheng, C.M. THP13
Cheng, Y.-S. **WEC2**, WEP11, WEP21, THCA3, ~~THPI6~~, THP06, **THP16**, THP20
Cheng, Y.T. FRCA2
Chevtsov, P. **WEP09**
Chiu, D. ~~**THKT1**~~
Chiu, P.C. WEC2, THCA3, ~~**THPI7**~~, **THP07**
Choi, Y. ~~THPI2~~, THP02
Chu, C.P. **WEC4**
Chuang, J.-Y. WEP14, **THP13**, FRCA2
Chung, F.-T. THP18
Cittolin, S. WEP17
Ciuffetti, P. WEC5, ~~WEPI8~~, WEP08
Clausen, M.R. **WEC1**
Cobb, T.M. THCB4
Coutinho, T.M. ~~WEPI2~~, WEP02
Cui, M.H. WEP34
Cuní, G. WEP25, THP22

— D —

D'Uffizi, A. WEC5, ~~WEPI8~~, WEP08
Da Silva Gomes, D. WEP17
Dach, M. WEP09
Darlea, G.L. WEP17
de la Morena, C. ~~WEP20~~
De Santis, A. WEC5
Deldicque, C. WEP17
Demiragli, Z. WEP17
Dhammatong, Ch. ~~THPI8~~, THP08
Di Calafiori, D.R.S. WEP16
Di Pirro, G. WEC5, ~~WEPI8~~, WEP08
Ding, J.G. WEC3
Dissertori, G. WEP16
Djambazov, L. WEP16
Dobson, M. WEP17
Dominguez, M.C. ~~WEPI2~~, WEP02
Doualot, N. WEP17
Duval, P. **WEP19**, THCB1, THCB2

— E —

Edelen, A.L. THCB5
Eichin, M. FRCC2
Elcim, O.F. **WEP23**
Erhan, S. WEP17
Escallier, J. FRCC4

— F —

Fagotti, E. ~~WEP20~~

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Farnham, B. ~~WEPI4~~, WEP04
Farnsworth, R.I. FRCC4
Fedotov, M.G. WEP22
Fehlinger, C.F. WEP10
Fernandez Carmona, P. **FRCC2**
Fernández, F.F.B. WEP25
Fiedler, S. ~~THPI3~~, THP03
Fujimaki, M. WEP15
Fujiwara, T. ~~WEP18~~
Fukunishi, N. WEP15, WEP30
Fulcher, J.R. WEP17
Funkner, S. FRCA1
Furukawa, K. THP19
Furusaka, M. ~~WEP18~~
Fuwa, Y. WEP13

— G —

Galletti, F. WEC5, ~~WEPI8~~, WEP08
Gaman, V. FRCB1
Gargana, R. WEC5
Gehrmann, T. ~~THPI3~~, THP03
Gigi, D. WEP17
Gindler, C. THP23
Gladki, G. WEP17
Glege, F. WEP17
Gomez-Ceballos, G. WEP17
Gonzalez Jimenez, P.G.J. ~~WEPI4~~, WEP04
Good, J.D. WEP24
Grespan, F. ~~WEP20~~
Groß, M. WEP24
Gu, K. ~~THP11~~
Guerrero, C.A. **FRCC4**
Guijarro, M. ~~WEPI2~~, WEP02
Guilloud, C. ~~WEPI2~~, WEP02

— H —

Hatje, J. WEC1
Hayashizaki, N. ~~WEP18~~
Hegeman, J. WEP17
Hidas, D.A. FRCC4
Higurashi, Y. WEP30
Hirata, Y. **THP15**
Holzner, A. WEP17
Homs, A. ~~WEPI2~~, WEP02
Hsiao, Y.M. WEP14
Hsu, K.T. WEC2, WEP11, WEP21, THCA3, ~~THPI6~~, ~~THPI7~~, THP06, THP07, THP16, THP20
Hsu, S.Y. WEC2
Hsueh, H.P. FRCA2
Hu, J.H. ~~**WEP35**~~
Hu, K.H. WEC2, ~~WEPI5~~, WEP05, WEP11, WEP21, THCA3, ~~THPI6~~, ~~THPI7~~, THP06, THP07, THP16, THP20
Huang, C.H. WEC2, **THCA3**, ~~THPI7~~, THP07, FRCA2
Huck, H. WEP24

— I —

Iitsuka, Y. THP17
Isaev, I.I. WEP24
Ishi, Y. WEP13
Ishii, H. **WEP26**, THCB2

— J —

Jahn, D. ~~THPI3~~, THP03
Janda, O. THCA5, FRCB1
Jiménez Estupinan, R.J. **WEP16**, **WEP17**
Jin, D.P. WEC4
Jin, H. ~~**THPI2**~~, **THP02**
Jokinen, A. ~~**WEP20**~~, THP15

— K —

Kaji, H. THP17, THP19
Kalantaryan, D.K. WEP24
Kamikubota, N. **THP21**
Kaneko, K. WEP30
Kasugai, A. THP15
Kimura, T. **THP14**
Kino, K. ~~WEP18~~
Klimpki, G. FRCC2
Komiyama, M. **WEP15**, WEP30
Kondo, K. ~~WEP20~~
Koonpong, P. THP10
Kopmann, A. FRCA1
Korth, O. WEC1
Kosuge, T. WEP26, THCB1, **THCB2**
Krasilnikov, M. WEP24
Križnar, I. WEP10
Kru'enko, A. THCA5
Kudo, T. THP19
Kudou, T. THP17
Kumagai, K. WEP15
Kuo, C.H. WEC2, ~~**WEPI5**~~, **WEP05**
Kuriyama, Y. **WEP13**
Kuroda, R. ~~WEP18~~
Kusano, S. THP17, THP19

— L —

Ledeul, A. ~~WEPI7~~, WEP07, **FRCC3**
Lee, D. WEC2, THP20
Lee, S. ~~THPI2~~, THP02
Lei, G. WEC4
Leng, Y.B. WEC3
Lettrich, M. WEP17
Li, C. ~~WEPI6~~, WEP06, ~~WEP12~~, FRCB2
Li, L. WEP34
Li, S. ~~**THP01**~~

ISBN: 978-3-95450-200-4

Li, Y.D. THP18
Liao, C.Y. WEC2, **WEP11**, WEP21, ~~THPI6~~, THP06, THP16, THP20
Lin, M.-C. THP18
Lin, Y.Z. WEP14, THP13, **FRCA2**
Lishilin, O. WEP24
Liu, C. WEP34
Liu, G. ~~WEPI6~~, WEP06, ~~WEP12~~, FRCB2
Liu, H.T. WEP33
Liu, J. **FRCB4**
Liu, Y.J. WEC3
Liu, Z.K. **THP18**
Lo, C.H. THP18
Loisch, G. WEP24
Lomperski, M. THCB1
Ludwig, M. **~~WEPI4~~**, **WEP04**
Lustermann, W. WEP16
Lv, H.H. WEC3
Lv, M.B. WEP34

— M —

Maier-Manojlovic, D. **~~WEPI1~~**, **WEP01**
Marqueta, A. ~~WEP20~~, THP15
Marsching, S. WEP10
Masuda, T. **THCA4**
Matilla, O. THP22
Matsumoto, T. THCA2
Mecionis, A. WEP17
Medvedev, A.M. **WEP22**, FRCC1
Meer, D. FRCC2
Meijers, F. WEP17
Melkumyan, D. **WEP24**
Meschi, E. WEP17
Mexner, W. **WEP10**, **FRCA1**
Meyer, J.M. ~~WEPI2~~, WEP02
Mi, Q.R. WEC3
Miao, H.F. WEC3
Michel, V. **~~WEPI2~~**, **WEP02**
Michelotti, A. WEC5, ~~WEPI8~~, WEP08
Michishio, K. ~~WEP18~~
Michizono, S. THCA2
Milton, S.V. THCB5
Minnig, V. FRCC2
Miura, T. THCA2
Miyahara, F. THP17, **THP19**
Möller, M. WEC1
Mommsen, R.K. WEP17
Monge, R. WEP25
Mori, Y. WEP13
Morovic, S. WEP17
Moya, I. ~~WEP20~~
Müller, A.-S. WEP10
Muroga, T. ~~WEP18~~
Musgrave, I. O. ~~THPI1~~, THP12

— N —

Nagatani, Y. THCB2
Nagatomo, T. WEP30
Nakagawa, T. WEP30
Niehues, G. FRCA1
Nikolova, M. WEP19
Nitani, H. WEP26, THCB2

— O —

O'Dell, V. WEP17
O'Rourke, B.E. **~~WEP18~~**
Ogawa, H. ~~WEP18~~
Ohnishi, J. WEP30
Okita, H. WEP13
Oliveira, P.B.M. ~~THPI1~~, THP12
Oppelt, A. WEP24
Orsini, L. WEP17
Oshima, N. ~~WEP18~~
Otevřel, M. WEP24

— P —

Palnati, V. ~~THPI3~~, THP03
Pang, L.L. WEP34
Papakrivopoulos, I. WEP17
Papillon, E. ~~WEPI2~~, WEP02
Paus, C. WEP17
Penning, J. WEC1
Pepler, D.A. **~~THPI1~~**, **THP12**
Perez, M. ~~WEPI2~~, WEP02
Petitdemange, S. ~~WEPI2~~, WEP02
Petrosyan, B. WEP24
Petrucci, A. WEP17
Phacheerak, W. ~~THPI8~~, THP08
Pieri, M. WEP17
Pistoni, M. WEC5
Pivonka, P. FRCB1
Plötzeneder, B. **THCA5**, FRCB1
Psoroulas, S. FRCC2

— Q —

Qian, H.J. WEP24
Qiang, C.W. WEP34
Qiu, F. **THCA2**

— R —

Rabady, D. WEP17
Racz, A. WEP17
Rapsevicius, V. WEP17
Reis, T. WEP17
Renier, Y. WEP24
Rickens, H.R. WEC1
Ristau, U. **~~THPI3~~**, **THP03**
Rota, L. ~~FRCB3~~

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Rubio-Manrique, S. **WEP25**
Rujanakraikarn, R. ~~**THPI8**~~, **THP08**, **THP10**
Ruprecht, R. WEP10

— S —

Sahoo, G.K. THP23
Sakulin, H. WEP17
Saotome, H.S. THP19
Sasaki, S. THP17
Sato, D. ~~WEP18~~
Sato, K.C. THP21
Satoh, M. THP17, THP19
Savulescu, A. ~~WEPI7~~, WEP07, FRCC3
Scalamera, G. WEP25
Schmelzer, T. WEP10
Schoeneburg, B. WEC1
Schuh, M. WEP10
Schwick, C. WEP17
Segura, G. ~~**WEPI7**~~, **WEP07**, FRCC3
Sei, N. ~~WEP18~~
Semenov, Yu.I. WEP22
Senchenko, A.I. WEP27, WEP28
Serra-Gallifa, X. **THP22**
Shatunov, P.Yu. WEP27, WEP28
Sheng, I.C. THP13, FRCA2
Sheng, Y.B. WEP34
Shinya, T. ~~WEP20~~
Shishido, T. ~~WEP18~~
Shubina, O.S. **WEP27**, **WEP28**
Simelevicius, D. WEP17
Sizov, M. M. WEP22, **FRCC1**
Smale, N.J. WEP10
Sobhani, B. A. ~~WEPI3~~, WEP03
Song, Y. ~~**WEPI6**~~, **WEP06**, **FRCB2**
Spigone, D. WEC5
Stankevicius, M. WEP17
Stant, L.T. **THCB4**
Starostenko, A.A. WEP22, FRCC1
Stecchi, A. **WEC5**, ~~**WEPI8**~~, **WEP08**
Stephan, F. WEP24
Styczen, B. ~~WEPI7~~, WEP07, FRCC3
Sugimoto, M. ~~WEP20~~
Sugimura, H. **THP17**, THP19
Sun, J.R. WEP34
Suzuki, R. ~~WEP18~~
Sys, J. FRCB1
Szczesny, J. WEP19, **THCB1**, THCB2

— T —

Tajima, Y. THP21
Takahashi, H. THP15
Tamura, M. WEP30
Tan Jerome, N. FRCA1
Tanaka, M. ~~WEP18~~
Tanigaki, M. ~~**THPI5**~~, **THP05**
Tempel, J.T. WEP19
Tesprasitte, S. THP10
Tian, Y. FRCC4
Toyokawa, H. ~~WEP18~~
Trdlicka, J. THCA5, FRCB1
Trowitzsch, G. WEP24
Tsyganov, A.S. WEP22

— U —

Uchiyama, A. WEP15, **WEP30**
Uesugi, T. WEP13

— V —

Varela, F. ~~WEPI4~~, WEP04
Vashchenko, G. WEP24
Vazquez Rivera, D. ~~WEPI7~~, WEP07, FRCC3
Vazquez Velez, C. WEP17
Vogelgesang, M. FRCA1

— W —

Walter, A. ~~WEPI3~~, WEP03
Wang, C.-J. WEC2, ~~WEPI5~~, WEP05
Wang, C.H. WEC4
Wang, Ch. THP18
Wang, D. FRCB4
Wang, F. WEP34
Wang, H.Y. WEC3
Wang, J. WEP33
Wang, J.C. FRCB4
Wang, J.G. ~~WEPI6~~, WEP06, ~~WEP12~~, FRCB2
Wang, L. ~~WEP12~~
Wang, W. ~~**FRCB3**~~
Wang, X.L. FRCB4
Watazu, A. ~~WEP18~~
Weber, D.C. FRCC2
Weber, M. ~~WEP20~~
Weber, M. ~~FRCB3~~
Wei, W. ~~**THP11**~~
Weisse, S. WEP19
Wernet, C. WEP17
Wu, C.Y. WEC2, WEP11, WEP21, THCA3, ~~**THPI6**~~, **THP06**, THP16, THP20
Wu, H. THCB1
Wu, Q.J. WEP33

— X —

Xu, W. ~~WEP12~~
Xuan, K. ~~WEPI6~~, WEP06, ~~**WEP12**~~, FRCB2

— Y —

Yamada, S. ~~**THPI4**~~, **THP04**
Yamauchi, H. WEP30

ISBN: 978-3-95450-200-4

Yan, Y.B. **WEC3**
Yang, W.F. WEP34
Yang, Y.C. **WEP14**, THP13, FRCA2
Yeh, M.-S. THP18
Yoshida, S.Y. THP21
Yu, C.L. WEC3
Yu, P.X. WEC3
Yu, T.-C. THP18

— Z —

Zejdl, P. WEP17
Zelepoukine, S. WEP16
Zhang, X. WEP34
Zhang, X.Y. WEP34
Zhao, H. WEC3
Zhao, Q. **WEP34**
Zhao, S.X. WEP34
Zheng, H. **WEP31**, WEP33
Zhu, L.X. WEC4
Zhu, Y.B. WEP34

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4

Institutes List

AIST
Tsukuba, Ibaraki, Japan
- Fujiwara, T.
- Furusaka, M.
- Kino, K.
- Kuroda, R.
- Michishio, K.
- O'Rourke, B.E.
- Ogawa, H.
- Oshima, N.
- Sato, D.
- Sei, N.
- Suzuki, R.
- Tanaka, M.
- Toyokawa, H.
- Watazu, A.

ALBA-CELLS Synchrotron
Cerdanyola del Vallès, Spain
- Avila-Abellan, J.A.
- Broseta, M.
- Cuní, G.
- Fernández, F.F.B.
- Matilla, O.
- Monge, R.
- Rubio-Manrique, S.
- Serra-Gallifa, X.

Ankara University Institute of Accelerator Technologies
Golbasi, Turkey
- Elcim, O.F.

Aquenos GmbH
Baden-Baden, Germany
- Marsching, S.

BINP SB RAS
Novosibirsk, Russia
- Blokhina, K.A.
- Fedotov, M.G.
- Medvedev, A.M.
- Semenov, Yu.I.
- Senchenko, A.I.
- Shatunov, P.Yu.
- Shubina, O.S.
- Sizov, M. M.
- Starostenko, A.A.
- Tsyganov, A.S.

BNL
Upton, Long Island, New York, USA
- Barbour, A. M.
- Bischof, G.
- Escallier, J.
- Farnsworth, R.I.
- Guerrero, C.A.
- Hidas, D.A.
- Sobhani, B. A.
- Tian, Y.
- Walter, A.

CEITEC, BUT
Brno, Czech Republic
- Otevřel, M.

CERN
Meyrin, Switzerland
- Arroyo Garcia, J.A.R.
- Bengulescu, M.
- Brummer, P.
- Da Silva Gomes, D.
- Deldicque, C.
- Dobson, M.
- Doualot, N.
- Farnham, B.
- Fulcher, J.R.
- Gigi, D.
- Gladki, G.
- Glege, F.
- Gonzalez Jimenez, P.G.J.
- Hegeman, J.
- Ledeul, A.
- Lettrich, M.
- Ludwig, M.
- Mecionis, A.
- Meijers, F.
- Meschi, E.
- Orsini, L.
- Rabady, D.
- Racz, A.
- Reis, T.
- Sakulin, H.
- Savulescu, A.
- Schwick, C.
- Segura, G.
- Simelevicius, D.
- Stankevicius, M.
- Styczen, B.
- Varela, F.
- Vazquez Rivera, D.
- Vazquez Velez, C.
- Wernet, C.

CIEMAT
Madrid, Spain
- de la Morena, C.
- Weber, M.

Cosylab
Ljubljana, Slovenia
- Bobnar, J.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4

- Križnar, I.

CSU
Fort Collins, Colorado, USA
- Edelen, A.L.

Dach Consulting GmbH
Brugg, Switzerland
- Dach, M.

DESY
Hamburg, Germany
- Bacher, R.
- Bartkiewicz, P.K.
- Behrens, U.
- Boeckmann, T.
- Clausen, M.R.
- Duval, P.
- Gindler, C.
- Hatje, J.
- Korth, O.
- Lomperski, M.
- Möller, M.
- Penning, J.
- Rickens, H.R.
- Sahoo, G.K.
- Schoeneburg, B.
- Szczesny, J.
- Tempel, J.T.
- Vashchenko, G.
- Wu, H.

DESY Zeuthen
Zeuthen, Germany
- Boonpornprasert, P.
- Chen, Y.
- Good, J.D.
- Groß, M.
- Huck, H.
- Isaev, I.I.
- Kalantaryan, D.K.
- Krasilnikov, M.
- Lishilin, O.
- Loisch, G.
- Melkumyan, D.
- Oppelt, A.
- Petrosyan, B.
- Qian, H.J.
- Stephan, F.
- Trowitzsch, G.
- Weisse, S.

DLS
Oxfordshire, United Kingdom
- Cobb, T.M.
- Stant, L.T.

DPNC
Genève, Switzerland
- Renier, Y.

EJIT
Hitachi, Ibaraki, Japan
- Iitsuka, Y.

Element Aero
Chicago, USA
- Biedron, S.
- Milton, S.V.

Elettra-Sincrotrone Trieste S.C.p.A.
Basovizza, Italy
- Scalamera, G.

ELI-BEAMS
Prague, Czech Republic
- Gaman, V.
- Janda, O.
- Kru'enko, A.
- Pivonka, P.
- Plötzeneder, B.
- Sys, J.
- Trdlicka, J.

EMBL
Hamburg, Germany
- Bueno, M.
- Fiedler, S.
- Gehrmann, T.
- Jahn, D.
- Nikolova, M.
- Palnati, V.
- Ristau, U.

ESRF
Grenoble, France
- Beteva, A.
- Coutinho, T.M.
- Dominguez, M.C.
- Guijarro, M.
- Guilloud, C.
- Homs, A.
- Meyer, J.M.
- Michel, V.
- Papillon, E.
- Perez, M.
- Petitdemange, S.

ETH
Zurich, Switzerland
- Di Calafiori, D.R.S.
- Dissertori, G.
- Djambazov, L.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

- Jiménez Estupinan, R.J.
- Lustermann, W.
- Zelepoukine, S.

F4E
Germany
- Jokinen, A.

Fermilab
Batavia, Illinois, USA
- Andre, J.M.
- Mommsen, R.K.
- Morovic, S.
- O'Dell, V.
- Rapsevicius, V.
- Zejdl, P.

Fusion for Energy
Garching, Germany
- Cara, P.
- Jokinen, A.
- Marqueta, A.
- Moya, I.

IBS
Daejeon, Republic of Korea
- Choi, Y.
- Jin, H.
- Lee, S.

IHEP
Beijing, People's Republic of China
- Chu, C.P.
- Jin, D.P.
- Lei, G.
- Liu, J.
- Wang, C.H.
- Wang, D.
- Wang, J.C.
- Wang, X.L.
- Zhu, L.X.

IMP/CAS
Lanzhou, People's Republic of China
- Cao, H.
- Chen, Y.X.
- Cui, M.H.
- Gu, K.
- Hu, J.H.
- Li, L.
- Liu, H.T.
- Lv, M.B.
- Pang, L.L.
- Qiang, C.W.
- Sheng, Y.B.
- Sun, J.R.
- Wang, F.
- Wang, J.
- Wei, W.
- Wu, Q.J.
- Yang, W.F.
- Zhang, X.Y.
- Zhang, X.
- Zhao, Q.
- Zhao, S.X.
- Zheng, H.
- Zhu, Y.B.

INFN - Roma Tor Vergata
Roma, Italy
- Catani, L.

INFN/LNF
Frascati, Italy
- Bisegni, C.
- Ciuffetti, P.
- D'Uffizi, A.
- De Santis, A.
- Di Pirro, G.
- Galletti, F.
- Gargana, R.
- Michelotti, A.
- Pistoni, M.
- Spigone, D.
- Stecchi, A.

INFN/LNL
Legnaro (PD), Italy
- Fagotti, E.
- Grespan, F.

Institute of Physics of the ASCR
Prague, Czech Republic
- Bastl, P.

Inventec
Taipei, Taiwan
- Chen, T.

ISMA
Ibaraki, Japan
- Muroga, T.
- Shishido, T.

J-PARC, KEK & JAEA
Ibaraki-ken, Japan
- Li, S.
- Sato, K.C.
- Yamada, S.

JAEA/J-PARC
Tokai-Mura, Naka-Gun, Ibaraki-Ken, Japan
- Takahashi, H.

ISBN: 978-3-95450-200-4

JASRI/SPring-8
Hyogo, Japan
- Masuda, T.

Kanto Information Service (KIS), Accelerator Group
Ibaraki, Japan
- Saotome, H.S.
- Yoshida, S.Y.

KEK
Ibaraki, Japan
- Furukawa, K.
- Ishii, H.
- Kaji, H.
- Kamikubota, N.
- Kimura, T.
- Kosuge, T.
- Matsumoto, T.
- Michizono, S.
- Miura, T.
- Miyahara, F.
- Nagatani, Y.
- Nitani, H.
- Qiu, F.
- Sasaki, S.
- Satoh, M.
- Sugimura, H.

KIS
Ibaraki, Japan
- Tajima, Y.

KIT
Eggenstein-Leopoldshafen, Germany
- Balzer, B.M.
- Blomley, E.
- Brunet, M.
- Bründermann, E.
- Caselle, M.
- Fehlinger, C.F.
- Funkner, S.
- Kopmann, A.
- Marsching, S.
- Mexner, W.
- Müller, A.-S.
- Niehues, G.
- Rota, L.
- Ruprecht, R.
- Schmelzer, T.
- Schuh, M.
- Smale, N.J.
- Tan Jerome, N.
- Vogelgesang, M.
- Wang, W.
- Weber, M.

Kyoto University, Research Reactor Institute
Osaka, Japan
- Fuwa, Y.
- Ishi, Y.
- Kuriyama, Y.
- Mori, Y.
- Okita, H.
- Tanigaki, M.
- Uesugi, T.

Lanzhou University
Lanzhou, People's Republic of China
- Liu, C.

LargitData
Taipei, Taiwan
- Chiu, D.

MIT
Cambridge, Massachusetts, USA
- Darlea, G.L.
- Demiragli, Z.
- Gomez-Ceballos, G.
- Paus, C.

Mitsubishi Electric System & Service Co., Ltd
Tsukuba, Japan
- Kudo, T.
- Kudou, T.
- Kusano, S.

National Technical University of Athens
Athens, Greece
- Papakrivopoulos, I.

NSRRC
Hsinchu, Taiwan
- Chan, C.K.
- Chang, C.-C.
- Chang, F.Y.
- Chang, L.-H.
- Chang, M.H.
- Chang, S.W.
- Chang, Y.-T.
- Chen, J.
- Chen, L.J.
- Cheng, C.M.
- Cheng, Y.-S.
- Cheng, Y.T.
- Chiu, P.C.
- Chuang, J.-Y.
- Chung, F.-T.
- Hsiao, Y.M.
- Hsu, K.T.
- Hsu, S.Y.
- Hsueh, H.P.
- Hu, K.H.
- Huang, C.H.
- Kuo, C.H.
- Lee, D.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

- Li, Y.D.
- Liao, C.Y.
- Lin, M.-C.
- Lin, Y.Z.
- Liu, Z.K.
- Lo, C.H.
- Sheng, I.C.
- Wang, C.-J.
- Wang, Ch.
- Wu, C.Y.
- Yang, Y.C.
- Yeh, M.-S.
- Yu, T.-C.

NSU
Novosibirsk, Russia
- Fedotov, M.G.
- Medvedev, A.M.
- Senchenko, A.I.
- Starostenko, A.A.

PSI
Villigen PSI, Switzerland
- Anicic, D.
- Arsov, V.R.
- Bula, C.
- Chevtsov, P.
- Eichin, M.
- Fernandez Carmona, P.
- Klimpki, G.
- Maier-Manojlovic, D.
- Meer, D.
- Minnig, V.
- Psoroulas, S.
- Weber, D.C.

QST
Aomori, Japan
- Hirata, Y.
- Kasugai, A.
- Kondo, K.
- Shinya, T.
- Sugimoto, M.

Rice University
Houston, Texas, USA
- Petrucci, A.

RIKEN Nishina Center
Wako, Japan
- Fujimaki, M.
- Fukunishi, N.
- Higurashi, Y.
- Komiyama, M.
- Kumagai, K.
- Nagatomo, T.
- Nakagawa, T.
- Ohnishi, J.
- Uchiyama, A.

RLNR
Tokyo, Japan
- Hayashizaki, N.

SHI Accelerator Service Ltd.
Tokyo, Japan
- Kaneko, K.
- Tamura, M.
- Yamauchi, H.

SINAP
Shanghai, People's Republic of China
- Liu, Y.J.
- Lv, H.H.
- Mi, Q.R.
- Miao, H.F.
- Wang, H.Y.
- Yu, C.L.
- Yu, P.X.
- Zhao, H.

Skywatch
Taipei, Taiwan
- Chen, W.-C.

SLRI
Nakhon Ratchasima, Thailand
- Dhammatong, Ch.
- Koonpong, P.
- Phacheerak, W.
- Rujanakraikarn, R.
- Tesprasitte, S.

SSRF
Shanghai, People's Republic of China
- Chen, G.H.
- Chen, J.F.
- Ding, J.G.
- Leng, Y.B.
- Yan, Y.B.

STFC/RAL
Chilton, Didcot, Oxon, United Kingdom
- Musgrave, I. O.
- Oliveira, P.B.M.
- Pepler, D.A.

UCLA
Los Angeles, California, USA
- Erhan, S.

ISBN: 978-3-95450-200-4

UCSD
La Jolla, California, USA
- Branson, J.
- Cittolin, S.
- Holzner, A.
- Pieri, M.

USTC/NSRL
Hefei, Anhui, People's Republic of China
- Chen, X.
- Li, C.
- Liu, G.
- Song, Y.
- Wang, J.G.
- Wang, L.
- Xu, W.
- Xuan, K.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4

Participants List

— A —

Anicic, Damir
PSI

— B —

Bacher, Reinhard
DESY

Bastl, Pavel
ELI Beamlines

Biedron, Sandra
Element Aero and UNM

Bischof, Garrett
BNL

— C —

Cao, Hu
IMP/CAS

Catani, Luciano
INFN-Roma Tor Vergata

Chang, Anna
TeraSoft

Chang, Ted
ADLINK

Chang, Yin-Tao
NSRRC

Chen, HongZhe
NSRRC

Chen, Jenny
NSRRC

Chen, Trista
Inventec

Chen, Wei-Chao
Skywatch

Cheng, Yung-Sen
NSRRC

Chiu, David
LargitData

Chiu, Pei-Chen
NSRRC

Chu, Chungming
IHEP

Chuang, Jyun Yan
NSRRC

Clausen, Matthias
DESY

Cuni, Guifre
ALBA

— D —

Duval, Philip
DESY

— E —

Elçim, Ömer Faruk
IAT, Ankara

— F —

Fan, Danny
TeraSoft

Fang, Chun-I
NI

Fernandez Carmona, Pablo
PSI

— G —

Guerrero, Christopher
BNL

— H —

Hirata, Yosuke
QST

Hsu, Kuotung
NSRRC

Hsu, San-Yuang
NSRRC

Hu, Kuohwa
NSRRC

Huang, Chih-Hsien
NSRRC

Hung, Aaron
ADLINK

— I —

Ishii, Haruno
KEK

— J —

Jimenez Estupinan, Raul
ETH Zurich

Jin, Hyunchang
Institute for Basic Science

Jokinen, Antti
LIPAc

— K —

Kamikubota, Norihiko
KEK/J-PARC

Kimura, Takuro
KEK/J-PARC

Komiyama, Misaki
RIKEN Nishina Center

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4

Kosuge, Takashi
KEK

Kuo, Changhor
NSRRC

Kuriyama, Yasutoshi
Kyoto University

— L —

Lee, Shu-Hwa
NSRRC

Liao, Chih-Yu
NSRRC

Liao, Yue
NI

Lin, Yu-Chi
NSRRC

Lin, Yu-Zheng
NSRRC

Liu, Gongfa
NSRL

Liu, Jeffrey
TeraSoft

Liu, Jia
IHEP

Liu, Zong-Kai
NSRRC

Ludwig, Michael
CERN

— M —

Maier-Manojlovic, Dragutin
PSI

Masuda, Takemasa
JASRI

McLean, John
D-tAcq

Medvedev, Aleksei
BINP SB RAS

Melkumyan, David
DESY

Mexner, Wolfgang
KIT

Michel, Vincent
ESRF

Michelotti, Andrea
INFN-LNF

Milne, Peter
D-tAcq

Miyahara, Fusashi
KEK

— O —

O'Rourke, Brian
AIST

— P —

Pepler, Dave
STFC RAL

Peters, Christopher
ANL

Ploetzeneder, Birgit
ELI Beamlines

— Q —

Qiu, Feng
KEK

Qiu, Zhi
Cosylab

— R —

Ristau, Uwe
EMBL

Rubio, Sergio
ALBA

Rujanakraikarn, Roengrut
SLRI

— S —

Sahoo, Gajendra Kumar
DESY

Segura, Gustavo
CERN

Serra-Gallifa, Xavier
ALBA

Shubina, Olga
BINP SB RAS

Sizov, Mikhail
BINP SB RAS

Song, Yifan
NSRL

Stant, Laurence
Diamond Light Source

Stecchi, Alessandro
INFN-LNF

Sugimoto, Takashi
JASRI

Sugimura, Hitoshi
KEK

Szczesny, Jan
DESY

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-200-4

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2018). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

— T —

Tanigaki, Minoru
Kyoto University

Ting, Sean
NI

Tseng, Kuanglee
ADLINK

— U —

Uchiyama, Akito
RIKEN Nishina Center

— W —

Wang, Chii-Jung
NSRRC

Wang, Fanny
JYEBAO

Wang, Weijia
KIT

Wei, Yuan
IMP/CAS

Weng, Chi-Hung
HongHuTech

Wu, Adam
ADLINK

Wu, Chunyi
NSRRC

Wu, John
NI

— X —

Xuan, Ke
NSRL

— Y —

Yamada, Shuei
KEK/J-PARC

Yan, Yingbing
SSRF

Yang, Yi Chen
NSRRC

Youxin, Chen
IMP/CAS

— Z —

Zhao, Qiang
IMP/CAS

Zheng, Hai
IMP/CAS