

9th International Workshop on Personal Computers and Particle Accelerator Controls

December 04-07, 2012

Organised by Variable Energy Cyclotron Centre Kolkata, India

Foreword

The Ninth International Workshop on Personal Computers and Particle Accelerator Controls (PCaPAC-2012) was held during December 04-07, 2012, at Kolkata, India. The workshop was organized by Variable Energy Cyclotron Centre (VECC). The number of submitted abstracts in PCaPAC-2012 was 152; the highest in the history of PCaPAC. It covered varieties of modern aspects of control system design, implementation and technologies used in the scientific fields including particle accelerators, fusion reactors, and telescopes available worldwide. A total of 101 abstracts were finally presented in the workshop including two keynote addresses. The Program Committee classified all abstracts into eleven scientific tracks which were distributed into thirteen scientific sessions during the workshop from Wednesday, December 5 to Friday, December 7, as per their relevance. Two keynote speeches were scheduled in the Inaugural session after the warm welcome address by Ranadhir Dey, Head, CPIESG, VECC followed by the scintillating presentation of Chairman, PCaPAC-2012, about the Workshop. "The CSS Story" was presented by the first keynote speaker, Matthias R. Clausen from DESY, Germany and Y S Mayya from BARC, India, talked about "Evolution of Control Systems for Large Telescopes and Accelerators: A retrospective". The vote of thanks was offered by the workshop Secretary, Dr. Sarbajit Pal. A total of 139 registered participants from 13 countries discussed vigorously about the subjects related to up-to-date control systems and its future, in the scientific sessions.

There were two pre-workshop tutorials held on Tuesday, December 4, in parallel sessions before the inauguration of the workshop. Dr. Norihiko Kamikubota from High Energy Accelerator Research Organization, Japan conducted the tutorial on the subject "EPICS & CSS Tutorial Seminar and Hands-on" while "Programming EPICS enabled Real-Time and FPGA" was arranged by National Instruments. A few leading-edge equipment providers from all over the world exhibited their commercial products during the workshop and the exhibition was inaugurated by the Chairman of the workshop on December 5.

The conference consisted of a set of plenary oral sessions with two poster sessions. The plenary sessions consisted of invited as well as contributed orals. The different plenary sessions with a total of 38 technical papers including 13 invited and 25 contributed oral were chaired by the leading experts of the related fields across the globe. In the afternoon sessions of 27 posters on Wednesday and 36 on Thursday, close interaction was observed among the participants to exchange their ideas.

A banquet dinner was held at "Eco Hub Conclave", Newtown on Wednesday and an instrumental duet featuring the Sitar and the Indian bamboo Flute was staged on that evening. A cultural programme of "Odissi" and "Kathak" dance was arranged on Thursday which was followed by dinner, hosted by Director, VECC.

In the valedictory session, Wolfgang Mexner from KIT, Germany, declared the organization of PCaPAC-2014 at Karlsruhe. The Isamu Abe Prize was presented to two winners, Rajendra Nath Dutta of IUAC and Shantonu Sahoo of VECC at the valedictory session.

This proceedings contain not only the 101 contributions received from the participants but also include the pdf copies of the orals and posters presented at the workshop. The editorial team, led by Volker Schaa worked very hard to improve the quality and uniqueness of papers for this proceedings to be published under JACoW collaboration.

The successful and memorable event of the workshop was the result of hard work and support of many people. It all started in August, 2011, when VECC accepted the proposal from International Advisory Committee, PCaPAC, to organize the ninth one in its series. We would like to thank all the members of the National and International Advisory Committees, and especially the Local Organizing Committee. Thanks are also extended to the colleagues of VECC who put their untiring efforts and support directly or indirectly to ensure the success of PCaPAC-2012, organized within this short period.

We also wish to thank the JACoW Team members especially Volker Schaa, Christine Petit-Jean-Genaz, Matt Arena, Ivan Andrian and Takashi Kosuge for their support to publish this proceedings under JACoW collaboration.

The quality talks and discussion by the delegates made the standard of this workshop very high. The feedback received from the participants ensured a grand success of PCaPAC-2012. The participants look forward to meeting again at PCaPAC-2014 in KIT, Germany.

leese

Debranjan Sarkar Chair, PCaPAC-2012

Contents

_		
Pr	eface	i
	Foreword	iii
	Contents	V
	Committees	ix
Pa	ipers	1
	WEKA01 – The CSS Story	1
	WEIC01 – Web2cToGo: Bringing the Web2cToolkit to Mobile Devices	4
	WECC02 – EPICS Channel Access Using WebSocket	7
	WECC03 – Qt Based GUI System for EPICS Control Systems	10
	WEPD01 – Data Logging System Upgrade for Indus Accelerator	12
	WEPD03 – Control System Studio Archiver with PostgreSQL Backend: Optimizing Performance and Reli-	
	ability for a Production Environment	15
	WEPD09 – Fast Data Acquisition System for Booster Supplies Readback	18
	WEPD10 – Embedded CAMAC Controller: Hardware/Software Co-optimization for High Throughput	20
	WEPD11 – Client Server Architecture Based Embedded Data Acquisition System on PC104	23
	WEPD12 – A Large Channel Count Multi Client Data Acquisition System for Superconducting Magnet	
	System of SST-1	26
	WEPD13 – Serial Multiplexed Based Data Acquisition and Control System	29
	WEPD14 – VEPP-2000 Logging System	32
	WEPD16 – Development of Data Acquisition Software for VME Based System	35
	WEPD18 – Microcontroller Based DAQ System for IR Thermography by Hot and Cold Water Flow	37
	WEPD19 – Smart Structured Measurement Process for Versatile Synchrotron Beamline Data at ANKA .	40
	WEPD22 – Post-Mortem Analysis of BPM-Interlock Triggered Beam Dumps at PETRA-III	43
	WEPD23 – Design & Implementation Of LabVIEW TM Based GUI for Remote Operation and Control of	
	Excimer Laser for Plasma Wakefield Accelerator Experiment	46
	WEPD24 – STARS on Android	51
	WEPD25 – Development of EPICS Channel Access Embedded ActiveX Components for GUI Development	54
	WEPD26 – Development of Fast Controls for Beam Wire Scanner for SuperKEKB	57
	WEPD27 – Graphical User Interface (GUI) for Testing CAMAC modules	60
	WEPD28 – Re-envisioning the Operator Consoles for Dhruva Control Room	62
	WEPD33 – Embedded PC Based Controller for Use in VME Bus Based Data Acquisition System	65
	WEPD34 – A Low-Cost High-Performance Embedded Platform for Accelerator Controls	68
	WEPD38 – A wireless control system for the HTS-ECRIS, PKDELIS and low energy beam transport	71
	WEPD39 – Development of an Ethernet Enabled Microcontroller Based Module for Superconducting Cy-	
	clotron ECR Beamline Control	73
	WEPD43 – A New Scheme for Direct Estimation of PID Controller Parameters	76
	WEPD44 – FPGA Data Block FIFO for the APS ID Measurement System	79
	WEPD47 – Low-cost EPICS Control Using Serial-LAN Module XPort	81
	WEPD48 – Facility-Wide Synchronization of Standard FAIR Equipment Controllers	84
	WEPD52 – Diamond Light Source Control Systems Relational RDB	87
	THIA02 – Current Status and Upgrade Plan of the Data-Acquisition System at SACLA	90
	THIA03 – The IUAC Tandem-LINAC Control System	94
	THCA04 – An Update on ConSys Including a New LabVIEW FPGA Based LLRF System	97
	THCA05 – PLC-based Control System for 10 MeV Linear Accelerator at EBC Kharghar, BARC	100
	THCA06 – Status of the Ultra Fast Tomography Experiments Control at ANKA	103
	THCB01 – HyperArchiver: an Evolution of EPICS Channel Archiver	106
	THCB02 – EPICS MySQLArchiver - Integration Between EPICS and MySQL	109
	THCB03 – Using Memcached as Real-time Database in the SPARC Control System	112
	THIB04 – Control System Interoperability, an Extreme Case: Merging DOOCS and TINE	115
	THIC01 – Tango for Experiment Control	118
	THCC02 – Controls Architecture for the Diagnostic Devices at the European XFEL	121
	THCC03 – PC Based Real Time Data Exchange on 10GbE Optical Network Using RTOS	124

ment System for High Pressure Physics Experiment at Indus-2 Beam Line	128
THCD05 A Elovible and Toetable Software Architecture: Applying Presenter First to a Device Server for	120
the DOOCS Accelerator Control System of the European XEEI	121
THCD06 Design Development and Analysis of a Comprehensive Open Source System for Presetive	101
Management of Society Appage of a Completion Network	104
TUDDOO What it Takes to Make a System Daliable	134
THPD02 – What it Takes to Make a System Reliable	139
THPD03 – PLC Controlled Search & Secure Safety Interlock System for Accelerator	142
THPD04 – Machine Throughput Improvement Achieved Using Innovative Control Technique	144
THPD05 – Design and Analysis of Second Harmonics Modulator for DC Current Transformer	145
THPD06 – FLogbook: From Concept to Realization	148
THPD09 – Development of a Monitoring System for the FL-net Protocol	151
THPD10 – Modular Beam Diagnostics Instrument Design for Cyclotrons	154
THPD11 – Facility Monitoring System using Storage Area Network for VEC and SCC	157
THPD12 – Design and Implementation of an IEEE 802.15.4/ZigBee based Star Network for Data Acquisi-	
tion and Monitoring	160
THPD13 – SocketCAN Device Support for EPICS IOCs	163
THPD14 – Status of the Migration of the S-DALINAC Accelerator Control System to EPICS	166
THPD15 – Multichannel High Voltage Power Supply Controls Solution Using Compact Distributed Ethernet	
Based Boards and Qt Based GUI	169
THPD16 – Fast Digital Feedback Control Systems for Accelerator RF System using FPGA	172
THPD17 – API Manager Implementation and its Use for Indus Accelerator Control	175
THPD18 – Adaptive Fuzzy Control for Transfer Channels in Particle Accelerators	178
THPD19 – Drive System Control for Kolkata Superconducting Cyclotron Extraction System	181
THPD20 – BE Distribution and Control System for Accelerators of the VEC-BIB Facility	184
THPD21 – Testing of Inductive Output Tube based BE Amplifier for 650 MHz SBE Cavities	187
THPD22 - Controls for a 10 Petawatt Class Laser Facility	190
THE D22 - Controls for a TOT etawatt Class Laser Facility	102
	132
TUPD27 Control Schome for Demote Operation of Magnet Dewer Supplies for Infrared Free Electron	
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron	105
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser	195
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD29 – Use Values Control System for 750 keV DC Accelerator	195 197
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Represent of the UND of the Accelerator 	195 197 200
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD32 – Operation of the Structure for Spectroscopy Diagnostics of SST-1 	195 197 200 203
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility 	195 197 200 203 206
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System 	195 197 200 203 206 208
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Pu- 	195 197 200 203 206 208
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m}Tc from (n, γ)⁹⁹Mo 	195 197 200 203 206 208 211
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m}Tc from (n, γ)⁹⁹Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System 	195 197 200 203 206 208 211 211
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m}Tc from (n, γ)⁹⁹Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System 	195 197 200 203 206 208 211 211 214 217
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m}Tc from (n, γ)⁹⁹Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD44 – The CS framework as a Control System for the HITRAP Facility at GSI 	195 197 200 203 206 208 211 214 217 219
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron LaserTHPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC AcceleratorTHPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control SystemsTHPD33 – Qt Based Control System Software for Low Energy Accelerator FacilityTHPD35 – Modeling and Simulation of Indus-2 RF Feedback Control SystemTHPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of 99m Tc from $(n, \gamma)^{99}$ MoTHPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply SystemTHPD43 – Electron Cyclotron Resonance Ion Source Control SystemTHPD44 – The CS framework as a Control System for 30MeV RF SourceTHPD45 – Overview of Control System for 30MeV RF Source	195 197 200 203 206 208 211 214 217 219 222
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m}Tc from (n, γ)⁹⁹Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD44 – The CS framework as a Control System for 30MeV RF Source THPD46 – Simulation Analysis of Analog IQ based LLRF Control of RF Cavity 	195 197 200 203 206 208 211 214 217 219 222 225
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m} Tc from (n, γ) ⁹⁹ Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD44 – The CS framework as a Control System for 30MeV RF Source THPD45 – Overview of Control System for 30MeV RF Source THPD46 – Simulation Analysis of Analog IQ based LLRF Control of RF Cavity THPD47 – Introduction of Non-Standard EPICS Controllers	195 197 200 203 206 208 211 214 217 219 222 225 227
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m} Tc from (n, γ) ⁹⁹ Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD44 – The CS framework as a Control System for the HITRAP Facility at GSI THPD45 – Overview of Control System for 30MeV RF Source THPD46 – Simulation Analysis of Analog IQ based LLRF Control of RF Cavity THPD47 – Introduction of Non-Standard EPICS Controllers THPD48 – Reachability in a Finite Distributed System Protocol Model by Backward Traversal	195 197 200 203 206 208 211 214 217 219 222 225 227 230
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m} Tc from (n, γ) ⁹⁹ Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD44 – The CS framework as a Control System for the HITRAP Facility at GSI THPD45 – Overview of Control System for 30MeV RF Source THPD46 – Simulation Analysis of Analog IQ based LLRF Control of RF Cavity THPD47 – Introduction of Non-Standard EPICS Controllers THPD48 – Reachability in a Finite Distributed System Protocol Model by Backward Traversal THPD49 – Design Considerations for Development of Distributed Data Acquisition and Control System	195 197 200 203 206 208 211 214 217 219 222 225 227 230
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m}Tc from (n, γ)⁹⁹Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD44 – The CS framework as a Control System for 30MeV RF Source THPD45 – Overview of Control System for 30MeV RF Source THPD46 – Simulation Analysis of Analog IQ based LLRF Control of RF Cavity THPD47 – Introduction of Non-Standard EPICS Controllers THPD48 – Reachability in a Finite Distributed System Protocol Model by Backward Traversal THPD49 – Design Considerations for Development of Distributed Data Acquisition and Control System (DDACS) for Radio-active Ion Beam (RIB) Facility 	195 197 200 203 206 208 211 214 217 219 222 225 227 230 234
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser	195 197 200 203 206 208 211 214 217 219 222 225 227 230 234 234
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m} Tc from (n, γ) ⁹⁹ Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD44 – The CS framework as a Control System for the HITRAP Facility at GSI THPD46 – Simulation of Non-Standard EPICS Controllers THPD47 – Introduction of Non-Standard EPICS Controllers THPD48 – Reachability in a Finite Distributed System Protocol Model by Backward Traversal THPD49 – Design Considerations for Development of Distributed Data Acquisition and Control System (DDACS) for Radio-active Ion Beam (RIB) Facility THPD49 – FPGA Based Amplitude Control System for the FAIR Facility	195 197 200 203 206 208 211 214 217 219 222 225 227 230 234 239 242
THPD27 - Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 - A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 - High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 - Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 - Qt Based Control System Software for Low Energy Accelerator Facility THPD35 - Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 - An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m} Tc from (n, γ) ⁹⁹ Mo THPD40 - Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 - Electron Cyclotron Resonance Ion Source Control System THPD44 - The CS framework as a Control System for the HITRAP Facility at GSI THPD45 - Overview of Control System for 30MeV RF Source THPD47 - Introduction of Non-Standard EPICS Controllers THPD48 - Reachability in a Finite Distributed System Protocol Model by Backward Traversal THPD49 - Design Considerations for Development of Distributed Data Acquisition and Control System THPD49 - FPGA Based Amplitude Control System for Accelerating Cavities THPD40 - FPGA Based Amplitude Control System for Accelerating Cavities THPD49 - The New White Rabbit Based Timing System for the FAIR Facility THPD40 - FPGA Based Amplitude Control System for Accelerating C	195 197 200 203 206 208 211 214 217 219 222 225 227 230 234 239 242
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser	195 197 200 203 206 208 211 214 217 219 222 225 227 230 234 239 242 245
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser	195 197 200 203 206 208 211 214 217 219 222 225 227 230 234 239 242 245 245
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m} Tc from (n, γ) ⁹⁹ Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD44 – The CS framework as a Control System for the HITRAP Facility at GSI THPD45 – Overview of Control System for 30MeV RF Source THPD46 – Simulation Analysis of Analog IQ based LLRF Control of RF Cavity THPD47 – Introduction of Non-Standard EPICS Controllers THPD49 – Design Considerations for Development of Distributed Data Acquisition and Control System (DDACS) for Radio-active Ion Beam (RIB) Facility THPD50 – FPGA Based Amplitude Control System for Accelerating Cavities FRIA01 – The New White Rabbit Based Timing System for the FAIR Facility FRCA02 – Status Report, Future Plans and Maintenance Issues of VME Based Cryogenic Control System at IUAC	195 197 200 203 206 208 211 214 217 219 222 225 227 230 234 239 242 245 248 251
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m} Tc from (n, γ) ⁹⁹ Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD44 – The CS framework as a Control System for 30MeV RF Source THPD45 – Overview of Control System for 30MeV RF Source THPD47 – Introduction of Non-Standard EPICS Controllers THPD48 – Reachability in a Finite Distributed System Protocol Model by Backward Traversal THPD49 – Design Considerations for Development of Distributed Data Acquisition and Control System (IDDACS) for Radio-active Ion Beam (RIB) Facility THPD50 – FPGA Based Amplitude Control System for the FAIR Facility FRCA02 – Status Report, Future Plans and Maintenance Issues of VME Based Cryogenic Control System at IUAC FRCA03 – Development of the Car-borne Survey System KURAMA FRCA04	195 197 200 203 206 208 211 214 217 219 222 225 230 234 239 242 245 248 251
THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD32 – Progress of the JINR e-Linac Accelerator Test-Bench Control Systems THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m} Tc from (n, γ) ⁹⁹ Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD45 – Overview of Control System for 30MeV RF Source THPD46 – Simulation Analysis of Analog IQ based LLRF Control of RF Cavity THPD47 – Introduction of Non-Standard EPICS Controllers THPD48 – Reachability in a Finite Distributed System Protocol Model by Backward Traversal CDDACS) for Radio-active Ion Beam (RIB) Facility THPD49 – The New White Rabbit Based Timing System for the FAIR Facility FRCA02 – Status Report, Future Plans and Maintenance Issues of VME Based Cryogenic Control System at IUAC FRCA03 – Development of the Car-borne Survey System KURAMA FRCA04 – Control System for BARC-TIFR Pelletron FRCA04 – C	195 197 200 203 206 208 211 214 217 219 222 225 227 230 234 239 242 245 248 251
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser HPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m} Tc from (n, γ)⁹⁹Mo THPD40 – Instrumentation Architecture for ITER Diagnostic Neutral Beam Power Supply System THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD44 – The CS framework as a Control System for the HITRAP Facility at GSI THPD45 – Overview of Control System for 30MeV RF Source THPD46 – Simulation Analysis of Analog IQ based LLRF Control of RF Cavity THPD47 – Introduction of Non-Standard EPICS Controllers THPD49 – Design Considerations for Development of Distributed Data Acquisition and Control System (DDACS) for Radio-active Ion Beam (RIB) Facility THPD50 – FPGA Based Amplitude Control System for the FAIR Facility FRCA02 – Status Report, Future Plans and Maintenance Issues of VME Based Cryogenic Control System at IUAC FRCA03 – Development of the Car-borne Survey System KURAMA FRCA04 – Control System for BARC-TIFR Pelletron FRCA04 – Control System for BARC-TIFR Pelletron FRCB01 – Maintaining an Effective and Efficient Control System for The Electromagnetic Calorimeter of the Compact Muon Solenoid Experiment During Long-term CERN Large Hadron Collider Operations FRCB02 – Development of the Control System for PEFP 100-MeV Proton Linear Accelerator 	195 197 200 203 206 208 211 214 217 219 222 225 227 230 234 239 242 245 248 251 254 254
 THPD27 – Control Scheme for Remote Operation of Magnet Power Supplies for Infrared Free Electron Laser THPD28 – A Distributed CAN Bus Based Embedded Control System for 750 keV DC Accelerator THPD30 – High Voltage Controller System for Spectroscopy Diagnostics of SST-1 THPD33 – Qt Based Control System Software for Low Energy Accelerator Facility THPD35 – Modeling and Simulation of Indus-2 RF Feedback Control System THPD36 – An Embedded System Based Computer Controlled Process Automation for Recovery and Purification of ^{99m}Tc from (n, γ)⁹⁹Mo THPD43 – Electron Cyclotron Resonance Ion Source Control System THPD44 – The CS framework as a Control System for the HITRAP Facility at GSI THPD45 – Overview of Control System for 30MeV RF Source THPD46 – Simulation Analysis of Analog IQ based LLRF Control of RF Cavity THPD47 – Introduction of Non-Standard EPICS Controllers THPD49 – Design Considerations for Development of Distributed Data Acquisition and Control System (DDACS) for Radio-active Ion Beam (RIB) Facility THPD50 – FPGA Based Amplitude Control System for Accelerating Cavities FRIA01 – The New White Rabbit Based Timing System for Accelerating Cavities FRCA03 – Development of the Car-borne Survey System KURAMA FRCA04 – Control System for BARC-TIFR Pelletron FRCB01 – Maintaining an Effective and Efficient Control System for the Electromagnetic Calorimeter of the Compact Muon Solenoid Experiment During Long-term CERN Large Hadron Collider Operations FRCB02 – Development of the Control System for PEFP 100-MeV Proton Linear Accelerator	195 197 200 203 206 208 211 214 217 219 222 225 227 230 234 239 242 245 248 251 254 257 260

FRCB04 – VEPP-2000 Collider Control System	263 268
FRCC02 – A FPGA Based High Speed Data Acquisition Card	271
FRCC03 – Development and Performance Analysis of EPICS Channel Access Server on FPGA based	
Soft-core Processor	274
FRCC04 – Digital Pulse Processing Techniques for High Resolution Amplitude Measurement of Radiation	
Detector	279
FRID01 – Introducing the ICHAOS Control Systems Framework	282
FRCD02 – Process Control for Parallel Run of Two Helium Liquefiers at VEC Centre, Kolkata	285
Appendices	289
List of Authors	289
Institutes List	293
Participants List	298

Committees

International Advisory Committee

Bacher Reinhard - DESY (Germany)	Gurd Dave - Oak Ridge (retired)(US)
Baer Ralph - GSI (Germany)	Heron Mark - Diamond (UK)
Bhandari Rakesh Kumar- VECC (India)	Hunt Steve - Alceli (Switzerland)
Catani Luciano - INFN (Italy)	Kamikubota Norihiko - KEK (Japan)
Chevtsov Pavel - Jefferson Labs (US)	Klora Jorg - CELLS (Spain)
Chunhong Wang - IHEP (China)	Kosuge Takashi - KEK (Japan)
Clausen Matthias - DESY (Germany)	Kurokawa Shin-ichi - Cosy Lab (Japan)
Duval Philip - DESY (Germany)	Lange Ralph - HZB / BESSY II (Germany)
Farnsworth Richard- ANL (US)	Matias Elder - CLS (Canada)
Gupta P.D RRCAT (India)	Mayya Y. S ECIL (India)
Mexner Wolfgang - KIT (Germany)	
Plesko Mark - JSI (Slovenia)	
Quock Deborah - APS (US)	
Roy Amit - IUAC (India)	
Sinha R.K BARC (India)	
Shen Guobao - BNL (US)	
Song Young-gi - PEFP (Korea)	
Verstovsek Igor - Cosy Lab (Slovenia)	

Yamashita Akihiro - Spring 8 (Japan)

National Advisory Committee

Chandra Umesh - NPCIL Majumdar A. K. - IIT-KGP Navathe C. P. - RRCAT Pithawa C. K. - BARC Sarkar Debranjan - VECC Satya Murty S. A. V. - IGCAR

Local Organizing Committee

Bhattacharjee Tanushyam Bhattacharyya Tamal Bhole R. B. Chaddha Niraj Datta Kaushik Fatnani Pravin- RRCAT Mandal Aditya Nabhiraj P. Y. Pal Sandip Pal Sarbajit (Secretary) Roy Anindya Samanta Tapas Sarkar Debranjan (Chairman)

THE CSS STORY

M. Clausen, J. Hatje, J. Penning, DESY, Hamburg, Germany

Abstract

Control System Studio (CSS) is designed to serve as an integration platform for engineering and operation of today's process controls as well as machine controls systems. Therefore CSS is not yet another replacement of existing operator interfaces (OPI) but a complete environment for the control room covering alarm management, archived data displays diagnostic tools and last not least operator interfaces. In addition we decided to use CSS as the platform for the whole engineering chain configuring EPICS based process control databases, configuring and managing the I/O, editing state notation programs, configuring role based access rights and many more. Due to the ease of use of CSS as an Eclipse based product, we decided to use the CSS core also for all our stand alone processes. This helped us to reduce the diversity of running products/ processes and simplified the management. In this presentation we will describe our experience with CSS over the last two years. How we managed the transition from old displays to new ones, how we changed our alarm/ message philosophy and last not least which lessons we learned.

INTRODUCTION

The development of a new tool set was initiated by the new project at DESY – the European XFEL. This project will lead us through the next decades to come. All existing hard and software for cryogenic and utility controls had to go through a verification process to decide whether components can pass 'as is' or whether they have to go through a process of refurbishment or even new design.

For the existing operator tools it was soon clear that they have to be replaced by a new tool set.

NEW TECHNOLOGIES FOR NEW OPERATOR TOOLS

To prepare the new tool set for the future it was decided to use Java as the programming language. The next question to be answered was: Which development environment tool shall be used? This covers also the question about the core technologies to be used. Two candidates were investigated: Eclipse and NetBeans.

At the time when Eclipse 3.1 was launched it was chosen for the CSS core technologies. Many more basic technology decisions had to be made. These were identified in a workshop about three years before the first beta release of CSS.

INITIATING COLLABORATION

As a result of the workshop the CSS collaboration was started between the initial partners DESY and SNS. In addition two companies were involved in the exploration of new technologies. But what is the best way to explore the new fields? How can one set up a contract with a company when the details cannot be specified? We had to walk on new ground with respect to contracting and project management. New techniques known from extreme programming were used to setup and run these development projects.

WATERFALL OR XP?

This is not a question any more. Waterfall specifications are by definition always out of date and are outdated. But which kind of extreme programming shall be followed? A very basic concept is working in iterations. Iteration steps are small they last only a few days or weeks. The work to be done in these steps is defined in tasks. The results are checked after each iteration step. Especially when new technologies must be explored it is important to stay in close contact and verify that the team is still on the right path. Waterfall – or long iteration steps will fail in this respect.



Figure 1: The Agile Methodology ©

© Wikipedia under Creative Commons Licence

Identifying Risk Factors

Risk factors first – this is one of the concepts for the iteration steps. This will make sure that those tasks which are critical for the whole project will be dealt with at first. Other tasks will run on a lower priority and will be solved later. In our case we had to check whether the GEF (graphical editing framework) will be adequate for the new operator interface. Several criteria needed exploration: Will it be possible to use GEF not only in edit mode but also in runtime mode? What will the performance of GEF dynamic widgets be? (several thousand updates per second were specified) And – can the layout created in the GEF editor be stored in XML and read back in?

All of these questions were checked and answered in several iterations throughout the development process. Each of these was not fully implementing the full functionality of the final product but just enough to verify the proof of concept of the individual step. All of them worked as a cut through the technology layers proofing that GEF can be used as the backbone for the new graphical user interface for the operators.

CHANGE MANAGEMENT: PATCHWORK OR POPPER?

During the course of software maintenance and change management one will experience several different kinds of approaches how changes get implemented.

Patchwork

If software changes are integrated 'on the fly' into existing code they will most likely be not well integrated. Many changes will in the end destroy the initial structure of the code base. A patchwork of adjacent code snippets will also make testing difficult – if not impossible.

It will be very likely that such code will not be accompanied by the required JUnit test cases. In some cases such code might even break existing functionalities.

Popper

A well structure code will ease modifications of the code. Even functionality/ code extensions will be easy to integrate when well defined interfaces (popper-alike) are foreseen in the design. Well defined interfaces are also mandatory for good test coverage with JUnit test.

TEST TEST TEST

Test Driven Development

Writing the test code before the production code gets written is the ideal world but the real world differs from that. If this order does not work it is important to keep in mind – and actually write – the test code afterwards. Writing JUnit test code is initially as popular as writing documentation. As soon as the benefit becomes obvious it will be written as an integrated part of the application code itself.

Healthy Collaborative Pressure

Any chain is as strong as the weakest link. This phrase is also true for software packages. The bigger these packages get the more complicated it gets to diagnose errors and potential problems. Developing code in collaboration with others will put a healthy pressure on all developers to provide stable code which does not interfere with anybody else's implementations.

Dependencies with 3rd Party Code

If your code is dependent from other code which is not developed in-house it is mandatory to 'protect' your own code base from that code by adding JUnit tests on your side to detect problems when new versions of that code (which you do not support yourself) gets committed.

CONTINUOUS INTEGRATION

Especially in a collaborative/ distributed development environment it is necessary to check whether your final product can still be built taking all of the latest commits into account. A special server is set up at DESY and on some of the other collaborator's sites to perform this job. It is continuously checking in all changes from the code repository. The new code and all of the existing code will be compiled. This method will identify code that does not compile and also incompatibilities between different packages. Ideally your own changes will be committed after you have checked your code with the latest core code from the collaboration.



Figure 2: Continuous Integration on a Jenkins Server @ DESY.

CONTINUOUS TESTING

The second step after the continuous integration of software changes is to check for runtime errors. JUnit test are meant to check for errors beyond compilation. While most of the JUnit tests are typically started manually in everybody's own environment the intent of 'Continuous Testing' is to run these test on the same server which performs the tests after each continuous integration test. Setting up these tests on a server platform is a complex task especially when database servers or control system protocols are involved. But – there is no doubt – any time invested in testing and integration pays off in double in time which is NOT spent in later debugging and maintenance and in the end unhappy end users.

THE COLLABORATION TODAY

Since the early days in 2006 the collaboration has grown. Besides DESY and SNS, BNL, ITER and KEK have joined. The use of shared applications differs from installation to installation. Many CSS instances are running outside the CSS collaboration. There is no need to become a core developer in order to actually run CSS.

SO WHAT IS CSS?

CSS can be configured in many different was. At DESY we decided to make things as simple as possible. So we create just one type of CSS product. Where product means the CSS core and a rich set of plugins. All of these files are zipped to a single zip file which can be unpacked on the end user's machine. In addition to these complete packages it is also possible to just update you CSS instance. The Eclipse update mechanism allows this kind of incremental update when an update site is set up. At DESY this update site is populated with the latest versions of individual plugins. So only this selected plugin gets its update while the rest of CSS stays the same. The core installation with all its settings and configuration files will not be altered.

Even though we are only providing a single kind of CSS product, we have individual user groups which have different views on CSS and are using completely different sets of plugins for their work. This is possible because a role based authentication scheme is implemented in CSS. Logging in to CSS with Kerberos authentication will make sure that only authorized people may perform certain actions on the control system or make changes in the configuration. What is CSS ...

... for the Operator

The main plugins for the operator are the tools he needs to control the equipment and to get information when something fails. In addition he wants to look back in history what happened at a certain time or before an event occurred.

The typical tools are: The synoptic display SDS (Synoptic Display Studio), the alarm toolset consisting of

the alarm table, alarm tree, message tree and archive message table and the trend browser.

... for the Engineer

The engineer actually is using CSS to configure the EPICS databases using the database creation tool DCT. The EPICS records need proper addresses to read and write from the distributed I/O system. The I/O is configured in the I/O configurator. Sequence programs are running in the front end controllers (IOCs – Input Output Controller). They are actually programmed using the state notation language (snl) editor.

All of these programs are plugins to the CSS toolkit.

... for the Developer

The developer is using the Eclipse Java IDE to create CSS which is based on Eclipse core technology. So - it is a toolkit to generate control system applications. Together with the other collaborators CSS is also a collaboration.

... for the Manager

The manager finds a stable basis he can rely on. New applications can be added easily to the CSS core. After a steep learning curve to create the first plugin it pays off in the following applications to come. A homogeneous look and feel ensures that new plugins will be easily adopted be the end user. There is no different behaviour of each implementation.

Last not least the collaboration has already reasonable support from industry. New developments can also be carried out by industrial partners. The latter is a successful practise at DESY.

SUMMARY

The CSS idea has found its way through the world of (mostly EPICS) control system installations. The community is slowly growing. The collaborators have found a practical non-bureaucratic way to define and implement incremental improvements in the CSS core while keeping as much freedom as possible on the individual applications.

We can be proud of this success!

ACKNOWLEDGMENT

CSS as it stands would not exist without the continuous effort of the collaboration to make it a better product in their day to day developments.

Thanks to the all of the collaborators!

Web2cToGo: BRINGING THE Web2cToolkit TO MOBILE DEVICES

R. Bacher, DESY, Hamburg, Germany

Abstract

The Web2cToolkit is a collection of Web services. It enables scientists, operators and service technicians to supervise and operate accelerators and beam lines through the World Wide Web. The toolkit includes a synoptic display viewer and editor, an archive viewer, a messenger service, a logbook facility, an administration manager and an HTTP gateway to control systems. Recently, a novel view (Web2cToGo) has been added which is especially designed for mobile devices such as tablet computers or smartphones running iOS, Android or other mobile operation systems. Web2cToGo is a frame which embeds instances of all kinds of Web2c tools. It provides a singlesign-on user authentication and authorization procedure. Web2cToGo supports single- or multiple-touch user gestures and is available as a platform-independent browser-based Web application or as a platformdependent native app. This paper describes the conceptual design of Web2cToGo and the technologies used behind the scenes as well as the experiences gathered so far. It presents an outlook of on-going developments including user-device interaction based on voice recognition.

INTRODUCTION

The Web2cToolkit [1] [2] is a collection of Web services including

(1) *Web2c Synoptic Display Viewer*: Interactive synoptic live display to visualize and control accelerator or beam line equipment,

(2) *Web2c Archive Viewer*: Web form to request data from a control system archive storage and to display the retrieved data as a chart or table,

(3) Web2c Messenger: Interface to E-Mail, SMS and Twitter,

(4) *Web2c Logbook*: electronic logbook with auto-reporting capability,

(5) *Web2c Manager*: administrator's interface to configure and manage the toolkit,

(6) *Web2c Editor*: graphical editor to generate and configure synoptic displays, and

(7) *Web2c Gateway*: application programmer interface (HTTP-gateway) to all implemented control system interfaces.

Web2cToolkit is a framework for Web-based Rich Client Control System Applications. It provides a userfriendly look-and-feel and its usage does not require any specific programming skills. By design, the Web2cToolkit is platform independent. Its services are accessible through the HTTP protocol from every valid network address if not otherwise restricted. A secure single-sign-on user authentication and authorization procedure with encrypted password transmission is provided. Registered and so-called privileged users have more rights compared to ordinary users (read-only permission).

The Web 2.0 paradigms and technologies used include a Web server, a Web browser, HTML (HyperText Markup Language), CSS (Cascading Style Sheets) and AJAX (Asynchronous JavaScript And XML). The interactive graphical user interface pages are running in the client's Web browser. The interface is compatible with all major browser implementations including mobile versions. The Web2cToolkit services are provided by Java servlets running in the Web server's Java container. The communication between client and server is asynchronous. All third-party libraries used by the Web2cToolkit are open-source.

The Web2cToolkit provides interfaces to major accelerator and beam line control systems including TINE [3], DOOCS [4], EPICS [5] and TANGO [6]. The toolkit is capable of receiving and processing JPEG-type video streams.

CONCEPTUAL DESIGN OF Web2cToGo

Web2cToGo is a novel Web2cToolkit service especially designed for mobile devices such as tablet computers or smartphones running iOS, Android or other mobile operation systems. Web2cToGo is a frame which embeds instances of all kinds of Web2cToolkit services.



Figure 1: Web2cToGo client-server architecture.

Web-Desktop

The Web2cToGo service consists of the Web2cToGo mobile app and the corresponding Web2cToGo servlet (Fig. 1). By design an app is a platform-dependent application. However, the Web2cToGo app consists only of a few lines of native code creating a component capable of parsing and executing platform-independent HTML- / CSS-tags and JavaScript code snippets implementing the Web2cToGo Web-Desktop Client.



Figure 2: Web2cToGo Web-Desktop.

Access to the platform-dependent and device specific resources such as the network, the file system or a microphone is provided by the PhoneGap (ver. 1.9.0) library [7] interfacing Android, iOS and other mobile operating systems. The Web2cToGo Web-Desktop client is coded in Java within the Google Web Toolkit framework [8] and compiled to HTML- / CSS-tags and JavaScript code which can be handled by all modern Web browsers or similar components ensuring platform independence.

The Web2cToGo Web-Desktop (Fig. 2) provides an Application Explorer which enables the user to select a standard Web2cToolkit Web-Client application by opening a browser window replacing the Application Explorer window^{*}. This browser window requests and displays the corresponding Web2cToolkit application page hosted by the associated Web server. Each application may consist of multiple pages. At most fifteen applications can be started at a single time and kept open in the Virtual Application Window Set.

Navigation and Application Control by Touch Gestures

The user sees only the so-called Active Application Window and navigates between all started applications or application pages and the Application Explorer window using specific single- or multiple-touch gestures. The navigation bars at the left and lower border of the display are sensitive to the user's Web-Desktop navigation actions, only. Implemented touch-gestures are (Fig. 3) (1) *tap "tile n"* (opens a new application window or displays the selected application window if already opened),

(2) swipe "left \leftrightarrow right" (switches between applications available in the Virtual Application Window Set),

(3) *swipe* "top \leftrightarrow bottom" (switches between application page windows of the active application available in the Virtual Application Window Set),

(4) *tap "up"*, *tap "down"*, *tap "left"*, *tap "right"* (scrolls the active application page window upwards, downwards, towards left, towards right),

(5) *swipe "top* \leftrightarrow *bottom"* and *swipe "left* \leftrightarrow *right"* (zooms the active application page window),

(6) *pinch "open"* (opens the Application Explorer window, and

(7) *pinch "close*" (terminates the active application).



Figure 3: Web2cToGo Web-Desktop navigation with singleand multi-touch gestures (see text).

^{*} The Application Explorer can also launch ordinary Web pages.

Touch events captured by the browser of the Active Application Window are accounted to and handled exclusively by the corresponding Web2cToolkit Web-Client application. They can be used for application control and will not be passed to the Web2cToGo Web-Desktop for navigation purposes.

Navigation and Application Control by Speech

In a simplified view neglecting the dynamics, speech is an acoustic sequence of utterances (words and nonlinguistic fillers) and pauses. Utterances consist of various distinct phonemes. A speech recognition system tries to identify phonemes according to an acoustic model representing the specifics of a language or the sound of an individual speaker. In the next step the system browses a phonetic dictionary listing words and their corresponding decomposition into phonemes in order to find the best match. Optionally, a specific language model constraining the list of words to be recognized and / or defining a grammatical order may guide or simplify the final search.

Web2cToGo uses the Sphinx-4 (ver. 1.0 beta) speech recognition software [9]. Sphinx-4 is an open-source library package entirely written in Java. Audio input to be analysed by Sphinx-4 has to be provided according to the WAVE/WAV audio file format. Web2cToGo speech recognition is based on two acoustic models, US English [10] and German [10].

In Web2cToGo short audio sequences which are only a few seconds long are subsequently recorded by the mobile device and uploaded to the corresponding Web server. If required the uploaded audio file is converted to the proper Sphinx-4 input format using Xuggler (ver. 5.4) [11] providing a Java interface to the FFmpeg (ver. 1.0) [12] media converter. The final audio file is asynchronously analysed by the Sphinx-4 software and the recognized words are finally published by the Web2cToGo servlet to be used by the Web2cToGo Web-Client for navigation or by other Web2cToolkit services for application control purposes.

In our prototype set-up the speech recognition quality turned out to be poor. Less than 20% of the words could be properly recognized. Various reasons for these results have been explored:

(1) An Acer ICONIA TAB A200 [13] tablet running Android (ver. 4.0.3) records the audio files with its internal microphone. The audio recording quality of this device is very limited (dominant humming noise, acoustic artefacts) and is the subject of numerous complaints in the corresponding internet forums.

(2) Android provides audio files with lossy data compression according to the AMR-NB (narrow-band, mono, 8 kHz sampling frequency, 12.2 kbit/s bit rate) or AMR-WB (wide-band, mono, 16 kHz sampling frequency, 6.6 kbit/s bit rate) compression standard. Both formats have to be converted prior to processing by Spinx-4. Dedicated tests with well-known audio test files demonstrate that data compression with AMR-WB and final conversion to WAVE/WAV (mono, 16 kHz sampling rate, 256 kbit/s bit rate) do not spoil the speech

recognition quality, while compression with AMR-NB results in a substantial degradation. †

Both acoustic models (3) (US English: WSJ 8gau 13dCep 16k 40mel 130Hz 6800Hz [10] and German: voxforge_de_sphinx.cd_cont_3000 [10]) suited for 16 kHz data sampling implemented in Web2cToGo provide similar satisfactory speech recognition results, while the acoustic model (US English: WSJ_8gau_13dCep_8k_31mel_200Hz_3500Hz [10]suited for 8 kHz data sampling reduces the recognition quality significantly.

In conclusion, the poor audio recording quality of the mobile device used has been identified as the primary source of degradation in the ultimate speech recognition quality.

PROJECT STATUS AND GOALS

Web2cToGo is an on-going project. The deliverables already implemented include Android 4.x.x Web2cToGo app, platform-independent Web2cToGo Web-Desktop client and servlet, Web2cToGo Web-Desktop navigation by touch gestures and navigation by speech based on the Sphinx-4 compliant acoustic models suited for 16 kHz data sampling.

The goal is to provide additional mobile apps for iOS and Windows RT as well as a Java application suitable for desktop computers. Furthermore, touch gestures dedicated to control Web2cToolkit applications such as history plot zooming in the Web2c Archive Viewer have to be defined and implemented. Finally, handling and reliability of the speech recognition service embedded in Web2cToGo has to be improved and application-specific speech-driven control of the various Web2cToolkit applications has to be implemented.

REFERENCES

- R. Bacher, "Light-Weight Web-based Control Applications with the Web2cToolkit", ICALEPCS'09, Kobe, Japan, October 2009, THP110, p. 889 (2009); http://www.JACoW.org
- [2] Web2cToolkit; http://web2ctoolkit.desy.de
- [3] TINE; http://tine.desy.de
- [4] DOOCS; http://doocs.desy.de
- [5] EPICS; http://www.aps.anl.gov/epics
- [6] TANGO; http://www.tango-controls.org
- [7] PhoneGap; http://phonegap.com
- [8] Google Web Toolkit; https://developers.google.com/webtoolkit
- [9] Sphinx-4; http://cmusphinx.sourceforge.net/sphinx
- [10] Sphinx-4 Acoustic Models; http://sourceforge.net/projects/ cmusphinx/files
- [11] Xuggler; http://www.xuggle.com/xuggler
- [12] FFmpeg; http://ffmpeg.org
- [13] Acer ICONIA TAB A200; http://www.acer

[†] Tests with MPEG-4 audio files (mono, 44.1 kHz sampling rate, 63 kbit/s bit rate) recorded with an iPhone4 running iOS (ver. 6.0) and down-converted to WAVE/WAV files show also promising speech recognition results.

EPICS CHANNEL ACCESS USING WEBSOCKET

A. Uchiyama[#], The Graduate University for Advanced Studies (SOKENDAI), Tsukuba, Japan K. Furukawa, High Energy Accelerator Research Organization (KEK), Tsukuba, Japan Y. Higurashi, RIKEN Nishina Center, Wako, Japan

Abstract

Web technology is useful as a means of widely disseminating accelerator and beam status information. For this purpose. WebOPI was implemented by SNS as a web-based system using Ajax (asynchronous JavaScript and XML) with EPICS [1]. On the other hand, it is often necessary to control the accelerator from different locations as well as the central control room during beam operation and maintenance. However, it is not realistic to replace the GUI-based operator interface (OPI) with a Web-based system using Ajax technology because of interactive performance issue. Therefore, as a nextgeneration OPI over the web using EPICS Channel Access (CA), we developed a client system based on WebSocket, which is a new protocol provided by the Internet Engineering Task Force (IETF) for Web-based systems. WebSocket is a web technology that provides bidirectional, full-duplex communication channels over a single TCP connection. [2] By utilizing Node.js and the WebSocket access library called Socket.IO, a WebSocket server was implemented. Node.js is a server-side JavaScript language built on the Google V8 JavaScript Engine. [3] In order to construct the WebSocket server as an EPICS CA client, an add-on for Node.js was developed in C/C++ using the EPICS CA library, which is included in the EPICS base. As a result, for accelerator operation, Web-based client systems became available not only in the central control room but also with various types of equipment.

INTRODUCTION

One of the advantages of an EPICS-based system is the unified communication protocol between the front-end controller and client systems provided by the CA (Channel Access) protocol. Thus, even when various types of controllers were used as control systems, all the client systems such as the OPI (operator interface) could be developed on the basis of the CA protocol without hardware dependencies. In general, a method that used the CA libraries or display manager supported by EPICS collaboration. such as CAJ/JCA. CA-Python, MEDM/EDM, and CSS (Control System Studio) [4], was adopted for the development of the GUI in the EPICSbased system. On the other hand, the engineers at SPring-8 proposed the development methods for the main OPI using WebSocket and conducted а prototype implementation [5]. The prototype system was constructed using a MADOCA (message and database oriented control architecture)-based system that was developed at SPring-8. It was different from the EPICSbased system. Since the web application with real-time have many advantages, we started to develop WebSocket server corresponding to EPICS CA, and implemented Web applications using WebSocket for the EPICS-based control system. The traditional Web application lacked the interactivity needed to operate some of the hardware for the accelerator control system. By utilizing WebSocket technology, it is possible to solve the problems of Web-based OPI, such as the interactive response.

WEBSOCKET PROTOCOL

WebSocket is a new protocol for achieving the bidirectional communication between a Web server and Web browser. In the beginning, WebSocket was part of HTML5. It was formulated as an RFC6455 by ITEF in Dec. 2011. An overview of the protocol is as follows.

- 1. A Web browser sends a handshake request to the server for connecting to the WebSocket.
- 2. The server returns the handshake response after approval.
- 3. After the establishment of the handshake, the protocol switches to the WebSocket, and then the bi-directional communication occurs between the Web server and Web browser.

WebSocket makes interactive response realizable and thus compensates for the disadvantage that could not be eliminated in traditional HTML. Thus far, various types of Web services in EPICS-based systems have been implemented. However, there were only implemented in Web-based systems, where a quick response and monitoring are unnecessary such as in archive viewers and electric-log systems. This is because it is difficult for a Web-based OPI to realize a real-time response similar to native applications such as EDM/MEDM/CSS, even if Ajax technology is utilized. Since it becomes bidirectional transmission between a server and clients has become possible, the aforementioned problem is solved by WebSocket. In practice, we confirmed sufficient interactive response in a 100 ms cycle using WebSocket on the Web browser, and it had a performance similar to native EPICS applications such as EDM. that of Furthermore, since periodic polling is not necessary like Ajax, it becomes possible to reduce network traffic. WebSocket does have a problem with Internet Explorer 9 (IE9), which is one of the main browsers, because a WebSocket connection is unavailable. However, this problem may be resolved in IE10, which is the next

[#]a-uchi@riken.jp

version, because Microsoft planned it to be compliant with WebSocket. [6]

SERVER SIDE SYSTEM

We utilized Node.js and Socket.IO [7] as a WebSocket library to develop the WebSocket server. Node is is one of the server-side JavaScript languages developed based on the Google V8 JavaScript Engine. As a main feature, Node.js works asynchronously with single-thread processing. Although many human resources may be required for the development of an asynchronous WebSocket server from scratch, Socket.IO solve the aforementioned problem. Node.js 0.6.18 and Socket.IO 0.9.6, which were stable versions of the languages, were utilized for the development. In order to implement the WebSocket server with a CA connection using Node.js, Node is has to call the CA API. Therefore, we developed add-on software to interface CA from Node.js, that is, NodeCA. Since Node.js can call the function developed by C++, NodeCA utilized the CA library provided by the EPICS base. To call CA from Node.js, caGet, caPut, and caMonitor, which are basic functions in EPICS, were prepared. In general, an event-driven caMonitor needs a non-blocking algorithm in the program, although singlethreading Node.js causes the thread to be blocked by the implementation negligently. Consequently, in order to prevent the thread blocking, NodeCA is in waiting the CA event into another thread, and then it send the message queue to the main thread. Figure 1 shows an overview of NodeCA. The software developed by this research is shown in the gray area.





CLIENT SIDE SYSTEM

It is possible to realize a text update like EDM by coding DOM (document object model) with JavaScript on the Web. In addition, flot [8] and jsgause [9], which are jQuery-based JavaScript libraries, were used for the visualization of the accelerator parameters in our system. Many JavaScript libraries, other than those mentioned above, are also available for the visualization of information, such as strip chart. As a result, the client system has the advantage of low development costs since money can be saved multiple steps. Figure 2 shows the whole system chart.

- 1. In the code of client side JavaScript, custom events (caGet, caPut, caMonitor) are sent to the server by using WebSocket protocol
- 2. WebSocket server connect to EPICS IOC via NodeCA.
- 3. It is available to get the accelerator parameters from EPICS IOC.



Figure 2: Whole System Chart.

IMPLEMENTATION

After satisfying the need for interactive access for WebSocket using an EPICS-based system, we attempted to implement NodeCA and the WebSocket server as a Web-based OPI. As a result, the OPI was found to have adequate ability for accelerator operation compared to the traditional EPICS-based OPI. Moreover, we confirmed that almost all the browsers for the iPhone4S/Android are capable of obtaining and displaying numerical values from EPICS IOC via WebSocket. The present software support for browsers and OS is shown in Table 1. In the case of iPhone, it is difficult to install self-produced software without the approval of Apple Store. On the other hand, approval of Apple Store is unnecessary to install the self-produced Web application on an iPhone, although a Web server is required. We consider that the Web application using the WebSocket server and NodeCA have satisfactory performance as an OPI compared with the native applications of the iPhone.

At present, we are implementing it for controlling the 28GHz-ECRIS at RIKEN RIBF on a trial basis. [10] We already verified that we had a good response when monitoring the vacuum of the plasma chamber, controlling the gas-valve, and so on.

Table 1: The Present Software Support

OS	Browsers		
Windows/Linux	Google Chrome 20, Firefox 14,		
	Opera 11		
Android 4.0	Google Chrome 18, Firefox 14		
iOS (iPhone)	Safari 5, Google Chrome 19, Mercury		

DISCUSSION

In 2000, GAN (global area network) was proposed by ICFA (International Committee for Future Accelerators)

as a network for ILC accelerator control. [11] Although GAN aimed at an accelerator-only network with international collaboration, it has not been realized thus far. Currently, a WAN with has sufficient high-speed data communication is already in place around the world for the development of Internet technology. On the other hand, it is difficult to gain direct access from other networks using this WAN with the CA protocol because firewalls are implemented between the accelerator network and gateways. For this reason, we considered the possibility of accelerator control using HTTP technology, which is a standard protocol for the Internet, and WebSocket.

First, we considered the following in order to protect the accelerator networks from outside intrusion. In the case of accessing the accelerator network from the WAN, we ensure the security of WebSocket access by using VPN and authentication with SSL. Additionally, accelerator operators have to completely understand the user attempting to access the networks (login ID and so on) and access route (full domain of Internet providers and so on) before WebSocket control is allowed. Therefore, all of the accessing logs for WebSocket control should be open to the accelerator operators.

Next, we considered instructions for accelerator operation to EPICS IOC from the Web browser via WebSocket. For caPut, which provides instructions to each device or component during accelerator operation, accelerator operators need to know the action and behaviour perfectly. On the other hand, it would not matter if they did not understand some simple instructions in detail such as caGet and caMonitor, which are used to monitor or obtain the numerical values for the accelerator parameters. If some accelerator parameters are changed using the Web application from outside the internal networks without considering the beam tuning, the accelerator condition will be complicated in many cases. Thus, when using caPut, a policy is needed for between the accelerator operators in the control room and a maintainer exerting control remotely over the Web. The following are proposed for this policy.

- 1. Before the maintainer exercises remote control over the Web, they have to call the accelerator operator or send an E-mail.
- 2. The maintainer authenticates users (and passwords) using SSL on the Web. After the authentication, it is possible to monitor the accelerator parameters via WebSocket.
- 3. The accelerator operators check the user ID and the domain or IP address of the Internet provider for the maintainer.
- 4. The maintainer sends a request to the accelerator operators when sending caPut instruction to the EPICS record.
- 5. In response to the request from the maintainer, the accelerator operators make a judgement decision about whether the component in the EPICS record may be controlled using

WebSocket. A response is returned to the maintainer about whether it is possible to operate the component in the EPICS record.

- 6. After receiving this response, the maintainer can not only monitor, but also operate the component via WebSocket.
- 7. If the accelerator operators make a judgement decision to interrupt the operation, the WebSocket operation is discontinued as soon as possible by the accelerator operator. In addition, if a certain period of time passes, the permission for the WebSocket operation will be cancelled by a timeout.

CONCLUSION

We developed a server that makes it possible to connect with EPICS CA by WebSocket. It enables the display of information on the accelerator conditions and make it possible to control it from the Web browser in real time. It also makes it possible to call CA API from Node.js by NodeCA as an add-on to the interface for the CA protocol. We confirmed that we could control and monitor one part of the accelerator parameters as well as the traditional EPICS-based application. Additionally, the developed Web-based OPI runs not only on the main PCbased browsers, but also on almost all of the browsers for Android and iPhone4S. We believe that this technology will be useful as a means of accelerator operation using Internet access, even if it has some problems of security. In order to resolve these security problem for the operation from a WAN, it is necessary to lay everything out on the table to develop an access policy. In the future, we will develop the NodeCA and WebSocket server for the latest version of Node.js, which will be released as open-source software.

REFERENCES

- [1] K. U. Kasemir, et al., Proceedings of ICALPECS 2011, Grenoble, France, 2011, THBHAUST01.
- [2] I. Fette and A. Melnikov, The WebSocket Protocol, IETF HyBi Working Group. 2011.
- [3] http://nodejs.org/
- [4] X. Chen, et al., Proceedings of 2011 Particle Accelerator Conference, New York, NY, USA, 2011, WEOBN3.
- [5] Y. Furukawa, et al., Proceedings of ICALPECS 2011, Grenoble, France, 2011, WEMAU010.
- [6] http://msdn.microsoft.com/en-us/library/ie/
- [7] http://socket.io/
- [8] http://code.google.com/p/flot/
- [9] http://code.google.com/p/jsgauge/
- [10] A. Uchiyama, et al., Proceedings of ECRIS 2012, Sydney, Australia, 2012, TUPP12
- [11] "A Global Accelerator Network: ICFA Task Force Reports", http://www.fnol.gov/dimeterato/iofo/iofo/tofo.cfg.

http://www.fnal.gov/directorate/icfa/icfa_tforce_repo rts.html, Dec. 2001

Copyright © 2012 by the respective authors

Ot BASED GUI SYSTEM FOR EPICS CONTROL SYSTEMS*

A. Rhyder[#], R. N. Fernandes, A. Starritt, Australian Synchrotron, Clayton 3168, Australia

Abstract

The Qt-based GUI system developed at the Australian Synchrotron for use on EPICS control systems has recently been enhanced to including support for imaging, plotting, user login, logging and configuration recipes. Plans are also being made to broaden its appeal within the wider EPICS community by expanding the range of development options and adding support for EPICS V4. Current features include graphical and non-graphical application development as well as simple "code-free" GUI design. Additional features will allow developers to let the GUI system handle its own data using Ot-based EPICS-aware classes or, as an alternative, use other control systems data such as PSI's CAFE.

INTRODUCTION

The Qt-based GUI system, known as QE framework or simply QE, is a layered framework based on C++ and Qt for accessing EPICS data using Channel Access (CA). It is used on several beamlines at the Australian Synchrotron. Channel Access is one of the core components of an EPICS system allowing a CA client application to access control system data which may be located on different hosts throughout a network [1]. While CA is the default means to access EPICS data, its use is not trivial. A significant understanding on how this component works is required to read or write data. The complexity of setting up and terminating CA requests leaves room for error. The QE framework handles much of this complexity including initiating and managing a channel. Applications using QE can interact with Channel Access using Qt-based classes and data types. CA updates are delivered using Qt signals and slots mechanism. It provides access to EPICS data at several levels including programmatic reading and writing of data, EPICS-aware widgets such as push buttons, sliders and text widgets for developing GUI applications [2]. When these plugins are used within Qt Designer, GUIs interacting with EPICS can quickly be assembled without the need for any code development by meaning of simple drag & drop operations.

FRAMEWORK OVERVIEW

The QE framework allows access to Channel Access graphically through Qt-based widgets or through Qt friendly data objects. The data objects manage Channel Access connections and provide a simple, comprehensive, object oriented view of the CA data and related attributes. The QE graphical widgets and supporting QE data objects

#andrew.rhyder@synchrotron.org.au

form a hierarchy of classes that is open at all levels to the developer. Appropriate use of these classes is shown in Table 1. Also, the framework includes an application, QEGui, which is available to present control centric Qt user interface files.

Table 1: QE framework classes and their functionality.

Classes	Functionality
Data objects: QEObject QEInteger QEString QEFloating	Provides a convenient object oriented way to access the CA library. Hides CA specific complexity and provides read/write conversions to and from EPICS data types where required. Adds Qt features such as signals and slots to handle data updates. These classes are used programmatically.
Standard widgets: QEComboBox QEForm QEFrame QEGroupBox QELabel QELineEdit QEPushButton QERadioButton QESlider QESpinBox	Graphical objects that allow users to interact with CA data using simple and familiar graphical controls. These classes may be used programmatically or within Qt Designer.
Extended widgets: QEAnalogProgressBar QEBitStatus QEConfiguredLayout QEFileBrowser QEImage QELink QELog QELogin QEPeriodic QEPoVProperties QERecipe QEScript QEShape QEStripChart QESubstitutedLabel	Graphical objects that allow users to view CA data through a broad range of display models and support sophisticated control system user interface design. These classes may be used programmatically or within Qt Designer.

^{*}Work supported by the Australian Synchrotron

USAGE PARADIGMS

Thanks to its architecture, the QE framework may be used by different people and in different ways. The simplest is as a "code free" development environment where the user builds GUIs without the need to program by means of dragging & dropping EPICS-aware widgets into a graphical user interface. Each widget can be then configured through a set of properties and inter-connected through Qt signals to exchange data with other widgets. All these can be done within Qt Designer in a userfriendly fashion (see Figure 1). A single application QEGui is used to present a set of user interface files as an integrated control system suite.



Figure 1: QE framework within Qt Designer.

Another way of building GUIs is to use QE programmatically (see Figure 2). This effectively allows full control of the framework and even extends it to solve specific issues through C++ inheritance mechanism. It also allows the creation of non-graphical applications – e.g. servers – interacting with CA. A complete rewrite of the API reference documentation (generated by Doxygen) is currently being done. It will help the developers to better understand and use QE programmatically.



Figure 2: Using the QE framework programmatically.

The development of the QE framework has been intense in recent months and more developments are expected for the near future. These will produce a framework more stable/mature capable of coping with ever changing requirements. Some key developments were already identified namely:

- Testing. A suite of test units will be implemented to ensure that QE complies with the requirements. It will guarantee that nothing breaks if the development effort intensifies and goes beyond the Australian Synchrotron.
- Guidelines. The specification of guidelines is crucial as the QE framework goes towards an international collaboration development. These guidelines will guarantee that QE maintains its architectural and implementation coherence even if several people are working on it in a scattered fashion. Some of these guidelines will be enforced through automated tests.
- Binding. More and more, the Python programming language is becoming the "lingua franca" of the scientific community. Plans are being made to export the QE framework into this language through bindings. After initial analysis of binding generators, SIP is the chosen one to accomplish such task.
- Layering. Further abstraction/separation between data and graphical (widgets) classes. This will allow other control systems data to be supported by the framework without modifying the graphical layer.
- EPICS V4. New concepts were introduced in the latest version of EPICS such as structured data or a new channel access protocol called pvAccess [3]. QE will be updated to support these.

CONCLUSION

QE is a framework which enables the creation of Qtbased GUIs interacting with EPICS data in a "code-free" fashion or programmatically via C++. This allows GUIs to be built by people without programming skills or by people that need more control. Future developments will allow the use of the framework in Python by the scientific community. Special concern will be made to support EPICS version 4 and its new features.

REFERENCES

- P. Stanley, "Channel Access Client Tutorial", Los Alamos National Laboratory, November 1997; http://lansce.lanl.gov/EPICSdata/ca/client/caX5Ftutor-4.html
- [2] A. Rhyder et al., "Qt EPICS Development Framework", PCaPAC 2010, Saskatoon, October 2010.
- [3] EPICS v4 Working Group, "pvAccess Protocol Specification", September 2012; http://epicspvdata.sourceforge.net/pvAccess_Protocol_Specification.ht ml

Copyright © 2012 by the respective authors

DATA LOGGING SYSTEM UPGRADE FOR INDUS ACCELERATOR

R. Mishra[#], B. N. Merh, R. K. Agrawal, P. Fatnani, C.P. Navathe, RRCAT, Indore, India S. Pal, VECC, Kolkata, India

Abstract

An accelerator has various subsystems like Magnet Power Supply, Beam Diagnostics and Vacuum etc. which are required to work in a stable manner to ensure required machine performance. Logging of system parameters at a faster rate plays a crucial role in analysing and understanding machine behaviour. Logging all the machine parameters consistently at the rate of typically more than 1 Hz has been the aim of a recent data logging system upgrade.

Nearly ten thousand parameters are being logged at varying intervals of one second to one minute in Indus accelerator complex. The present logging scheme is augmented to log all these parameters at a rate equal to or more than 1 Hz. The database schema is designed according to the data type of the parameter. The data is distributed into historical table and intermediate table which comprises of recent data. Machine control applications read the parameter values from the control system and store them into the text files of finite time duration for each sub-system. The logging application of each sub-system passes these text files to database for bulk insertion. The detail design of database, logging scheme and its architecture is presented in the paper.

INTRODUCTION

Indus-1 and Indus-2 Synchrotron Radiation Sources (SRS) housed in Indus complex are cyclic particle accelerators that accelerate electrons up to 450 MeV and 2.5 GeV respectively [1]. These are supported by Microtron injector, Booster Synchrotron and various transport lines. In a multi accelerator complex like this, involving various complex subsystems, an effective system for comprehensive parameter data logging is considered essential.

Ideally, data logging system should contain all the information necessary to provide an accurate representation of the state of system parameters at any given time. The existing data logging system acquires and stores all the subsystem parameters at varying time interval of one second to one minute. Although this strategy has worked satisfactorily but this data logging scheme does not scale well as the number of control parameters grow and it begins to strain network and storage capabilities. This becomes a bottleneck in efficient operational diagnosis of all sub-systems in Indus accelerator. So an up gradation is aimed to design a data consistent can provide a platform to log all the machine parameters at the same rate at which all the parameters are acquired from the field i.e. at 1Hz.

#rohitmishra@ rrcat.gov.in

Hence, a new data logging approach of gathering the time series data in text files in the form of strings with delimiters along with the timestamp is adopted [2]. Besides this some major modification at software level has been introduced to achieve fast data logging of all the parameters at 1 Hz.

HARDWARE AND SOFTWARE ARCHITECTURE



Figure 1: High level layout of data logging architecture.

The data collection process of Indus SRS machine is based on three tier architecture. The top layer architecture is shown in Figure 1. In the Indus control systems, middleware collects and processes the data from the field devices and sends it to the Indus control server based on the PVSS [3] industrial SCADA system. The PVSS control server continously puts the data into text files with the timestamp for time span of five minutes and copies the text files (data files) to Java application server over TCP/IP. This process is repeated with time for each subsystem. It also deletes the previous text files only on receiving the successful acknowledgement of copy operation from Java server. Hence the control application running on PVSS control server also prevents the data loss in critical situations of logging failure by retaining the data in text files.

For each sub-system of Indus-2, Java application interface exists between PVSS control server and database server. The Java application peridocally checks for recently received data file from the PVSS server and pass on the data file name to database server through a stored procedure call. These stored procedures after recieving the the data file information, perform the bulk insert operation into corresponding subsystem database

table. The Java application also manage temporal synchronization and serve as watchdog for any application failure.

	-
Hardware Resource	Specification
Database Server	HP Proliant DL 360 G7
Processor	Intel(R) Xeon(R) CPU E5630 @2.52 GHz(dual processor)
Disk	6G DP 10K SAS,
	4x146 GB
Memory	8.00 GB

Table 1: Hardware Specifications

Table 2: Software

Software Category	Description
System Software	Windows Server 2008 R2®
Application Software	MSSQL Server 2008 R2® (Enterprise edition)
Application Software	Java(TM) SE Development Kit 6 Update 26 (Oracle Product version:1.6.0.260)
Application Software	PVSS 3.9® (ETM professional Control GmbH Product version:3.9)

DATABASE DESIGN

Separate databases are created for each subsystem of Indus-1 and Indus-2 and database schema pattern followed in this upgraded data logging scheme is based on '*data table per data type*' approach [4] i.e. same data-type parameters values are stored in a single table. The main advantage of this approach is its scalibility. The collected data is distributed into two categories of tables: main table and intermediate table on the basis of timestamp associated with the particular data .

Main table holds all the past available data except data residing in intermediate table. The intermediate table holds the most recent available data (in case of Indus database, intermediate table holds last one hour available data). Table partitioning feature [5], supported by MSSQL server 2008 R2 (**R**), has been implemented on the intermediate table for achieving better performance in data loading operation.

There is one main configuration table created in each subsystem database. This table comprises of metadata or configuration information of each parameter associated with the subsystem. Besides this, there are two staging tables in each subsystem database which support the efficient flow of data at the time of switching of table partition using sliding window scenario concept [5].

Data Flow Cycle

There are certain prerequisites that are essential for realizing data flow for implementing sliding window model of table partitioning [5].To comply with the prerequisites, the schema design of tables are kept same among staging, intermediate and main tables i.e. table schema for storing analog data type parameters and table schema for storing status data type parameters are same in all three tables categories.

Figure 2 shows the data flow cycle for analog data type parameters inside SQL server database. A similar data flow cycle is followed for table carrying status parameters.





Diagnostics and Error Handling

In order to diagnose and monitor the data logging system, a status panel is designed in PVSS. The Java application edits the status information in the log file at every five minute logging interval. The control scripts at PVSS server reads the information in the log file and subsequently reflect any undesired behaviour in the whole logging system on the status panel. Error is reported if required data files are not found or if bulk insertion operation is not successful.

RESULT AND CONCLUSION

A new upgraded data logging system has been designed which is tested at the experimental server machine. It has been demostrated that the scheme has successfully logged approximately 1 GB of data per day. Almost 8000 parameters are logged at the rate of one hertz without any loss of data at any time instant. Thus this logging system is more suitable in terms of performance & scalibility and it is being deployed in Indus control system architecture.

SUMMARY AND FUTURE PLAN

In comparison to previous data logging system of logging system has enhanced the Indus, this new reliability and data availability of the time critical data of Indus systems. It improves the data logging rate by the use of advanced features like table partitioning and modifying the internal database schema.

Although the development phase of the data logging system has been accomplished as per our requirements but there still exists scope of improvements in the database performance from database administrative and maintenance aspect.

REFERENCES

- [1] R. K. Agrawal et al., "Software Scenario for Control System of Indus-2", 10th ICALEPCS Int. Conf. On Accelerator & Large Expt. Physics Control Systems. Geneva, 10 - 14 Oct 2005, WE3B.3-70 (2005).
- [2] E. Goman, S. Karnaev, O. Plotnikova, E. Simonov, "The database of the VEPP-4 Accelerating facility parameters, Proceedings of PCaPAC08, Ljubljana, Slovenia.
- [3] PVSS-II is a SCADA package from ETM, Austria.
- [4] R. Billen, C. Roderick, "The LHC Logging Service," Paper presented at the UKOUG Conference 2006 14-17 November 2006, Birmingham, UK.
- [5] "Partitioned Tables and Indexes in SQL Server 2005" http://msdn.microsoft.com/enus/library/ms190787(v=SQL. 90).aspx

CONTROL SYSTEM STUDIO ARCHIVER WITH PostgreSQL BACK-END: OPTIMIZING PERFORMANCE AND RELIABILITY FOR A PRODUCTION ENVIRONMENT*

M. Konrad[†], C. Burandt, J. Enders, N. Pietralla TU Darmstadt, Darmstadt, Germany

Abstract

Archiving systems based on relational databases provide a higher flexibility with regard to data retrieval and analysis than the traditional EPICS Channel Archiver. On the other hand they can suffer from poor performance compared to the Channel Archiver for simple linear data retrieval operations. However, careful tuning of the database management system's configuration can lead to major performance improvements. Special care must be taken to ensure data integrity following power outages or hardware failures.

This contribution describes the hardware and software configuration of an archiving system used in the production environment at the S-DALINAC. It covers performance and reliability aspects of the hardware as well as tuning of the Linux operating system and the PostgreSQL server.

INTRODUCTION

Archiving systems that collect and store data from an accelerator control system play an important role in modern control system environments. They can e.g. help to identify broken hardware, decreasing performance of components or wrong operation of systems by an operator. Thus archiving systems with high availability are demanded. Since wrong or inconsistent data from an archiving system can lead to wrong decisions by the operators it is also important to ensure data integrity. On the other hand data from the archive should be (read-only) accessible from many different applications ranging from simple spreadsheet applications to mathematical software for sophisticated data analysis. These requirements make relational databases (RDBs) a reasonable choice as a storage back-end.

The archiving system of the S-DALINAC is based on the Control System Studio (CSS) Archive Engine. This service collects data from control system channels and writes it to different RDB back-ends (MySQL, Oracle or PostgreSQL). PostgreSQL 9.1 running on Debian Linux has been chosen as a back-end because it is a powerful opensource RDB solution that comes without license fees.

Archiving systems usually have to manage a growing amount of data making read and write performance an important issue. In the following we describe how to tune an archiving system for maximum performance while ensuring data integrity.

Control Databases

DISK PERFORMANCE AND RELIABILITY

Disk access is the most important factor for the performance of a database system that deals with much more data than can be stored in main memory. While modern disk drives can deliver data rates of more than 100 MB/s for sequential reads and writes they can be very slow when they have to deal with random access patterns. Heavy random access workload can occur if data is read from an archive by multiple clients using indexes. Thus database disks have to be optimized for high random access performance.

Write Caching

To improve access time all recent hard disk drives use integrated write-back caches and techniques to speed up disk access by optimizing the order in which read and write commands are executed to reduce the amount of drive-head movement. To insure data integrity database management systems (DBMS) flush operating system write caches after each database commit to make sure all relevant data has been stored on disk before marking the transaction as complete. Write-back caches on the disk drive itself are not flushed and thereby can lead to data loss and corruption of the database files in case of a loss of power. To avoid this, disk write caches have to be disabled for a database server.

Unfortunately switching off this cache severely reduces write performance. Part of this performance loss can be regained by using an appropriate operating system I/O scheduler that sorts the commands before sending them to the disk. But frequent flushing limits the possibilities for optimizations. A solution that provides better performance is using a hardware RAID controller with a battery backup unit. These controllers contain a non-volatile write-back cache that is powered by a battery in case of power outages. Data still in this cache when a loss of power occurs is written to disk after power resumes. Note that disk write caches often have to be disabled explicitly using the configuration tool of the RAID controller.

In contrast to hard disks, solid-state drives provide very high random access performance, but usually their writeback cache is volatile and cannot be disabled making them a bad choice if data integrity is a concern. Write caching is also an issue if a DBMS runs in a virtual machine because reliably flushing data to disk is impossible with most of today's virtualization products. Thus a physical machine is required for the RDB back-end of an archiving system.

^{*} Work supported by DFG through CRC 634

[†] konrad@ikp.tu-darmstadt.de

Table 1: Characteristics of RAIDs Consisting of N Disks with Different Organization (Assuming Pairs of Two Drives for RAID 10)

RAID level	Space efficiency (disks)	Fault tolerance (disks)	I/O in read	mpact write
0	N	0	1	1
1	1	N-1	1	N
5	N-1	1	1	4
6	N-2	2	1	6
10	N/2	1 per pair	1	2

The CSS Archive Engine on the other hand can safely be run on a virtual machine.

RAID Organization

Redundant arrays of independent disks (RAID) are a way to improve disk performance and reliability [1]. Table 1 gives an overview of the characteristics of the most common RAID levels. RAID 0 increases performance by striping data over multiple disks. It is unsuited for most archiving purposes because data is lost if a single disk fails. RAID 1 mirrors data to multiple disks to improve reliability. The size of the array is thereby limited by the size of a single disk which is insufficient for big archives. RAID levels 5 and 6 store additional parity information to make the array tolerant against failures of one or two disks, respectively, while still providing high space efficiency. The disadvantage of these RAID levels is that modifying a single block leads to 4 or 6 I/O operations, respectively. This can significantly reduce performance of archiving systems that constantly archive thousands of channels at high data rates. RAID 10 stripes data over pairs of mirrored disks and thereby provides high reliability along with good read and write performance, even if a drive has failed. Therefore RAID 10 has been chosen for the S-DALINAC's archiving system. If space efficiency is more important than write speed (e.g. for long-term archiving systems) RAID 6 might also be an option. Regardless of the RAID level, modern RAID controllers distribute random reads over all N disks resulting in an N times higher random-read performance than a single disk can deliver.

The performance of a RAID can be improved by using faster spinning disks or a higher number of spindles. For the S-DALINAC archiving system a higher number of slower disks (10,000 rpm) turned out to be more economic. Direct-attached storage has been favored over networkattached storage for its lower latency. For systems that need more storage than can be directly attached to a single machine storage area network technology might be an option.

Separate RAID volumes are used for the database itself, the write ahead log (WAL) of the database, and the operating system (see Table 2). This makes sure that operating system activity does not slow down database access. Separate disks are used for the WAL because this file is written

Table 2: Disk Organization of the S-DALINAC RDB Server

Volume	RAID level	Number of disks
operating system	1	2
write ahead log	1	2
database	10	30
hot spare		2

very heavily while data is inserted into the database. Since the write pattern of the WAL is sequential this almost completely avoids disk seek times. In addition to that separate volumes simplify identification of bottlenecks as well as continuous load monitoring.

Operating system configuration can also have a severe impact on the performance of an archiving back-end. On machines using a hardware RAID the noop operating system I/O scheduler should be used to avoid false optimization. In addition to that increasing read-ahead size can be necessary to exploit full sequential read data rates.

MEMORY CONFIGURATION

The database back-end can answer queries a lot faster if it does not need to read the data from disk. With 128 GB the size of the system main memory has been chosen to be large enough to hold the data of the last days including the relevant indexes.

PostgreSQL spawns a dedicated process for each database connection. All these processes perform read and write operations against a common memory region called buffer cache. Improper configuration of this memory can severely reduce read and write performance. Note that this parameter is configured for maximum compatibility instead of maximum performance by default. For optimal performance we had to increase the buffer cache size by three orders of magnitude to roughly 25% of the main memory. This still leaves the operating system enough RAM for caching reads. On Linux systems it is necessary to increase the maximum shared memory segment size of the kernel accordingly.

PARTITIONING

If the size of the sample table grows much larger than physical memory and even its index stops fitting into memory query times can escalate. One way to improve performance is to split data into several partitions that each fit into main memory. At the S-DALINAC most database queries ask for data from the last three or four days making partitioning over time with a partition size of a week a reasonable choice. When executing queries the DBMS can skip partitions that are outside the requested range. For most queries only one or two partitions have to be considered.

In contrast to enterprise databases like Oracle or MS SQL, PostgreSQL does not provide built-in functions for partitioning. Using PostgreSQL's extensive server-side

Table 3:	Write	Rates	Obtained	with	Different	Configura-
tion of th	e Arch	iver D	atabase Ta	ıble		

Foreign key constraints	Partitioning	Rows/s
no	no	18289
yes	no	6075
yes	yes	3865

Table 4.	Hardware	of the	S-DAI	INAC	PostareS(א ור	erve
1 aute 4.	панимане	or the	S-DAL	JINAU.	rusigiesu	ノレン	

Main-board	Supermicro X9DRi-F
CPU	2×Intel Xeon E5-2643 @3.3 GHz
Main memory	128 GB DDR3 registered ECC
RAID controller	Adaptec 6805 SAS2
Disks	36×Toshiba MBF230LRC 300 GB

programming features a partitioning solution tailored to the particular needs of the CSS archiver has been developed. It uses table inheritance to combine the data of weekly subtables into one table. A trigger function redirects INSERTs to the appropriate partition. New partitions are added automatically. The partitioning feature only affects the database side of the archiver and has been included into the Control System Studio distribution.

While partitioning can significantly speed up read access it introduces checking of additional constraints during INSERTs. This results in higher CPU load and can limit maximum write rates.

PERFORMANCE MEASUREMENTS

Write Performance

Before the archiving system has been put into operation, write performance has been measured using the RDBArchiveWriterTest.testWriteSpeedDouble() test included in the CSS Archive Engine code. This test is very close to real archiving load because it uses the same write scheme and the same code on the client side. It has been used to verify the success of the tuning steps described in this paper. Typical write rates obtained with this test are presented in Table 3. An overview of the hardware used for this benchmark is given in Table 4.

Write performance strongly depends on the configuration of the database. Adding foreign keys between tables to ensure referential integrity forces the DBMS to check each row against all foreign keys before an insert operation can be performed.

Performance of database writes also strongly depends on the command that is used to write the data. The current implementation of CSS Archive Engine improves performance by submitting INSERTs to the database server in batches before committing the data. Benchmarks implemented in Perl confirm this performance gain but they also show that other write techniques can provide even better performance. Figure 1 clearly shows that much higher write rates can be obtained by using multi-row INSERTs or



Figure 1: Write performance obtained with different write schemes and batch sizes.

the PostgreSQL-specific COPY command. Although performance might depend on the library used for database access, results suggest that performance might be improved in the future by using more efficient write commands.

Read Performance

Benchmarking read performance is more complex than benchmarking write access since synthetic benchmarks can lead to unrealistic caching of relevant data. In contrast to the write case multiple clients can issue queries at the same time. Up to now read performance has only been determined by measuring execution times of single queries. These results confirm that partitioning improves read performance. A more realistic benchmark is under development.

SUMMARY

A battery backed up write cache as well as careful configuration of the PostgreSQL back-end are important to achieve high performance while at the same time ensuring data integrity. Write performance can be improved by removing foreign key constraints while at the same time losing referential integrity checks. Another way to improve performance in the future might be to use more efficient write patterns like multi-row INSERTs or the COPY command. Read performance can be significantly improved by partitioning the sample table. As soon as solid-state drives are reliable enough they can also help to further improve performance.

ACKNOWLEDGEMENTS

We thank Kay Kasemir (ORNL) and Lana Abadie (ITER) for fruitful discussions and their valuable help in understanding the implementation of the Archive Engine.

REFERENCES

 D. A. Patterson et. al., "A case for redundant arrays of inexpensive disks (RAID)". Proceedings of SIGMOD 1988, Chicago, IL, USA, p. 109–116.

FAST DATA ACQUISITION SYSTEM FOR BOOSTER SUPPLIES READBACK

K. Saifee, A. Chauhan, P. Gothwal, P. Fatnani, C.P. Navathe, RRCAT, Indore, India

Abstract

Booster synchrotron at RRCAT is used to inject electron beam in Synchrotron Radiation Sources - Indus-1 & Indus-2. Booster gets 20 MeV beam from Microtron, ramps up its energy to 450/550 MeV which is then extracted for injection in Indus-1/Indus-2. The Ramping cycle repeats every second. For this, various magnet power supplies are fed reference voltage & current waveforms synchronously and accordingly they feed the magnets with current for ~800 msec. A system was required to synchronously capture data of all power supplies to on cycle to cycle basis. Global machine data monitoring system polling data at 1 Hz cannot acquire sufficient points to do this. So a VME and PC based system has been developed for parallel and fast capture of data from 13 such power supplies.

INTRODUCTION

There are 13 power supplies in the booster which participate in the ramping process to increase electron beam energy from 20 MeV to 450/550 MeV. Successful operation of booster depends upon the synchronous, reliable and reproducible performance of these power supplies. The development of fast, synchronous data acquisition system provides simultaneously captured waveform data sets for all the power supplies. This data provides quantitative measurement of power supply performance regarding the reproducibility of ramp waveforms and tracking between them. The hardware has been developed on VME platform consisting of CPU (68040), 12 bit ADC cards and control card. User can select- permit to capture, start delay, samples and time interval between samples. Advantages are - Isolated, simultaneous capturing on 13-channels, capturing synchronized with an event and selectable capturing-rate and samples. The CPU board on the VME runs RTOS OS-9 and control program on PC is developed using LabVIEW.

SCHEME OF THE DATA ACQUISITION **SYSTEM**

The implemented scheme has two-tier architecture. The first layer has a PC and the second layer has a VME station. The two communicate over Ethernet using TCP/IP sockets. A control program in LabVIEW® runs on the Arc. Ine VME-station has a CPU board, 1 and a Control board as shown in Figure 1. PC. The VME-station has a CPU board, 13 ADC boards



Figure 1: Scheme of Data Acquisition System.

The ADC board has a 12-bit sampling ADC with conversion rate of 100 KSPS and analog input range of 0-10 V. There is also a FIFO memory of size 8 K words on each ADC board. The board converts the input analog signal and stores the digitized data in this memory in response to an external trigger. The data that is captured in FIFO is read by the CPU.

There is a Clock and Mode 'Control Board' in the VME station. The CPU writes/ reads on this board to set the various modes and parameters related to the data capturing scheme. The ranges/values of the various parameters are:

- Permit capturing: ON / OFF
- Sampling Delay: 20 µsec to 200 msec
- Sampling interval: 20 µsec to 10 msec
- No. of Samples: 256, 512, 1K, 2K, 4K and 8K words.

The CPU board is Motorola® MVME-162 board based on 68040 Microprocessor with Real Time Operating System OS-9® ported on it.

OPERATION OF THE DATA ACQUISITION SYSTEM

The 'Start of Ramp' signal is connected to the input of 'Control Board'. The card responds to this input if the 'Permit Capturing' parameter is set to ON. In response to this input and as per the parameters set, the card outputs the clock pulses. These clock pulses are given to a 'Clock Receiver and Distributor Board' that duplicates this input clock on its 13 output channels. Each output channel is connected to one ADC card and this clock acts as the trigger for the ADC card. The parameter 'Sampling interval' decides the frequency of the clock and 'Samples to be captured' decides the number of clock pulses.

The OS-9 application program on the CPU Board communicates with control program running on PC. It receives various commands and acts on them. It sets the parameters on the 'Control Board'. It reads the FIFOs on ADC boards and sends back the captured data. This program accesses various VME cards using the OS-9 Device Drivers.

The control program running on the on the PC has following functionalities:

- Operation mode selection.
- Get data and represent it in graphical form.
- Selectable plotting of multiple graphs.
- Logging of data with comments from operator

Presentation of the captured ramp data waveforms in graphical form is shown in Figure 2.



Figure 2: Graphical representation of captured ramp data waveforms.

RESULTS AND CONCLUSION

The Fast data acquisition has been deployed in the Booster hall and has been interfaced with the power supplies. The data captured form the system is often analyzed to investigate the operation of the Booster system. Future plan includes extending the FIFO memory on the ADC in order to capture more samples at higher sampling rates, comparing captured waveform with the reference and automatic report generation for the captured waveform.

EMBEDDED CAMAC CONTROLLER: HARDWARE/SOFTWARE CO-OPTIMIZATION FOR HIGH THROUGHPUT

P. M. Nair, K. Jha, P. Sridharan, A. Behere, M.P. Diwakar, C.K. Pithawa, Electronics Division, BARC, Mumbai, India

Abstract

Advances in technology have resulted in availability of low-power, low form-factor embedded PC based modules. The Embedded CAMAC Controller (ECC) is designed with ETX (Embedded Technology eXtended) standard Single Board Computer (SBC) having PC architecture with Ethernet connectivity. The paper highlights the software and hardware design optimizations to meet high throughput requirements of multi-parameter experiments and scan mode accelerator control applications. The QNX based software is designed for high throughput by adopting design strategies like multi-threaded architecture, interrupt-driven data transfer, buffer pool for burst data, zero memory copy, lockless primitives and batched event data transfer to host. The data buffer and all control logic for CAMAC cycle sequencing for LIST mode is implemented entirely in hardware in Field Programmable Gate Array (FPGA). Through this design, sustained throughput of 1.5MBps has been achieved. Also, the host connectivity through Ethernet link enables support for multi-crate configuration, thus providing scalability. The ECC has been installed for accelerator control at FOTIA BARC. Pelletron and LINAC-Pelletron. TIFR and for multiparameter experiments at NPD, BARC.

INTRODUCTION

PC compatible architectures have become a popular choice for embedded systems because of easy availability of low powered, low form-factor embedded PC based modules. Also, a large base of knowledge and resources exist for these architectures. An Embedded PC meets special needs of mechanical design, power consumption, product lifetime, customized software and professional support. All these readily available features of Embedded PC-based technology coupled with optimum hardware and software design allowed to develop an Embedded CAMAC Controller (ECC) for high throughput requirements of multi-parameter experiments. The paper discusses various design optimizations in the hardware and software of ECC.

EMBEDDED CAMAC CONTROLLER

Embedded CAMAC Controller [1] has been designed to cater to the needs of process control systems and high throughput of large nuclear physics experiments. Control applications need more number of physically distributed crates with regular scanning of all the parameters, the control being with a centralized computer, whereas nuclear physics experiments need a high throughput with a large number of parameters in one or more crates.

The Single Board Computer (SBC) of ECC interfaces with PCI-CAMAC Interface logic through PCI controller (see Fig. 1). The ECC allows interaction with remote host by means of standard Ethernet services, such as TCP socket based communication protocol. The processor runs a version of QNX optimized for low memory footprint. The FPGA holds List buffer, Data buffer and all control logic for CAMAC cycle sequencing. Three modes of operation have been supported:

- *Single Cycle mode:* In this mode, single ECC command is executed in every cycle.
- *Scan mode:* In scan mode of operation, a list of CAMAC commands is sent to the ECC and is executed at regular intervals, and is suitable for control applications.
- *List mode:* The list mode is optimized for multiparameter experiments. In this mode a list of CAMAC commands is stored in Field Programmable Gate Array (FPGA) which is executed on every Look At Me (LAM) from signal acquisition module.

SALIENT FEATURES: HARDWARE

- List sequencing mode with 1.1 µs CAMAC cycle for multi-parameter experiments
- Scan mode with a period of 50msecs for control applications
- Trigger source: LAM / External event / Timer
- Ethernet connectivity for remote and multi-crate configurations.
- Built-in test features with CAMAC data way display

List Mode Optimization

ECC operates in List mode for multi-parameter experiments. There are three lists of NAF commands in multi-parameter mode of acquisition; EVENT_LIST, INI_LIST and SCALER_LIST. The INI_LIST is executed once at start of acquisition to initialize the ECC. EVENT_LIST of up to 256 CAMAC commands is executed upon generation of LAM and data is stored in FIFO buffer. In List mode processing, once the setup is complete and acquisition is started, the host does not intervene other than to stop the acquisition. In this mode, all acquired event data are stored in communication buffers and are send to host PC in batches for achieving better throughput. The size of communication buffer is user programmable so that throughput can be optimised with reference to the number of parameters at experiment time. At the end of every batch of event data which are to be send to the host, SCALER LIST is executed. The acquired batch of events and the scalar counts are sent to the host PC.

The list processing is completely implemented in FPGA hardware to achieve optimum CAMAC throughput of 1.1µs with only 100ns overhead per readout. Optimization in hardware is done by maintaining two event buffers in FPGA. When one buffer is full, event data is acquired in the other buffer and first data buffer is transferred to PC memory.

SALIENT FEATURES: SOFTWARE

The application software for ECC has been designed to achieve high throughput for a burst nature of data for nuclear physics experiments and the stringent periodic scan requirements of control applications. ECC application runs embedded ONX based software with a low memory footprint designed for high throughput by adopting design strategies like multi-threaded architecture, interrupt-driven data transfer, buffer pool for burst data, zero memory copy, lockless primitives and batched event data transfer to host.

Architecture

Multi-threaded programming architecture was selected to achieve maximum CPU utilization and to make the application responsive to the user. Acquisition of experiment data and transmission of data to host are done concurrently. Through proper assignment of thread priority, data transmission to the host is interspersed while waiting for CAMAC readout. Acquisition is assigned highest priority and sufficient buffers with a provision to expand/contract the buffer pool have been provided to help absorb event burst.

Communication module for data transmission to host is designed based on Reactor pattern which allowed simple coarse-grain concurrency without adding complexity of multiple threads to the system. This pattern also allowed prompt data transmission of List mode data to host by use of common de-multiplexer (select) to notify communication events immediately while remaining responsive to user commands.

Object Pool

Event data generated during nuclear physics experiments in bursts with an average rate of about 1K events per second. Hence sufficient buffering is provided to handle this burst data. Application, on Start-up, preallocates all anticipated memory requirements on memory pool. Allocation of memory on pool reduced the need for dynamic memory allocation during application run thereby optimizing on memory allocation delays. The buffer pool is expanded as needed to maintain the specified number of free buffers in the pool. The buffer pool contracts as necessary, to prevent the pool from consuming system resources permanently as the result of usage peaks. Application software provides a feature to batch the acquired event data by providing adequate communication This buffers. strategy reduces communication overhead. Number of events that can be batched is programmable from the host PC.

Interrupt-Driven Data Transfer

To achieve maximum throughput in list mode, software has implemented Interrupt-Driven Data transfer. This offloaded the CPU and resulted in low overhead, low latency and better performance.

Lockless Primitives

The traditional multi-threaded approach to programming is to use locks to synchronize access to shared resources. Synchronization primitive such as mutex, semaphore and critical section are kernel objects and hence kernel switch times and also associated bookkeeping add lot of overhead to acquire and release lock. Application software was, therefore, optimized using lock-free data structures such as Lockless Queues. This avoided lock acquisition and associated overheads. In cases where it is impossible to eliminate the need for locks, application software used spin-locks based on atomic operations such as compare and swap (CAS).

Zero Memory Copy

A zero copy message transfer mechanism was incorporated which completely eliminated the overheads involved in copying or cloning. The same pre-allocated memory object was used throughout from data acquisition to data transmission. This relieved the CPU of the task of copying data from one memory area to another and improved performance by saving processing power and memory use while sending acquired data to the host.

Objected Oriented Design

Adoption of Object oriented design for the application software kept the design modular, reusable and composable. The use of well proven architectural and design patterns [2] and generics has resulted in the software being adaptable and extendable.

Selection of OS

QNX OS with its robust scalable microkernel architecture and bounded response times ensured deterministic behaviour to meet high throughput requirements. The embedded software also needed to be optimized for low memory footprint which could be achieved using QNX. QNX with its microkernel architecture allows embedded applications to be highly configurable. The configured footprint for ECC consists 🖾 of ECC application software, QNX microkernel; TCP/IP stack, drivers and QNX file system.



Figure 1: The Embedded CAMAC Controller with its block Schematic.

APPLICATIONS

ECC have been deployed for accelerator control at FOTIA BARC, Pelletron and LINAC-Pelletron, TIFR and for multi-parameter experiments at NPD, BARC, TIFR and SINP.

PERFORMANCE

The major improvement for accelerator based multiparameter experiments is in the CAMAC readout, which has been reduced to 1.1μ s. With optimum embedded software using QNX operating system, sustained throughput in excess of 1.5MBPS has been observed in the lab irrespective of number of parameters and event rate. This is a vast improvement over earlier CAMAC controllers developed in-house with capability of 250KB/s. ECC also meets requirement of accelerator beam line control application. A scan time of 50 ms has been achieved for 280 parameters in scan mode operation.

CONCLUSION

PC compatible architectures coupled with optimization done in software and hardware design has helped in achieving high throughput for CAMAC based data acquisition systems, which are widely used in accelerator based nuclear physics experiments.

REFERENCES

- [1] K. Jha et al., "Ethernet CAMAC crate controller for data acquisition and control," in DAE Symposium of Nuclear Physics,2008.
- [2] Erich Gamma, Richard Helm, Ralph Johnson and JohnVlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, 1995.

CLIENT SERVER ARCHITECTURE BASED EMBEDDED DATA ACQUISITION SYSTEM ON PC104

J.J. Patel[#], R. Rajpal, P. Kumari, P.K. Chattopadhyay, R. Jha, IPR, Gandhinagar, India

Abstract

The data acquisition system is designed on embedded PC104 platform Single Board Computer (SBC) with running Windows XP Embedded operating system. This is a multi channel system which consists of 12 Bit. 10 MSPS Analog to Digital Converters with on board FIFO memory for each channel. The digital control and PC104 bus interface logic are implemented using Very High Speed Hardware Description Language (VHDL) on Complex Programmable Logic Device (CPLD). The system has provision of software, manual as well as isolated remote trigger option. The Client Server based application is developed using National Instrument CVI for remote continuous and single shot data acquisition for basic plasma physics experiments. The software application has features of remote settings of sampling rate, selection of operation mode, data analysis using plot and zoom features. The embedded hardware platform can be configured to be used in different way according to the physics experiment requirement by different top level software architecture. The system is tested for different physics experiments. The detailed hardware and software design, development and testing results are discussed in the paper.

INTRODUCTION

The main objective to develop this system is to provide PC based simple and easy solution for low channel data acquisition requirement to support the basic physics experiments carried out by the students. These experiments often require few channels with high sampling rate with short acquisition time. To avoid using high end resources for data acquisition application for small experiments, the system is developed which requires very minimum resources and knowledge to operate it. The Client Server architecture based embedded data acquisition system is developed on PC104 [1][2] platform which houses inside standard industrial 6U, 16T enclosure which requires minimum space to install and operate. As far as resources concerned, It requires only Ethernet local area network connectivity for full fledge operation. The basic data acquisition hardware consists of 5 channels with simultaneous sampling pipelined ADCs, which are capable of doing sampling from 1Khz to 10MHz. Each channel consists of 64K location of FIFO memory. The hardware design is very flexible and all data acquisition parameters are software controlled. The system can be operated in manual as well as external trigger mode and it also accepts external clock. The basic system level operational block diagram is shown in Figure 1.

#jjp@ipr.res.in



Figure 1: Basic system block diagram.

HARDWARE

This embedded board is designed and assembled inhouse and it contains PC104 based Single Board Computer which is procured commercially. The designed and developed board contains analog, digital and mixed signal components. The board is designed and fabricated on four layered PCB. The hardware level block diagram and the selected components are shown in Figure 2.

- Analog front end is designed with AD524 [3] high bandwidth Instrumentation Amplifier followed by AD8041 high speed ADC driver which translates the signal level to dynamic signal range of pipelined ADC AD9220.
- Each channel contains dedicated FIFO memory of size 64K which has total three memory status flags i.e. half, full and empty which can be used for status monitoring and control purpose.
- All digital operational logic like hardware initialization, clock control, ADC and memory control, PC104 bus signal decoding are implemented in Xilinx Complex Programmable Logic Deive (CPLD), which is written in VHDL.
- The acquisition module supports internal as well as isolated external clock and trigger.
- The system contains Advantech make PCM-3353F [4] Single Board Computer which runs on Windows XP Embedded Operating System installed on 4GB size of compact flash card.



Figure 2: Basic hardware block diagram.

SOFTWARE

The client server architecture is the most suitable for this type of application, as it requires minimum installations at client side and uses local area network for data exchange. It also supports remote controlled operation which is our prime requirement.

The software development is done in National Instrument's LabWindows/CVI [5] which is ANSI C based development environment on windows platform which also provides good support for graphical user interface (GUI) development.

The LabWindows/CVI TCP Support Library provides easy-to-use callback functions to create TCP server and client applications, which are shown in Figure 3. The Callback functions provide the mechanism for receiving notification of connection initiation, connection termination, and data availability.



Figure 3: LabWindows/CVI TCP support functions.

APPLICATION

The software development is divided in main two parts server application and client application. The custom protocol is designed using the data packets, which are exchanged between server and client application to establish the operation, control and status monitoring of the embedded hardware.

Server Application

The Server application runs on the embedded hardware platform and it mainly controls the hardware parameters of the module depending on user inputs provided through client application running on remote networked machine. Server application generates the control commands and reads the status of the hardware through PC104 bus. The generated PC104 bus signals are decoded by the firmware implemented on CPLD and it generates the low level hardware signals as shown in table 1.

Table 1: PC104 Address Decoding

PC104 Address	Decoding
0x306,0x308,0x30A, 0x30C,0x30E	16 bit read cycle for each channel data read out(read)
0x301	FIFO Reset (write)
0x302	FIFO Clock Enable (write)
0x303	Trigger Status (read)
0x304	Trigger Mode (write)
0x305	Clock setting (write)
0x309	Half Flag Status (write)

The GUI of server application is shown in Figure 4. The connections details and the commands received from the client application are displayed in the text boxes. After decoding the command, the server application sets the hardware parameters and the status is indicated on the panel in form of illuminated leds.

		<u>_</u>	Start Aquisit
			۲
Clea	ar Beceive Screen	<u> </u>	Mode Selec
eceive:			
New connection from 192.1	68.201.78	_	
- New connection from 192.1 110491020910309101	68.201.78	<u> </u>	Clock Sele
- New connection from 192.1 110491020910309101 - Online - 🎲	68.201.78	4	Clock Sele
- New connection from 192.1 10491020910309101 - Online - 🎲	68.201.78	4	Clock Sele Initialise
- New connection from 192.1 10491020910309101 - Online	68.201.78		Clock Sele
New connection from 192.1 10491020910309101 -Online - 🌑 Server IP: 192.168.204.133 Server Name:	68.201.78 Connected: Client IP: 192.168.201.78 Client Name:	*	Clock Sele

Figure 4: Server application GUI.



Figure 5: Client application GUI with acquired data.

Client Application

The client application runs on any networked computer and it remotely controls the embedded data acquisition system. The GUI of client application is shown in Figure 5. It supports plotting of all channels and some basic data analysis utilities like zoom, restore and plotting from files.

After successful connection with server, all the parameter settings buttons are activated through which user can set the data acquisition parameters of the embedded system. These are embedded in the data packets with the starting string number '91' followed by the data parameter. After completion of the data acquisition, the server application transfer the data of each channel which are received by the client application and stored in the file based database for particular defined shot number. The proper reception of the file is indicated in the receive text box with the starting string number '92' which is sent by the server application and followed by the channel number. The connection details and the acquired data are shown in Figure 5.

The assembled and tested system with all components is shown in Figure 6.

ACKNOWLEDGMENT

The authors sincerely expresses thanks to all the members of Electronics Group.



Figure 6: Embedded Data Acquisition System.

REFERENCES

- Y. Guozhen, S. Qiufeng, L. Li, "Design of electric power data acquisition card based on PC/104 bus", International Conference on Electrical and Control Engineering-2010 (ICECE 2010), pp. 5130–5133.
- [2] PC/104 Consortium; http://www.pc104.org
- [3] http://www.analog.com
- [4] http://www.advantech.in
- [5] http://www.ni.com

A LARGE CHANNEL COUNT MULTI CLIENT DATA ACQUISITION SYSTEM FOR SUPERCONDUCTING MAGNET SYSTEM OF SST-1

K. Doshi[#], S. Pradhan, H. Masand, Y. Khristi, J. Dhongde, A. Sharma, B. Parghi, P. Varmora, U. Prasad, D. Patel, IPR, Gandhinagar, India

Abstract

The magnet the Steady-state system of Superconducting Tokamak-1 at the Institute for Plasma Research, Gandhinagar, India, consists of sixteen Toroidal field and nine Poloidal field Superconducting coils together with a pair of resistive PF coils, an air core ohmic transformer and a pair of vertical field coils. These coils are instrumented with various cryogenic grade sensors and voltage taps to monitor its operating status and health during different operational scenarios. A VME based data acquisition system with remote system architecture is implemented for data acquisition and control of the complete magnet operation. Client-Server based architecture is implemented with remote hardware configuration and continuous online/ offline monitoring. A JAVA based platform independent client application is developed for data analysis and data plotting. The server has multiple data pipeline architecture to send data to storage database, online plotting application, Numerical display screen, and run time calculation. This paper describes software architecture, design and implementation of the data acquisition system.

INTRODUCTION

The Steady-state Superconducting Tokamak (SST-1) is a medium size fusion device to study the plasma behaviour in steady state operation of 1000s [1]. SST-1 magnet system consists of sixteen Toroidal field (TF) and nine Poloidal field (PF) Superconducting coils, together with a pair of resistive PF coils, an air core ohmic transformer and a pair of vertical field coils [2]. These magnets need to be monitored during all operational scenarios like cool down, ramp up, flat top, plasma breakdown, dumping/ramp down and warm up. More than 500 sensors, like temperature sensors, voltage taps, venturi flow meters, hall probes, pressure sensors, and displacement transducers are mounted on the magnet system and its auxiliaries. These sensors produce very low level signals and so they impose very strict requirements on the signal conditioning and acquisition electronics [3]. Magnet Data Acquisition System (DAS) is required to amplify and filter the signal, digitize the signal, acquire and store it, display the data in engineering units and communicate with other subsystems, all in real time. To satisfy all these requirements, a complete VME bus-based real time data acquisition system is designed and implemented for SST-1 magnet system. The hardware consists of 64 bit backplane VME chassis, SBS- VG4 crate controller with PowerPC-755 processor from SBS technologies, Analog input module IP330 and Analog output module IP 220 from ACROMAG, 6U AVME 9660 carrier board with four daughter card slots from ACROMAG, Digital I/O module VMIVME-2528 from VMIC and a GPS timing module BC635VME from Symmetricom Inc.

PROJECT SCOPE

The VME based DAS is suppose to monitor different sensors mounted on the SST-1 magnets and its auxiliaries, continuously and reliably during different phases of SST-1 magnet operation. The total numbers of required channels are more than 500 in numbers which are required to be monitored continuously over long duration during an experimental campaign. Table 1 shows typical distribution of data acquisition channels for magnet systems with different sensors and its primary application. Sensor signals are coming from the different signal conditioning cards mounted in the instrumentation racks in the magnet control room [4] and are interfaced with VME IOs through signal connection box. Figure 1 shows the functional block diagram of VME DAS with its associated hardware subsystems.

Table 1: Channel distribution for Data Acquisition system

No.	Sensor	Channels	Application
1	Temperature Sensors	150	Cryogenic Temperature Measurement
2	Voltage-tap channels	224	Quench detection and magnet voltages
3	Hall probes	16	Magnetic field direction and intensity measurements
4	Flow meters (Venturi)	32	Flow measurements
5	Absolute Pressure	16	Helium Pressure measurement
6	Displacement transducers	12	Displacement measurement in cold mass
7	Strain gages	8	Stress measurement on magnets
8	Joint Resistance Measurement	102	Inter-pancake and Inter-coil joint resistance measurement

Je

þ

2012

C


Figure 1: Functional block diagram of the system.

APPLICATION ARCHITECTURE

To cater to the varying need of magnet system, a versatile multi client software application was developed for the VME based hardware. The software application can be divided in three software modules. First module is a target board application which will take care of configuration parameters and data transfer from the data acquisition hardware channels. Second module is a Java server application, which will receive data from the target board application, store data into database and sends data to the client applications. Third module is a Java desktop intranet application, which is a graphical user operator interface used for set configuration, data plotting, data analysis and data monitoring etc. TCP/IP socket programming is used for communication with the data server and user database. One TCP/IP server is required on target side that will communicate with host PC running a TCP/IP client. The server application is developed in JAVA and is ported on freely available sun java / Apache Tomcat web server. This web server communicates over TCP/IP with the dedicated target application as well as multiple client applications. Raw data is archived on host PC while online trends and numerical values are displayed on client applications. Figure 2 shows the application architecture block diagram of the DAS. Following section discuss each module in detail.



Figure 2: Application architecture of the data acquisition software system.

Target Board Application

Target board application takes care of the configuration parameters of the data acquisition hardware settings like set no. of channels to be used, acquisition mode, scan rate, trigger mode etc. This is a platform dependent application running over VxWorks 5.4 real time operating system (RTOS) platform and is developed in gnu C/C++, with Tornado 2.0.2 as IDE. It acquires the analog signal from the signal conditioning hardware and converts it into digital data using ADC cards, and store into logical memory. Moving average type filter is implemented on raw data which can be configured for selected channels. No. of samples to average is adjustable depending upon signal sampling rate. It has TCP/IP socket API to communicate with the host program running web server software. Using DAC channels, Digital data can be converted into analog form selected by the application server. It also has the VxWorks API for configuring DI/O card to set I/O modes, no. of channels etc to perform the operation of displaying the status of the quench channels or to detect the trigger from the different subsystems to indicate the same to the Application server. Time stamping is implemented with GPS receiver time from the central control room to time synchronize the dtat acquisition with all other subsystems. In the absence of GPS card the application will automatically switch over to the onboard controller RTC time with a ststus update at server and an indication display at client window.

Application Server

Second module is a Java server application, which will receive data from the target board application, store data into database and forward it to multiple client application as per demand. It contains business logics like Roles, Data management, Authentication, Data storage and analysis. The main aim of this module is to acquire data at high speed and transmit this data over Ethernet intranet network. Along with server the host machine accommodates two databases for the application. One is MYSql database which is used to store user login details, roles, configuration details etc. Whereas other database is H2DB which is the file database used to store the acquired data. This database facilitates offline data analysis as per user requirements at client end through different analysis tab. Here, in this application the file database is used instead of the file because of the performance advantage of the file database over the file in sense of the searching facilities for the off-line data plotting and analysis.

Socket Utility is implemented to receive the set configuration, get configuration, start acquisition, stop acquisition, acquire raw data and acquire calibrate data commands from the client application. It forwards these commands to the target board application, receives the acquired data from the target board application, stores it into the database and forwards acquired data to all clients who had requested it for the on-line plot. Delegate is implemented for activities like authenticate user, user creation and role assignment, provides the configuration details, provides off line data and configuration of the application.

Desktop Client Application

Third module is a JAVA desktop intranet application, which is a rich graphical user interface (GUI) used to set configuration, send commands, data plotting, data analysis and many other functions. Initially, this module was planned to be implemented as a web based interface, but then it was implemented as a desktop application due to its higher speed and better reliability. Set configuration is used to program hardware for the required application setting through a series of pop-up windows. There are two types of users defined for the software operation, Administrator and Normal user. Administrator has all the access rights mainly to configure VME hardware configuration parameters. Normal users have read only permission, he can see current configuration setup and has access to data for analysis and plotting. Only one administrator login is allowed at time. Different tab windows are provided for data monitoring and plot display. As the no. of channels are large, different groups, colours and line styles are provided to show large number of channels at once. Output channel display supports sensor readings in absolute unit (Kelvin, Gauss, mm etc) using internal conversion of raw data with interpolation equation, formula or by reading nonlinear calibration curves. Java client application will provides login facility for the security purpose with defined user rights as per role assignment. GUI provides the facility for the shot comparison, export the data in form of excel, binary and CSV and a scheduler for data backup at predefined interval. A report generation tab is provided for generating report of analyzed data as well as channels configuration.

RESULTS AND DISCUSSION

The DAS was used successfully during the recent SST-1 engineering validation experimental campaign. The VME processor, Server PC and an administrator client were kept in the field at magnet control area. A local intranet network was established using a five port switch with an uplink connection from the central control room SST network. All the user client machines were operated remotely from the central control room. In the event of network problems or failure client will be disconnected but the VME and server will always be running and acquiring the data due to local intranet connection. Client application is made as a desktop application instead of web based application and has worked fine. In case of web application the rich graphic part will take large amount of network bandwidth and can make online plotting slower in case of large data access. Desktop application has a local graphics and only data will be sent over network reducing the bandwidth requirement. Web server and database were integrated in the same machine and has worked uninterruptible fashion.

CONCLUSION

A large channel count multi client DAS was developed for the magnet systems of SST-1. The application has run continuously over several weeks of magnet operation and different modules and features were tested during the execution. The software has shown minor bugs initially and malfunction in data storage sequence and labelling when all the channels were selected simultaneously which were corrected during the usage. The software application will further be modified in client user interface for better visualization and ease of use.

ACKNOWLEDGMENT

The authors are thankful to their control group colleagues and SST-1 team members for their support and help during the development.

- [1] S. Pradhan, "Status of SST-1 Refurbishment", J. Plasma Fusion Res., vol 9 (2010), pp 650-657.
- [2] S. Pradhan, "SST-1 Toroidal field magnet tests: some results and lessons learnt", IEEE Transactions on Applied superconductivity, vol 22, No. 3 June 2012, 9501804.
- [3] K. Doshi, Khristi Y., Kedia S., & Pradhan S., IEEE trans. on Inst. and Meas., vol. 60, no. 3, March 2011.
- [4] K. Doshi et. al., IEEE Trans. on Plasma Science, 40, 641-645, 2012.

SERIAL MULTIPLEXED BASED DATA ACQUISITION AND CONTROL SYSTEM

N. C. Patel*, C. Chavda, K. Patel, IPR, Gandhinagar, India

Abstract

Data acquisition and control system consists of analog to digital converter (ADC), digital to analog converter (DAC), timer, counter, pulse generator, digital input / output (DIO) depending upon requirement. All the system components must communicate with personal computer (PC) for data and control signal transmission via one of the communication protocol like Serial, Parallel, USB, GPIB. Serial communication is advantageous over other protocol due to several reasons, like long distance data transmission, less number of physical connection, ease of implementation etc. The developed Serial Multiplexed based Data Acquisition and Control System, which can control different modules like ADC, DAC, DIO, Timer card using single serial port. A LabVIEW based program is developed for the individual communication of each module.

INTRODUCTION

The basic data acquisition and control system consists of different types of application module like ADC, DAC, timer, counter, pulse generator, DIO etc. The module is selected depending upon the requirement. The modules are individually controlled with personal computer (PC) for data and control signal transmission using one of the communication protocol. If communication is done using different protocol for different modules in system requires knowledge of different communication protocols, communication hardware and large number of connection with the PC.

In order to overcome the above mentioned difficulties, we developed Serial Multiplexed based Data Acquisition and Control System (SMDACS). In house developed system consists of different application modules and a controller module. The application modules are controlled by controller module having serial connectivity. The control program for each application module is developed in Lab-VIEW environment.

FUNCTIONAL DESCRIPTION

A block diagram of SMDACS is shown in Fig. 1. The system consists of different application modules which are installed on back panel (mother board) of the system. The physical address for the application module is set on mother board from 000 to 111. When system is switched ON, the application module read the physical address and saves in local register of microcontroller. While transferring command for Read / Write, the logical address is sent first. The microcontroller in application module compares logical address and physical address.

*ncpatel@ipr.res.in

The module program command sequence is executed in the application module whose logical address and physical address matches. All the commands are treated as either read or write considering as receive or send by PC. Data and commands are sending or receive by the PC to the application module via serial interface. The application module can be installed in one to eight locations while controller is placed on ninth position. The application module has no physical position limitation, i.e., any module can be installed in any position except the controller module.



Figure 1: Block diagram of developed SMDACS.

Figure 2 shows the developed SMDACS assembly. The assembly consists of 3U size, nine slot chassis. The chassis as controller module, application module, back panel and in-built power supply. The right-most module is the controller module. Each application module communicates with the controller module using back panel 15 pin D-type for transmit, receive and power. Three pins of the15-pin D-type connector are used for physical address of the module. The application module is used physical address for data and control information transmission. Each application module consists of logic gates, microcontroller (AT89C52) and other related components.

Controller Module

Controller module is the heart of the system. It is the interface module which transmits data and control signals between PC and application module. Front panel of the module has power indicator and a data transmission indicator. It has an external trigger which is used as bus

trigger for the entire application module simultaneously. A 9 pin D-type connector is used for communication using serial bus .A 25 pin edge connector on the back panel of this module is used to transmit and receive data or control signal.



Figure 2: Developed serial multiplexed based data acquisition and control system hardware.



Figure 3: LabVIEW based GUI program for application module.

The system uses two types of programs for its operation namely system program and module program. System program as LabVIEW Graphical User Interface (GUI) which is used by users while the Module program is the program written in the microcontroller of the application module. The LabVIEW GUI program controls the operation of system as shown in the Fig 3. While executing the LabVIEW GUI program, the following sequence is followed:

- Configure serial port by VISA resource name. •
- Select Station ID (logical address) and other related parameters which are to be passed to the different application modules.

- Activate required Station ID for passing the parameters (All or specific station).
- Select appropriate command (Write or Read). •
- The program can stop forcefully by Stop command. •

Application Module

Module program is written inside The the microcontroller [1] of the individual application module. When power is switched ON, the microcontroller inside individual module will read physical address from the back panel and write it in the local memory uC. Figure 4 shows the flow diagram of module program. The LabVIEW program sends the logical address to application module through serial communication. The logical address send by the LabVIEW program and physical address written in microcontroller memory is compared. If the match in address is found, next commands end by the LabVIEW program will be executed on that particular module. Depending upon the command send by the LabVIEW program, data / control word will be read /write in particular application module using the serial interrupt. All the other interrupts are disabled whenever application module places data on the serial bus.



Figure 4: Flow diagram for module program.

The details of the different application modules are as follows:

TTL Delaved Pulse Generator Module

The TTL delayed pulse generator module is one of the application modules. It generates a variable delay between trigger input and two independent channel output. The trigger can be external hardware trigger or bus trigger or software trigger. Both channels of the module have 50 ohm driving capability. The duration of the delay can be set by LabVIEW GUI. The delay duration data is written in the application module microcontroller local memory by LabVIEW program. When this application module receives the trigger, it will generate pulses with required delay with respect to the trigger. Figure 5 shows the delayed pulses generated with respect to trigger for two different channels.



Figure 5: Timing diagram (CH-1 External Trigger Pulse, CH-2 output pulse of channel-1 and CH-3 output pulse of channel-2).

Digital I/O Module

The DIO module consists of eight digital inputs and eight digital outputs. This module is developed using AT89C52 microcontroller. The module read the digital input and displays the status of each bit on the Lab-VIEW GUI. The different digital pattern is generated by setting the bit pattern in the GUI. The GUI transfers the digital pattern to the digital output port.

Digitizer Module

Figure 6 shows block diagram of the developed 8bit digitizer module using ADC (ADC0804), Memory (K6X4008) and microcontroller. The module is initialized by the number of sample to read and the mode of trigger. The ADC module continuously read the required number of samples after getting start trigger. The start trigger of the module is selected using either software or bus trigger or external hardware trigger by GUI. The developed module has maximum storage capacity of about 64 KB. After getting the trigger, the microcontroller read the digital data from ADC0804and writes the required number of sample in the memory. When module is selected for reading the data, it will transfer data from the module memory to PC via serial bus. The data for the "number of data to be read" is written by LabVIEW program. It will write the data in user defined file format. User can define a file name or append to a file. The stored data can be retrieved and analysed as per requirement using Origin or MATLAB software.



Figure 6: Block diagram of the digitizer module.

Digital to Analog Module

The 8 bit DAC (AD558) is used for the conversion of the digital to analog signal. The required control signals for the DAC are generated using microcontroller. The required output voltage from the DAC is defined in the LabVIEW program. The digital number corresponding to the voltage is loaded in the DAC input using microcontroller. This number is loaded in the microcontroller using LabVIEW "write" command. The DAC module generates corresponding analog voltage from 0 to 10 V on front panel BNC in step of 39 mV.

CONCLUSION

The developed SMDACS is stand alone and used in small experiment. The system can be used for acquiring slower sampling data for longer duration using digitizer module. Other than this the module generates different timing pulses using TTL delay generator to synchronize with other system. The system also generate digital pattern or to acquire system status using digital input / output module. The analog signals generated using digital to analog converter is used for analog pattern generation.

ACKNOWLEDGMENT

The authors wish to acknowledge the contributions of the ADITYA data acquisition team. The authors are thankful to S. Sunil and Jinto Thomas for critically evaluating the manuscript.

REFERENCES

[1] Muhammad Ali Mazidi and Janice Gillispie Mazidi, "The 8051 Micro controller and embedded systems using assembly and C", second edition.

VEPP-2000 LOGGING SYSTEM*

A. Senchenko[#], D. Berkaev, BINP SB RAS, Novosibirsk, Russia

Abstract

The electron-positron collider VEPP-2000 has been constructed in the Budker INP at the beginning of 2007 year. The first experiments on high-energy physics has been started at the end of 2009. The collider state is characterized by many parameters which have to be tracked. These parameters called channels could be divided into continuous (like beam current or beam energy) and pulsed. The main difference is that the first one related to the moment of time while the second one to the beam transport event. There are approximately 3000 continuous channels and about 500 pulsed channels at the VEPP-2000 facility. The Logging system consists of server layer and client layer. Server side are a specialized server with an intermediate embedded database aimed at saving data in case of external database fault. Client layer provide data access via API, CLI and WUI. The system has been deployed and is used as primary logging system on VEPP2000.

VEPP-2000 PROJECT

VEPP-2000 is a new collider with luminosity up to 10^{32} cm⁻²s⁻¹ and the beam energy up to 2×1 GeV [1, 2].



Figure 1: VEPP-2000 facility layout.

This project is a development of a previous facility of VEPP-2M which has worked at BINP over 25 years in energy range up to 1.4 GeV in c.m.s. and has collected of about 75 pb⁻¹ integrated luminosity. New collider uses the existing beam production chain of accelerators: ILU – a pulsed RF cavity with a voltage of 2.5 - 3 MeV, a 250 MeV synchrotron B-3M and a booster storage ring BEP with the maximum project beam energy of 900 MeV (see Figure 1). The lattice of VEPP-2000 has a two-fold symmetry with two experimental straight sections of 3m length, where Cryogenic Magnetic Detector [3] and Spherical Neutral Detector [4] are located. Two other long straights (2.5m) are designed for injection of beams

and RF cavity, and 4 short technical straight sections accommodate triplets of quadrupole.

The closed orbit steering and gradient corrections are done with 1-2% coils placed in the dipole and quadrupole magnets.

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, there are also 4 pick-up stations in the technical straight sections and one current transformer as an absolute current monitor.

The magnetic field of 2.4 T in the bends is required to reach the design energy of 1 GeV in the constrained area of the experimental hall.

CONTROL SYSTEM (SOFTWARE)

Modern accelerator facility is a complex system both physical and technical meaning. It consists of many subsystems and units, which are difficult to automatize.

Collider state is characterized by many parameters which have to be tracked. These parameters could be divided into two groups: continuous (like beam current or beam energy) with typical speed of change from two time at second up to one time at ten minutes and pulsed which are related to some event (e.g. beam transport event called "Shot"). There are approximately 3000 continuous channels and about 500 pulsed channels at VEPP-2000 facility.

Control system is multilayer distributed system with specialized hardware server at lower layer, client program at upper and utility server between them. It was build according to the concept of independent subsystem control. Such approach allows us to create a small program with strictly divided responsibility.



Figure 2: Control system schema.

^{*} Work partially supported by Russian Ministry of Education and Science, basic project of BINP SB RAS 13.3.1, Physics branch of RAS project OFN.1.1.2, Scientific school NS-5207.2912.2 and Grant of the Novosibirsk region Government 2012

^{*}senchenko.alexander@gmail.com

LOGGING SYSTEM

Rationale

The collider state is characterized by many parameters which changes with a different rate. Some time operator can't detect deviation of parameter and analyse it. That's why it is necessary to save changes of all parameters.

Such information could be used for investigation of system faults and correlation of subsystems.

Architecture

The Logging system is built on client-server architecture. Client layer provide data access via API, CLI and WUI. Server side are a specialized server with an intermediate embedded database aimed at saving data in case of external database fault.



Figure 3: Logging system architecture.

Client Layer

Client layer consists of API for interacting with logging server and CLI for accessing data. API and CLI were implemented in C/C++, since most of the client applications were implemented in C++. To simplify extending VCAS [5] with logging, Qt version of API was developed as well.

Web User Interface

Web user interface provide interactive access to stored data. At the present time WUI is implemented in Python using web2py framework [6] and mathplotlib [7]. This framework was chosen for several reasons: author's wide experience with Python development and fast application development.



Figure 4: Web user interface.

Server Layer

The server layer consist of logging server. Logging server is a core of logging system. It perform data collection and data optimization. Logging sever consists of network module, continuous module, pulsed module and storage module(see Figure 5).

Network subsystem handles network communication and dispatches request to the appropriate module.

Continuous module perform data optimization and simple data validation.

Pulse module handles pulse data connect interesting continuous data records to certain pulsed data record.

Storage module saves data to temporary database and moves saved records from temporary database to main database.



Figure 5: Logging server structure.

Temporary databased is used for data buffering before bulk insert to DBMS. It allows to save data in case of main DBMS fault or inaccessibility. For this purpose BerkeleyDB Java Edition [8] was chosen. It has no external non-Java dependencies and could be launched regardless of the installed software.

The prototype of server was developed in python. It had acceptable throughput, but peak loads may cause instability of whole system. Current version of server was implemented in Java. Test shows good server stability in case of 8000 channels at constant load and up to 20000 at peak load.

Optimization Algorithm

The main purpose of optimization algorithm is to reduce disk resources used by stored data.

All continuous channels have different rate of change. Some channels change every second (beam currents), some every 5 seconds or more infrequently. That's why is seems rational to introduce some threshold. If the difference of the previous and new values is less than threshold, then value considered unchanged.



This algorithm allows to reduce the rate of data growth from 3.4 MB/day to 1.28 MB/day per channel.

Data Storage Layer

Logging system produces a large amount of data. To access data in reasonable time, it should be indexed. The

most common decision is to use relational DBMS. Postgresql [9] was chosen. There are several reasons: team of VEPP-2000 has experience of working with it, it is in active development, it has tools for replication and balancing.

CONCLUSION

VEPP-2000 logging system is a part of control system. of system is based Architecture the on client-server approach. It provides interfaces to save and access data. Temporary databased is used for data buffering before bulk insert to DBMS. Test shows ability to process up to 8000 channels at constant load and up to 20000 at peak load without affecting entire system.

Optimization of data is held by server. Optimization reduces data rate reduce the rate of data growth from 3.4 MB/day to 1.28 MB/day per channel, producing up to 2 Gb per day.

- [1] Yu.M.Shatunov et al., "Status of VEPP-2000 Collider", RuPAC'12, St.-Petersburg, Russia, September 2012; MOYCH01, www.JACoW.org
- [2] "VEPP-2000 Project", http://vepp2k.inp.nsk.su
- [3] "CMD-3 Project", http://cmd.inp.nsk.su
- [4] "SND Project", http://www.snd.inp.nsk.su
- [5] A. Senchenko et al., "VEPP-2000 Collider Control System", PCaPAC'12, Kolkata, India, December 2012, FRCB04, http://www.JACoW.org
- [6] Web2py framework, http://www.web2py.com/
- [7] Mathplotlib, http://matplotlib.org/
- [8] BerkeleyDB Java Edition, http://www.oracle.com/technetwork/database/ berkelevdb/overview/index.html
- [9] Postgresql, http://www.postgresql.org/

DEVELOPMENT OF DATA ACQUISITION SOFTWARE FOR VME BASED SYSTEM

A. Kumar, A. Chatterjee, K. Mahata, K. Ramachandran, BARC, Mumbai, India

Abstract

A Data Acquisition system for VME has been developed for use in accelerator based experiments. The development was motivated by the growing demand for higher throughput in view of the increasing size of experiments. VME based data acquisition system provides a powerful alternative to CAMAC standards on account of higher readout speeds (100 ns/word) resulting in reduced dead time. Further, high density VME modules are capable of providing up to 640 channels in a single VME crate with 21 slots. The software system LAMPS, earlier developed for CAMAC based system and used extensively in our laboratory and elsewhere has been modified for the present VME based system. The system makes use of the VME library to implement Chain Block Transfer Readout (CBLT) and gives the option of both Polling and Interrupt mode to acquire data. Practical throughput of ~250 ns/word in zero suppressed mode has been achieved.

INTRODUCTION

With increase in size of Nuclear physics experiments, a secular trend toward higher throughput bus standards has been observed. VME [1] - an acronym for VERSA Module Euro card – provides a powerful alternative to the CAMAC standard on account of higher readout speed (100ns/word) leading to significant reduction in dead time. High channel density VME modules significantly bring down the expenses involved in acquisition of data per channel. Further, usage of optical fiber link makes sure that the interconnect technology doesn't become a bottleneck during data transfer. LAMPS [2] software, earlier developed for CAMAC based system, has been modified for VME based acquisition system. It currently supports CAEN [3] V785 ADC, V775 TDC and V862 QDC and V830 scalar modules. Practical throughput, using VME data acquisition system and LAMPS, of ~250ns/word in zero-suppressed mode has been achieved.

ARCHITECTURE

System Architecture

The VME data acquisition system (see Figure 1) consists of VME digitizer modules plugged into the VME backplane, controlled by master module. VME master module (V2718) is a VME to PCI (Peripheral Component Interconnect) Optical Link Bridge, housed in a 1-unit wide VME 6U (19" x 10.5" x 19.5") module. Master module is controlled using a computer equipped with PCI optical controller card (A2818), capable of transferring data at 80 Mbytes/second. The connection between the master module and controller card takes place through an optical fiber cable.

CAEN digitizer modules for VME provide features like zero suppressed readout and overflow suppression. Zero suppression, once enabled, prevents conversion of value which is lower than user defined threshold. Overflow suppression, once enabled, aborts the memorisation of data which constitutes an ADC overflow.



Figure 1: VME Data Acquisition system: System and Software Architecture.

The maximum VME address space is made of 2^{64} bytes for VME64 standard. Each slave occupies a portion of this space. Geographical addressing and address relocation methods for setting base address are absent in VME64. Thus, base address for slaves has been set at hardware level by means of rotary switches. Various registers of VME slave can be accessed, thereafter, by adding relevant offset to the module's base address.

LAMPS program

LAMPS is a data acquisition and analysis package that supports, apart from VME, a number of CAMAC controllers. It has been written in C and user interface has been implemented using GTK.

It allows for online spectra building (see Figure 2) and can also be used for offline data analysis. It provides, inter-alia, tools for performing calibration, peak fit, peak search, quick fit and obtaining the area and centroid of a spectrum region.

It runs under Linux and can also be made to execute on Microsoft Windows through Cygwin [4]. It is designed for large scale experiments and is in use at BARC-TIFR Pelletron laboratory, since August 2002.

Integration with LAMPS program

The data acquisition program LAMPS was adapted to work with VME hardware. It makes use of the VME library provided by CAEN to transfer data using Chain Block Transfer Mode (CBLT) over an optical fiber link. The vendor supplied library provides function calls to, inter-alia, open and close communication and execute different read/write cycles.

CBLT with interrupts has been used for acquiring data. CBLT mode allows sequential readout of multiple slave modules selected by a single address cycle. It allows for data readouts belonging to same physical event from several contiguous boards in a crate limited to 256 words per CBLT cycle.

Assignment of address to modules in CBLT chain is done at the start of acquisition. The first module in CBLT chain raises an interrupt after output buffer crosses a threshold number of events. Upon receipt of interrupt, CBLT data transfer is initiated which is completely transparent to the master. It makes use of IACKIN-IACKOUT daisy chain line present in VME backplane to propagate the readout token across the CBLT chain.



Figure 2: Spectrum acquired using VME DAQ through LAMPS.

Master gate blocking is essential to have any meaningful acquisition with VME, failing which a good number of events could be corrupt depending on data rate. In our setup, BUSY outputs, from digitization modules in the CBLT chain, are ORed together to block the master gate. This prevents event mismatch/corruption by ensuring that the modules which have finished digitizing quickly, as compared to others, cannot accept the next master gate.

To present the same familiar GUI to users and retain the overall structure of the LAMPS program, only minimal changes in GUI were introduced to accommodate VME specific features. The system has been tested successfully and is in use in at BARC-TIFR Pelletron facility.

CONCLUSION

The VME DAQ in the current form provides us with a powerful system because of the large number of parameters which can be acquired simultaneously and its ability to handle high event rates. Broad based support from all leading vendors has given an impetus to the adoption of VME based system and has equipped users with a potent alternative.

- [1] http://www.vita.com/home/Learn/vmefaq/vmefaq.html
- [2] http://www.tifr.res.in/~pell/lamps.html
- [3] http://caen.it/csite/Product.jsp?parent=11
- [4] http://cygwin.com

MICROCONTROLLER BASED DAQ SYSTEM FOR IR THERMOGRAPHY BY HOT AND COLD WATER FLOW

M. S. Khan, K. D. Galodiya, S. M. Belsare, S. S. Khirwadkar, T. H. Patel IPR, Gandhinagar, India

Abstract

There are many Non Destructive Techniques used in science and industry to evaluate the properties of a material, component or system without causing damage. Infrared Thermography (IR) is one of them. Different types of IR thermograph is used for different purpose. We are using hot and cold-water flow IR Thermography method to evaluate the Performance of Plasma Facing Components (PFC) for Divertor Mock-up [1].

The Set-up is designed in such a way that hot and cold water can flow in both direction inside mock-up, like left to right and right to left using electric pumps. Eight numbers of Solenoid Valves have been used for selection of Water Flow Direction, thermo-couples for temperature measurement of water, IR camera to take the images and many others devices. This needs a very good and versatile DAC system. We have developed a DAC system using micro controller and LabVIEW for the acquisition of various parameters and controlling & synchronization of other system. Development of DAC is described in this paper.

INTRODUCTION

The main aim of this IR thermography is to evaluate the quality of the braze joints between PFC tiles and the copper alloy (CuCrZr) heat sink. Figure 1 shows the schematic of PFC test mock-up and experimental arrangement for IR-NDT. PFC is placed in front of IR camera at a particular distance such that the FOV (Field of View) of IR camera can cover array of tiles. The IR camera captures thermal evolution at the tile surface and transfers the information to the computer for data storage device for further processing. We have developed an IR thermography to test the PFC components by Hot and cold water flow method. This paper described the DAQ for hot and cold water IR thermography by using microcontroller 8051 and Lab VIEW.



Figure 1: Schematic of IR-Thermography.

HOT & COLD WATER FLOW LOOP SETUP FOR IR-NDT OF PFCS

A schematic diagram of water flow loop facility to be developed for IR-NDT of PFCs is given in the figure-2. The setup is designed in such a way that hot as well as cold water can flow in both direction inside PFCs, like Left to Right and Right to Left.

Eight numbers of water valves have been used for selection of hot & cold water and also used to alter their flow directions. Flow rate is controlled by VFD. A digital water flow meter is located at the out-let of the water pump for measuring the flow rate of water passing through it. Two numbers of 1.5 kW electric water-heaters are used to heat the water in hot water reservoir. Four numbers of thermocouples are used to monitor the temperature of water at various locations. A water cooler is used to provide cold water. Table 1 shows the selection procedures for Hot & Cold water and their flow directions.



Figure 2: Hot and Cold water flow loop setup.

BLOCK DIAGRAM OF DAQ

Block diagram of micro controller based DAQ System is shown in Figure 3. We have used microprocessor 8051. Serial communication RS-232 is used between PC and micro controller. Solenoid driver circuit is used to operate the solenoid valves. Variable frequency driver is used to run the water pumps at different pressure require for IR thermo-graphy. A signal is given to VFD from micro controller to operate the water pump and same signal is also applied to IR camera to store the images. A flow chart is shown in Figure 4.

S.N.	Selection of Hot/Cold Water	Direction	Open Valves	Close valves
1	Hot	Right to Left	1,7,6,2	3,5,8,4
		Left to Right	1,5,8,2	3,7,6,4
2	Cold	Right to Left	3,7,6,4	1,5,8,2
		Left to Right	3,5,8,4	1,7,6,2

Table 1: Selection of Hot & Cold Water and their Flow Direction.

CIRCUIT DIAGRAM

Circuit diagram of DAQ system using micro controller P89V51RD2 is shown in Figure 5. Port P1 is used to control the solenoids, water pumps and hot & cold conditions. MAX 232 IC is used for the serial communication with the PC.



Figure 5: Circuit diagram of DAQ system using micro controller P89V51RD2.

USER INTERFACE

We have developed the graphical user interface using LabVIEW 2011. Front panel of the LabVIEW is shown in Figure 6. First we have to select between right and left, after that we have to select between hot and cold.



Figure 6: Front panel.



Figure 3: Block diagram of microcontroller based DAQ system.



Experimental Data Acquisition

LABVIEW CODE

We have developed Lab VIEW code for four conditions. Two for cold water flow and two for hot water flow. For each hot and cold water again divided in two parts. One is for left to right flow and second is for right to left flow. One of the LabVIEW code is shown in Figure 7.



Figure 7: LabVIEW code for hot water, left to right flow.

HARDWARE SETUP

Hardware circuit is shown in Figure 8. We have tested the circuit using LED in place of solenoid and water pumps. Now we will integrate the circuit with the electrical control panel.



Figure 8: Hardware circuit on general purpose PCB.



Figure 9: Electrical control panel.

ELECTRICAL CONTROL PANEL OF HOT AND COLD WATER CIRCULATION SYSTEM

Hot and cold water system electrical panel is shown in Figure 9. We will use the developed DAQ system in such a way that we can operate the system for the control panel as well as from the PC remotely.

RESULT

We have successfully implemented and tested the hardware circuit on a general-purpose board. at present we have tested it with led in place of solenoid valve and motors. now we will integrate it with the electrical control panel.

REFERENCE

[1] A.Durocher, R.Mitteau, V.Paulus, J.Schlosser, Interface Quality Control by Infrared Thermography Measurement

SMART STRUCTURED MEASUREMENT PROCESS FOR VERSATILE SYNCHROTRON BEAMLINE DATA AT ANKA

T. Spangenberg*, D. Haas, W. Mexner KIT, ANKA - Synchroton Light Source, Karlsruhe, Germany

Abstract

An unstructured measurement process might deliver the needed quantity of primary data for an experiment. But the achievement of the scientific results depends more and more from the offered opportunities of embedding these measurement data into its specific context with a metadata description and a complete life cycle management.

Obviously the design of a measurement process influences the potential applicability of an experimental setup for its scientific purpose and of course its options to fulfil a contemporary data management. ANKA's Tango based environment offers in principal varying approaches with different implementation efforts and coverage of scientific or information technology requirements.

At ANKA we have set up a smart structured measurement process which stand out due to its seamless integration into the overall data management, the support of recent control concepts for fast data generation as well as its support of well time-tested SPEC based scan systems. The presented measurement process focuses to the minimal implementation for all involved components without a break of well accepted habits.

INTRODUCTION

Currently at synchrotrons used control systems like EPICS, TACO, or TANGO [1] are focusing to the problem of retrieving and delivering digitized data from the measurement process at the experimental stations as well as at all peripheral equipment.

The growing amount of available digitized measurement data, which is organized by hand in a first straight forward approach, led to a situation which does not permit an automated data management. Solutions for crosslinking and archiving them are strongly demanded. Some effort is done - not only in the synchrotron community, e.g. LSDF[2] and HDRI[3] - to implement an extended data management to these scientific data.

The new at ANKA introduced smart concept offers the base for the implementation of the desired automated overall data management with minimised efforts.

The experimental beamline stations at ANKA were originally designed on the basis of SPEC [4] (see Fig 1, part A). In this context the SPEC output defines which meta- and measurement data will be saved in an ASCII-file. Measured data is collected from fast (~ms) directly by spec driven devices (green part in Fig. 1) or other slow speed-triggered-TANGO-devices (blue part in Fig. 1). As SPEC can be used as TANGO server and client at the

same time [5] it is used as a *user interface* as well as a *scan server* for X-ray beamline experiments.



Figure 1: The logical beamline layout (The realised part A and the currently developed extension in part B).

Autonomous devices outside SPEC are partially synchronized and triggered by an electrical signal based trigger and gating system (yellow in Fig. 1). The disadvantage of this scenario is that each device is generating its own unmanaged data file separately. Also the WinCC OA [6] overall SCADA system collects asynchronously and independently peripheral data in parallel. WinCC OA was extended by TANGO server and client capabilities [7].

The resulting already three data sources (symbolised as disks in Fig. 1, part A), indicates a strong evidence that there is a reasonable demand for an *experiment coordination service* (ECS).

The role of such an ECS in the experiment workflow is in general not a new idea and was in our case already implicitly present by the coded scheme of SPEC. The sequential synchronous nature of that data acquisition approach so far was the major reason to separate it from spec and set up a special dedicated ECS device server.

The key for interlinking the retrieved information is the *metadata*. Therefore the objective will be achieved by a careful attention to the dedicated retrieval and processing

^{*}thomas.spangenberg@kit.edu

this type of data. The currently used control systems are offering no generalised approach to this issue.

The missing standardized functionality on control system level currently is leading to different site dependent solutions. At ANKA we are implementing a new smart approach addressed to that problem which respects the current control system implementation. Very special attention was given to avoid bottlenecks in speed or to loose flexibility for the future.

EXPERIMENT COORDINATION SERVICE

A distributed TANGO setup of independent servers as it is shown in Fig. 1 requires a task scheduler and a dedicated system for collecting metadata. Both are provided by the ECS which acts as a TANGO server and client.

Created data has to be announced to the ECS by the cooperating scan server or other special devices (e.g. WinCC OA). Typical collected information are filenames and data locations, owner of the data and directives for their further processing. The measured data itself is not touched at this stage of processing. Furthermore peripheral logging data (e.g. vacuum pressure, air humidity) can be collected and processed for archiving.

The ECS is focused to the needs of a further automated organisation and archiving of the measured data. The implemented ECS manages an order queue (FIFO) of experimental demands (ED) (see Fig. 2). An ED-tag contains 3 sub tags referencing the scan server, the results and the administration part. As the majority of other interactions to that device they are injected as a string by a simple TANGO command. The string is formatted as a XML based telegram.

A XML telegram might be considered as a container for arbitrary information since any additional content can be added to the ED-tag of the experimental demand. Conforming to XML processing guidelines this additional unchanged content needs to be transferred to following processing stages.

As at ANKA the current user interface and scan server are covered by SPEC, this new ECS approach becomes elegant possible by only a few macro extensions. For any future development no conceptual limitations are expected. The only specific requirement is that GUI and scan server are cooperating by the XML transmitted content.

INTEGRATION

The workflow of measurements with SPEC is on one hand defined by macros and might be easily varied. On the other hand the habits of the users are strongly engrained and therefore recommend a hidden change in the data collecting strategy.

From the users view the data collection is defined by a 3 level process. *First* of all the general output directory for the whole experimental series is defined. This is



currently done by an external script. In a second step the

basis name of any output is defined by a macro in SPECs

Figure 2: Schematic construction of processing EDs.

command line (newfile). As a *third* step the actual scan macro realises the desired scan and with it the data collection is done. The data is stored in SPECs ASCII file.

Additionally triggered devices - electrically by the yellow trigger or by the TANGO bus (see Fig. 1) are offered directories to store their data. Following-up the smart integration concept the produced results has only to be declared to the ECS. A modification or special adaption of the used device servers with respect to storage behaviours is avoided.

The hook concept of SPEC permits the customisation of the latter two steps in such manner that SPEC respects the ECS order queue and cooperates by exchanging logging information.

Virtually SPEC is divided thereby into a (G)UI and a scan server and becomes a simple server which might trigger other devices. The logical division is not visible for the user but technically the command line UI from SPEC might be used for any scan server and vice versa a separate GUI might drop a SPEC scan server request.

SUMMARY

At ANKA we have set up a new smart structured measurement process (ECS). The new at ANKA introduced smart concept offers the base for the implementation of the desired automated overall data management with minimised efforts. The implemented ECS manages in a first approach an order queue (FIFO) of experimental demands. Currently the ECS is collecting metadata in a XML file added to the experimental results.

The ECS itself was introduced into the well time-tested SPEC measurement environment without breaking accepted habits and SPEC was improved to act in the TANGO environment as a virtually divided cooperating UI and scan server.

- [1] TANGO, www.tango-controls.org/.
- [2] R. Stotzka, W. Mexner, T. Dos Santos Rolo, H. Pasic, J. van Wezel, V. Hartmann, T. Jejkel, A. Garcia, D. Haas, A. Streit, "Large Scale Data Facility for Data Intensive Beamlines", Proceedings of the 13th International Conference on Accelerator and Large Experimental Physics Control Systems, (2011): p. 1216 1219.
- [3] HDRI, http://www.pni-hdri.de/
- [4] SPEC, http://www.certif.com/.
- [5] T. Spangenberg, K. Cerff, W. Mexner, "MACRO PACKAGE BASED ENHANCEMENT OF SPEC CONTROLLEDEXPERIMENTAL SETUPS", PCaPAC 2010, Saskatoon, Oct. 2010, THPL015, p 184 (2010), http://epaper.kek.jp/pc10/papers/thpl015.pdf
- [6] PVSS2, http://www.pvss.com/.
- [7] T. Spangenberg, K. Cerff,W. Mexner V. Kaiser, "Tango Integration of a SIMATIC WinCC Open Architecture SCADA System at ANKA, "Proceedings of the 13th International Conference on Accelerator and Large Experimental Physics Control Systems, (2011): p. 749-752.

POST-MORTEM ANALYSIS OF BPM-INTERLOCK TRIGGERED BEAM DUMPS AT PETRA-III

G.K. Sahoo[#], K. Balewski, A. Kling DESY, Hamburg, Germany

Abstract

PETRA-III is a 3rd generation synchrotron light source dedicated to users at 14 beam lines with 30 instruments. This operates with several filling modes such as 60, 240 and 320 bunches with 100mA or 40 bunches with 80mA at a positron beam energy of 6 GeV. The horizontal beam emittance is 1nmrad while a coupling of 1% amounts to a vertical emittance of 10pmrad. During a user run unscheduled beam dumps triggered by Machine Protection System may occur. In many cases the reason can be identified but in some it remains undetected. Though the beam is lost some signature is left in the ring buffers of the 226 BPM electronics where last 16384 turns just before the dump are available for post-mortem analysis. Scrutinizing turn by turn orbits and the frequency spectrum measured at a BPM can improve understanding of such a beam loss and may help to increase the efficiency of operation by eliminating the sources. Here we discuss in detail the functionality of a Java GUI used to investigate the reasons for unwanted dumps. In particular, the most effective corrector method is applied to identify correctors that might have perturbed the golden orbit leading to violations of the interlock limits.

INTRODUCTION

PETRA-III [1] is a 3rd generation synchrotron light source commissioned with positron beam energy of 6 GeV and 100mA stored current at betatron tune values (36.12, 30.28). The horizontal beam emittance is 1nmrad while a coupling of 1% amounts to a vertical emittance of 10pmrad. The machine is dedicated to users for experiments from 14 beam lines with 30 instruments. This operates with several filling modes, such as 60, 240 and 320 bunches with 100mA or 40 bunches with 80mA. During the normal user operation, there are unscheduled beam dumps triggered by Machine Protection System (MPS) [2, 3]. These triggered dumps may be before or some times after the loss of beam. The loss of beam after the beam dump by MPS is understood as the reason as implied by the protection system. But the loss of beam before the beam dump by MPS or sudden fall of beam current and subsequently filled up by top up are unexpected. In these cases the reason is not identified or some cases it is undetected. But, though the beam is lost, it left some signature on its post-mortem data. From these data proper scrutinization of turn by turn orbits and the frequency spectrum measured at a BPM can improve understanding of such a beam loss and may help to increase the efficiency of operation by eliminating the sources of disturbance. For these purposes, there are 226 Beam Position Monitors (BPM) distributed in 2303.952m ring to monitor the transverse orbits by Libera System[4]. These BPMs are connected with a Ring Buffer where continuously 16384 latest turns of data for each BPM is stored in Libera. When the Libera server receives a beam dump signal from MPS, it dumps 16384 turns of orbit data for each BPM to an Archive Server with an event time stamp for post-mortem analysis. It is worth to mention that the revolution frequency of PETRA III is 130.121 kHz. Though there are 226 BPMs, all are not activated to trigger a MPS signal. The BPMs in damping wiggler sections in West and North, the BPMs in DBA sections are activated with lower and upper limit of orbit deviations in transverse plane. If the orbit deviations are beyond these limits at any of such special BPM then Libera initiates an interlock event to MPS to fire a beam dump signal. As mentioned above the post-mortem data not only contains orbits but includes these interlock orbit bounds and the Σ -signal of beam intensity. These postmortem data are huge and contains a lot of information which we want to extract and analyze in this paper using a Java Graphics User Interface (GUI) Web application.

The GUI is developed in Java. It requires JRE 6+ or Java Web Start 1.5+ to run with a memory of 1024MB. It uses FORTRAN subroutine MICADO[5] of CERN library of MADx program for its most effective orbit corrector [MEOC] method. The FORTRAN subroutine is called via a Java Native Interface (JNI) written in C. To run in every computer, where FORTRAN library routines are not available, this code temporarily copies FORTRAN runtime library files (.DLL) into TEMP directory of the user. The post-mortem data are downloaded iteratively from the Archive Server [6-8]. Due to heavy demand on this server, sometimes maybe bit difficult to retrieve data. If occasionally null data is retrieved then one has to read the event once again.

Here we describe the beam dump events for the tripping of corrector magnets, wrong setting of power supplies, failure in RF system temperature regulations, unclosed injection kicker bumps etc. The main features are looking into the frequency spectrum [9] of turn by turn orbits of a particular BPM; 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 transient malfunction of a magnet, the orbit will grow and surpass the interlock limits at some special BPM and the beam will be dumped by MPS. In post-mortem analysis this change in orbit can be corrected by a few correctors using MICADO to investigate the cause of beam loss in transverse plane.

[#] Gajendra.Kumar.Sahoo@desy.de



Figure 1: The graphics user interface GUI-MEOC used for Post-Mortem Analysis of BPM-Interlock Triggered Beam Dumps.

GUI-MEOC

The Java GUI-MEOC as shown in Figure-1 has 3major panels along with the menu bar at the top. They are the Graphics Panels in the left, command Button Panel in the middle and information Table Panel at the right. These are briefly discussed below.

The Table Panel contains information about the 226 BPMs, 687 horizontal correctors, and 608 vertical correctors. These are real or artificial correctors assembled to analyze the cause for beam loss. For this purpose the dipoles, quadrupoles, and sextupoles are split into two parts and an artificial corrector is inserted in between.

In the left there are 5 Tabbed Panes that display the graphics for four different types of observations. These tabbed panes are:

Orbits: This displays the transverse orbits (x, y) for selected BPM and selected range of turns. Once you are in this pane and want to show the orbits for another BPM. Then select the BPM and click '**Update Plot for (Orbits/Sum Signal)**' button.

Sum Signal: This shows the \sum -signal (sum of BPM 4 plate voltages) of the selected BPM in the top and the transverse orbits below. With this you can observe the orbits as a function of sum signal. Once you are in this pane and want to see the orbits for another BPM. Then select the BPM and click 'Update Plot for (Orbits/ Sum Signal)' button.

Orbit Correction: Mostly it is blank as you select it. This pane is automatically selected for most effective orbit corrector method and the corrected and uncorrected orbits are displayed.

Frequency Spectrum: As you observe the orbits in the first pane. In the back ground the frequency spectrum is calculated and plotted in this pane. One can observe the spectrum only after the first pane is clicked.

Archive Interlock Events: This pane is open in the beginning of initialization of the GUI. It displays the recent interlock events occurred for last 7 days including the current day. By default the most recent event is selected in the Event List Table of 'Select an Interlock Event' Panel. Now select the event of your interest by

clicking over the list. Click the button '**Read data for** Selected Event' to read post-mortem data from Archive Server.

In the middle, the Button Panel has four subpanels such as Correction Panel, Turn by Turn Panel, Update Panel and Axes Control Panel. They are briefly described here.

Correction Panel: This Panel is mostly used for MEOC. The button 'Save Reference Orbit', once clicked, saves the reference orbit for the turn number as mentioned in the text field of 'Reference'. The button 'Save Perturbed Orbit', once clicked, saves the perturbed orbit for the turn number as mentioned in the text field of 'Perturbed'. nCx and nCy are the number of correctors to be used for MEOC in horizontal and vertical planes respectively. In the left, there are choices for horizontal or vertical plane for orbit correction. By default both are selected. One can select or deselect for particular plane. Once the reference orbit, perturbed orbit, No. of correctors and plane are selected, it is ready for orbit correction. Click the button 'Execute Correction'.

Turn by Turn Panel: This Panel is used for turn by turn data analysis. The current turn number may be entered by typed in or be selected by clicking in the chart. The turn by turn orbit is displayed by clicking the button **'TbyT Show Orbit'**. If the selection box **'Difference Orbit'** is ticked, then the difference orbit between the reference orbit and the current orbit is displayed.

Update Panel: This Panel is used for update of transverse orbits or sum signal display. To navigate to another BPM you may click the button '**Update Plot for (Orbits/Sum Signal)**'. As you know the Fast Fourier Transform (FFT) works for 2^{N} data points. To keep the data points to this order, once the upper limit of turn number is selected, you may use Combo Box to select the lower limit of turns so that it will lie within 16 to 16384.

Axes Control Panel: This last portion of the Panel is used for graphics axes control. In addition, the Menus in the Menu bar extend some additional jobs that necessary for users. '**Show Poincare Map**' shows the Phase Space plots at the center of undulator sections for all the DBA cells for 16384 turns or for a selected range of turns.

RESULTS AND DISCUSSIONS

Tripping/Wrong Setting of a Magnet

The orbits are corrected using slow orbit correction [10] with SVD using 191 horizontal correctors and 187 vertical correctors employing 226 BPMs. In normal operation, the golden orbits are maintained with Fast Orbit correction using SVD with 40 fast correctors on either plane. During the process of correction some correctors may set higher currents than desire leading to high orbit oscillations. This can also happen due to improper setting of other magnets such as quadrupoles and sextupoles etc. A small change in set current can be treated as an effect of an artificial corrector incorporated in it. So, in case of unknown beam dumps, the change in orbits in post-mortem data may be corrected choosing a

few numbers of correctors. The MEOC is utilized to investigate the suitable corrector that might have perturbed the orbit beyond the interlock limits. For example, the event (Sun Dec 04 09:18:13 CET 2011) was due to the failure of the vertical corrector magnet PKVSX_SR_82 which was setting wrong values that lead to beam dump. You can see from Figure 2 that the difference vertical orbit was well corrected to zero using the same corrector.



Figure 2: Vertical orbit correction for the Interlock Event on Sun 04 December 2011 at 09:18:13 which indicates PKVSX_SR_82 vertical corrector as the source of disturbance.

Regulation Problem & Tripping of RF System

As per statistics the major beam dump is associated with RF system of PETRA III, though gradually it is delivering more stable operations. We have seen drop in beam current during top up mode as shown in Figure 3(a,c).Sometimes it is refilled with top up and sometimes not, rather leads to self-triggered beam dump. This may be due to not proper regulations in temperature control of cavities. We have seen the frequency in spectrum analysis of orbits during such a failure. As shown in Figure 3(b), we clearly see the dispersion orbit in arc sections looking to the difference orbit in the trailing edge of BPM sumsignal, indicating the failure of RF system when all magnet power supplies are on.



Figure 3: (a)The effect on sum signal and the transverse orbits during the unregulated temperatures of the cavities in the RF system; (b) The dispersion orbit is produced as you look into the difference orbit during the failure of RF system while all magnet power supplies are on; (c) frequency spectrum at a BPM.

Unclosed Injection Kicker Bump

The machine operates in top up mode with 1% current variation. Depending on number of bunches filled, the beam lifetime varies and as a result the beam is injected at different time intervals. It is observed that due to some unforeseen reasons the injection bump is not closed in some rare cases. This led to beam dump due to large oscillations in orbit in the injection plane as shown in Figure 4(a).

Drifting of a Quadrupole Magnet Power Supply

In rare cases the read back current of quadrupoles or other magnets drift from their corresponding set values resulting gradual drift in orbits leading to MPS beam dump. In Figure 4(b,c) one can observe the behavior of quadrupole QA4_OL_61 and the dipole magnets in new DBA sections respectively.



Figure 4:(a) unclosed injection kicker bump; (b)the transient behavior of QA4_OL_61;(c) 600Hz oscillations on power supplies of new DBA octant dipole magnets.

CONCLUSION

This Java GUI Web application is used for post-mortem analysis of MPS beam dumps for PETRA III. It reproduces the source of disturbance accurately for simulated beam dumps by setting higher currents in corrector magnets. This may be intuitively used to apply for other unknown cases, where a corrector set wrong higher value temporarily and come back leading to a beam dump. Same may be applicable for quadrupoles, sextupoles etc. This program is also utilized to trace the orbit perturbation and frequency spectrum left behind by RF system, injection kicker, quadrupoles and corrector magnets which has enabled us to remove the cause of such perturbations for optimal operation of PETRA III

- K. Balewski, W. Brefeld, et al, "PETRA III: A new high brilliance synchrotron radiation source at DESY", EPAC-2004, Lucerne, 2004, pp.2302-2304.
- [2] T. Lensch, M. Werner, "Machine Protection System for PETRA III", Proceedings of DIPAC09, Basel, Switzerland, pp 351-353, 2009.
- [3] http://ttfinfo.desy.de/petra/show.jsp?dir=/2011/17/29.04_a &pos =2011-04-29T15:26:23
- [4] Instrumentation Technologies, Slovenia; http://www.i-ech.si/
- [5] B. Autin and Y. Matri, "Closed Orbit Correction of Alternating Gradient machines using a small number of magnets", CERN/ISR-MA/73-17, CERN, 1973.
- [6] P. Duval, M. Lomperski, "TINE Event Archive System".
- [7] H. G. Wu, Private communications.
- [8] R. Bacher, P. Bartkiewicz et al, "TINE: The Control System for PETRA3 and DESY preaccelerators", ICFA Beam Dynamics Newsletter, No.47, December 2008.
- [9] Michael Thomas Flanagan's Java Scientific Library, http://www.ee.ucl.ac.uk/~mflanaga/java/
- [10] G.K. Sahoo, K. Balewski, et al, "Closed orbit correction and orbit stabilization scheme for the 6 GeV synchrotron light source PETRA III", EPAC-2004, Lucerne, 2004, pp.2302-2304.

DESIGN AND IMPLEMENTATION OF LABVIEWTM BASED GUI FOR REMOTE OPERATION AND CONTROL OF EXCIMER LASER FOR PLASMA WAKEFIELD ACCELERATION EXPERIMENT

K.K. Mohandas, K. Mahavar, [#]R.A.V. Kumar, IPR, Gandhinagar, India

S. Joshi, A. Sharma, Nirma University, Ahmedabad, India

Abstract

The paper describes the development of a GUI based control software for control/operation, maintenance and data logging of a Coherent CompexPro 102 Excimer Laser (ArF, 193 nm) using LabViewTM instrument control software.

INTRODUCTION

The Coherent COMPexPro102 Excimer laser (Fig.1) is an ArF gas based UV laser operating at a wavelength of 193 nm. This laser uses Ar, F2, Ne, and He gases for its operation. The laser can deliver a maximum energy of ~ 200 mJ at 15 ns pulse width and capable of operating at a pulse repetition rate of 1-20 Hz. The laser is normally controlled using a tethered control panel which has a cable length restriction of around 3 meters. In order that the laser can be controlled and monitored remotely, it is essential that a GUI based interface be developed. The block-diagram of system showing the gas feed system, the I/O system and the energy meters is shown in Fig.2.

Currently, the operation of the laser is carried out manually using the wired keypad. The main motivation of the development of this GUI is for easier user convenience as well as for future use when the laser will be used to generate lithium photo-ionized plasma for the plasma wakefield accelerator experiment system. In the experiment, the laser operation will be carried out remotely and the laser system will be integrated with the accelerator system operations *via* LabViewTM instrument





Figure 1: The Excimer laser system.

control software. The GUI of the laser will be integrated with the GUI's of other components of the experiment



Figure 2: Schematic diagram of the excimer laser system showing the gas lines, I/O interfaces and energy calibration layout.

such as accelerator operation, precision heater controls for the plasma chamber, vacuum /gas fill systems as well as data acquisition equipment, thus providing a single window control of the whole experimental system. Routine laser operation (setting of parameters and running of laser), maintenance operations like gas fill and energy calibration, logging of laser parameters etc., which are currently being carried out manually will be automated using the GUI.

This paper presents the development of the module for the excimer laser, *i.e.*, the required LabVIEWTM virtual instrument modules. GUI interfaces and the required display screens for laser control, parameter display, gas fill operation as well as documentation/logging of the operational parameters of the laser system. The paper also presents the development of a completely automated module for calibration of the internal energy meter of the excimer laser with a calibrated external laser energy meter (Coherent FieldMax-II). This procedure is necessary as the performance and calibration of the internal energy meter which is used by the laser system to stabilize its laser output can vary over a period. This oneswitch, automated operation provides a linear-fitted calibration of the internal energy meter with that of a precalibrated commercial energy meter. This procedure is

usually carried out after each gas fill to make sure that the health of the laser system is good.



Figure 3: Flow-chart for Gas-Fill Operation.

The I/O control of the laser is established via a standard RS-232 interface coupled to the PC through a RS-232-to-USB interface. LabVIEWTM tools like web-publishing Tool, structure palette, string, VISA palette have been used to design this graphical user interface.

The laser is designed to be controlled either from the control panel or from a remote PC but not from both at the same time.

Field-MAX-II-P energy meter with an energy sensor optimized and calibrated for use with 193 nm UV laser is used for external energy measurement. The UV sensor incorporates a DUV quartz diffuser for increased coating damage resistance. The energy meter has USB I/O connectivity and hence could be seamlessly integrated into the laser GUI.

MODULE FOR GAS-FILL OPERATION

Excimer Laser uses a mixture of various gases depending on the wavelength of operation of the laser. The 193 nm operation of the laser uses he, Ne, Ar and F gases in various proportions which are decided by the laser system at the time of gas filling operation. The gas



Figure 4: Front-Panel for Flushing gas lines.



Figure 5: Front-Panel for New fill and Purge line.

fills have to be carried out periodically so as to maintain the laser at its operating energy. The flow-chart of the gas-filling process based on which the GUI has been designed is shown in Fig.3. In this GUI, gas flushing, new-fill, purge-gas line and data logging are automatically carried out.

Front Panel for Flushing gas Lines

Gas flushing is an essential step to clear the gas lines of all impurities before the actual gas fill operation is initiated. When this operation is initiated from the "Flush Line" GUI, all the gas lines will be flushed using helium gas and since Fluorine is a corrosive gas, that line is flushed twice as a precautionary measure. The image of the GUI for Flush Line is shown in Fig.4.

Front Panel for New-Fill and Purge Line

This GUI assists in the gas-fill procedure. Once the "NEWFILL" operation is initiated, the GUI will prompt the user to open or close appropriate gas valves (MV1/SV1 to MV4/SV4) and go through the process of first evacuating the gas lines and the laser cavity and then filling all the





Figure 7: Flow chart for monitor and set laser parameter.

required gases at appropriate pressures into the laser system. Currently, these gas valves are manually operated. In future, the whole operation could be fully automated if the manual valves are replaced by electrically controlled valves. Once the gas fill is over, the GUI will automatically flush the gas lines and purge the lines with inert gas for safety. The GUI for the gas fill is shown in Fig.5. Post gas-fill parameters of the system are shown via GUI shown in Fig. 6. This also logs the data to the log file.

Front Panel for Display of Laser Maintenance Parameters

Other system maintenance parameters such as gas pressures, filter ratio (lifetime of the halogen filter), total laser pulses, pulses after refill, tube temperature etc can be displayed.

By using the "STARTRP" command, these parameters can be logged into an Excel file for laser maintenance purposes. The flow chart for this GUI is shown in Fig.6.

MODULE FOR LASER MONITOR AND SET PARAMETERS

This GUI is basically to display the various operating and set parameters of the laser. Parameters such as laser energy, laser repetition rate and mode of operation (Constant energy or constant voltage) can be set. Also, there is an option to activate the burst-mode operation of the laser if desired so by the user. Flowchart for this GUI is shown in Fig.7. Front-panel for this is shown in Fig.8.

There are four basic operating modes under "Select Mode" *viz*, HV-PGR, HV-NGR (constant voltage mode with and without gas purge), EGY-PGR, EGY-NGR, (constant energy mode with and without gas purge) for setting the running mode for the excimer laser. At a time, only one of the four options can be activated and can be changed only after laser is in off mode.

MODULE FOR LASER ENERGY CALIBRATION

The internal energy meter of the excimer laser taps a small percentage of the laser beam to monitor the beam energy. This is used for stabilizing the laser output. Over time, the calibration of this internal energy meter deteriorates and needs to be corrected over time. The calibration of the internal energy meter can be done against a pre-calibrated external energy meter. By plotting the outputs of both the internal and external energy meters at various operating voltages, one can determine if the internal energy meter is correct. If the calibration is found to off by more than an acceptable percentage, then the internal energy meter needs be physically recalibrated. The flowchart for energy calibration is shown in Fig.9. The "START CALIBERATION" process requires the outputs of both the laser and the external energy meter to be read, the data averaged and standard deviation determined for



Figure 8 : Front-panel of Monitor and laser Set parameter.



Figure 9: Flow-chart for Energy Calibration.



Figure 10: Front-Panel of Energy-Calibration GUI.

various voltage levels. The averaged data is plotted in the graph *in-situ*. After the process ends, the data is saved in Excel form, and finally plotted against laser high voltage, as shown in Fig 10. The data points are then linearly fitted to obtain slopes, which are then compared to determine if the calibration of the internal energy meter is within acceptable limits. The internal statistical math modules of LabVIEWTM are used for this. The logged data in Excel format is shown in Fig.11.

Time	HV	Eia	Linfit E _{ex}	En	Linfit E _{is}	Slope E _{ia}	Slope E _{ex}	STD En
H:M:S	kV	mJ	mJ	mJ	mJ			
11:13:52	24	103.13	104.18	109.43	108.85	14.65	14.53	7.35
11:15:50	25	123.2	118.83	113.64	123.39	14.65	14.53	8.15
11:17:32	26	137.42	133.48	129.25	137.92	14.65	14.53	7.20
11:19:46	27	155.79	148.12	151.34	152.45	14.65	14.53	7.45
11:21:52	28	168.85	162.77	163.17	166.99	14.65	14.53	6.43
11:23:43	29	181.06	177.42	178.28	181.52	14.65	14.53	7.97
11:25:52	30	193.73	192.07	191.75	196.06	14.65	14.53	7.35

Figure 11: Data for energy calibration saved in excel file.

LABVIEWTM DRIVERS FOR EXCIMER LASER

In this Virtual Instrument driver (VI) file, all the Sub-VI drivers for different commands related to the excimer laser operation have been developed and incorporated. The VI-Tree showing the various VI's generated for compiling this GUI depicted schematically in Fig.13.

Flush Command

There are four "Flush" commands are used to flush gas lines of the various gases used for operation of the excimer laser (Halogen, Buffer, Rare, and Inert gases).

Write Command

The four sub-routines under this VI are set/control functions related to running of the laser *ie.*, "Run Laser", "Stop Laser", "Change value of HV", "Repetition Rate".

Read Command

All parameters (like operation-mode, mode, repetitionrate,HV, Energy, Pressure, System time and date, tube temperature, etc.) of excimer laser are read by running this VI.

LOGICAL SUB-ROUTINES

There are five sub-routines under this VI. Password sub VI is used for the security purpose. Call VI is used for to call VI from folder where it has been saved. Regression is used to find relation between two parameters by linear fit.

Mathematics

These sub-VIs are used for to estimate mean, min, max, standard deviation and slope for data in the energy calibration operation.

Save file

All data logging operations are carried out by this VI.

Multi Y-axes Graph

All multi X-Y graphical representations are carried out by this VI.

Steps for Gas-fill

All user messages related to step-wise explanation/confirmation of the manual part of the gasfill operation are displayed to the user when this VI is run.

MODULES FOR EXCIMER LASER

There are four sub VI modules (like Port Initialization, Gas-Fill operation, Monitor and set parameter and Energy calibration) for excimer laser which have been designed under this module.

CONCLUSIONS

GUI for port initialization, setting and monitor all parameters of excimer laser, gas fill operation and energy calibration for excimer laser are designed by serial interface using RS-232, DB-9 connector. But, all parameters of excimer laser are monitor and set with one second or two second delay. Virtual instrument of Excimer laser and external energy meter is designed using LabVIEW[™] software. Real-Time Data acquisition will be done in various file like .doc,.pdf,.xls,.txt etc. All laser operation will be done on internet in LabVIEW[™] environment if all computers are connected



Figure 12: View of the plasma accelerator laboratory showing the ArF laser and the plasma accelerator chamber.

in LAN.Automatic programming will be design for Laser operations and it will be change. Laser operation will be become faster and safe than it operate manually. Laser



Figure 13: The V-I tree for the Excimer laser.

operations are easy to operate and understand by user. Laser Data will be transferred for long distance using RS-232 to RS-485 converter.

ACKNOWLEDGMENTS

The authors wish to thank Mr. Yogesh Yeole and Mr. Jinto Thomas for various help rendered with LabVIEWTM.

REFERENCES

[1] John Park, Steve Mackay, and Edwin Wright, "Practical Data Communications for Instrumentation and Control," Elsevier (2003).

[2] National Instruments LabVIEWTM 2010 –software.

[3] Jovitha Jerome, "Virtual Instrumentation using LabVIEW", PHI Learning Private Limited (2010).

[4] Hexin RS-485 to RS-S232 interface converter Model-485.

[5] Coherent Interfacing manual for COMPEXpro (RoHS).

[6] User Manual for Coherent FieldMaxII-P Laser Energy Meter.

[7] Béla G. Lipták, "Process Software and Digital Networking" Instrument Engineer's Handbook, 3rd edition, CRC press, (2002).

[8] User Manual, Coherent Energy Max Sensors.

STARS ON ANDROID

T. Kosuge, Photon Factory, KEK, Tsukuba, Japan

Abstract

STARS (Simple Transmission and Retrieval System) [1, 2] is a message transferring software for small-scale control systems with a TCP/IP socket, and it works on various types of operating systems. STARS is used as a beamline control system for controlling the optical devices (mirrors, monochrometers, etc.) for beamlines at the Photon Factory.

We have succeeded in running a STARS GUI client on Android using the STARS Java interface library. This achievement has brought with it the capability of developing a user-friendly GUI terminal using smartphones or tablet devices. Such a GUI terminal will help beamline users check movement near optical devices.

OVERVIEW OF STARS

STARS consists of a server program (STARS server) and client programs (STARS clients). Each client is connected to the server through a TCP/IP socket, and communicates using text-based messages (Fig. 1).



Figure 1: Message transfer between STARS clients and a STARS server.

Each client program has its own unique node name, and it sends text-based messages using the destination node name to the server, which then delivers the messages to the destination client. Through this extremely simple solution, STARS is able to provide basic control system functionality.

The STARS server program was written in Perl, and it can therefore run on various operating systems.

BEAMLINE CONTROL SYSTEM USING STARS

STARS has been installed in more than 20 beamlines of the Photon Factory as a beamline control system.

Before STARS was developed, the beamlines of the Photon Factory used various control systems (e.g., the originally developed software, LabVIEW, or SPEC) and the hardware was controlled using a software directory. At that time, staff members had to prepare their own hardware driver for each control system. Since the installation of STARS, however, hardware drivers are now developed by a "beamline control group," and the beamline control system using STARS provides a common interface to GUI programs, etc., (Fig. 2) for handling the beamline components. This interface can also be accessed by various data acquisition systems.



Figure 2: Common interface for handling beamline components using STARS.

Several types of driver software for beamline devices and common GUI programs have recently been developed at the Photon Factory.

STARS JAVA INTERFACE FOR ANDROID

STARS client programmers are required to use TCP/IP sockets and handle text-based messages, and although beginners may find it difficult, it is very easy for programmers with prior knowledge of TCP/IP socket programming to develop a STARS client. STARS is equipped with interface libraries for certain programming languages (Perl, VB, C# [3], C [4], and Java). Programmers need not be concerned with TCP/IP socket programming using these interface libraries. We modified a few parts of the source code for the STARS Java interface for Android.

DEVELOPMENT OF CLIENTS USING STARS JAVA INTERFACE FOR ANDROID

Development Environment

We used Eclipse for the development of the STARS client program for Android. Eclipse requires the

installation of the Android SDK and the Android Plugin. Figure 3 shows a STARS client for Android developed using Eclipse.



Figure 3: Development of STARS client for Android using Eclipse.

Methods

The STARS Java interface for Android was ported from the original STARS Java interface. The same methods used in the original interface were also used in the ported interface, although UI (User Interface) functions, such as Form Widgets and Text Fields, were handled differently. This difference is described further in the Callback Function section.

Before employing the STARS Java interface methods, the following objects need to be defined:

import com.example.stars_control_panel.StarsInterface; import com.example.stars_control_panel.StarsCallback; import com.example.stars_control_panel.StarsException; import com.example.stars_control_panel.StarsMessage;

static StarsInterface stars

= new (myNodeName, starsServerName, keyFileName, starsPort);

In the above program, "stars" is used as the object name, whereas "myNodeName," "starsServerName," and "keyFileName" are string values, and "starsPort" is an integer value. In addition, "keyFileName" and "starsPort" are omissible, in which case default values are used.

Connect

The "connect" method is used for establishing a connection. This method executes the keyword checking procedure of STARS automatically, and throws an exception if a connection is not established. An example of the "connect" method and error handling is given as follows:

try{

stars.connect();
}catch(StarsException se){
 //Error handling
 viewPresent.setText(se.toString());
}

Send

The "send" method is used to send messages to the STARS server in the following manner:

//Send "GetValue" command to a node name "Dev1".
stars.send("Dev1 GetValue");
//or
stars.send("Dev1", "GetValue");

The "send" method throws an exception, in which case the "try" and "catch" functions must be used.

Receive

The messages received by the client program can be read from the receive buffer using the "receive" method given below:

StarsMessage rcvMsg = stars.receive(timeOut);

"timeOut" is an integer value (in m seconds) and is omissible (default: 5000).

"StarsMessage" has the following fields and methods:

- The "getAllMessage()" method is used for receiving all messages.
- The "from" field is the message source node name.
- The "to" field is the destination node name.
- The "command" field retains the command part of the message.
- The "parameters" field retains some of the parameters used in the message.
- The "getMessage()" method returns "command" and "parameters."

Callback Function

When a message arrives from the STARS server, the function set by the programmer is called using "startCallbackHandler," an example of which is shown below.

public void onCreate(Bundle savedInstanceState){
 super.onCreate(savedInstanceState);

setContentView(R.layout.activity_stars__control__panel
);

//Set callback "StarsMessageHandler()" will be called
// when a message arrives from the STARS Server.
handler = new Handler();
StarsCallback Cbh = new StarsMessageHandler();

try{

stars.starsCallbackHandler(Cbh);

Proceedings of PCaPAC2012, Kolkata, India

}catch(StarsException se){ //Write error handling codes here.

//Handle messages.
//Handle messages.
class StarsMessageHandler implements StarsCallback{
 public void starsCallbackHandler(StarsMessage st){
 if(st.command.equals("@GetValue"){
 //Write message handling codes.
 handler.post(new Runnable(){
 public void run(){
 viewPresent.setText(curVal.toString());
 };
 };
 });

The "Callback" function of the STARS Java interface is delivered using a "thread." However, the Android UI cannot be handled from another thread. One solution to this issue is creating a "Runnable" object and then calling a "post" method. Thus, this is the difference between the STARS Java interface and the interface for Android.

MOTOR CONTROL PANEL GUI FOR SMART PHONES

We developed a Motor Control Panel for smart phones, which is anticipated to will become a common beamline GUI at the Photon Factory. Figure 4 shows a design view of the Motor Control Panel on Eclipse.

(ML - STARS_Control_Panel/r	es/layout/activity_stars_control_	panel.xml - Ecipse SDK		Con Con
Edit Refactor Navigate	Search Project Run Window	Help		
3 • 13 16 16 18 19 19	9 • 🕾 🌿 🖾 🏚 • O • (🎍 🕶 🛷 🕶 💽 XML File 🤌 👻 🤃 🕈 🗇 🔹		
		Quick Access	🐉 Java 🕸 Debug 👷 Java Browsing 🚳 E	DOMS X
Project Expl 12 😑 🖽	🕑 STARS_Control_Paneljava 🔡	activity_stars_control_panel.xml 😒	• B	11 -
E 😨 👻	• Palette		Structure >	0.0
STARS_Control_Panel	Palette •	default • Nexus One • 💌 •	13 Outline	-
a 🐸 src	i Form Widgets	* AppTheme • @ STARS_Control_Panel •	· Fill Relations month	Þ 😫
a tt com.example.st	Territore Large Medium tout Better Small		The best family "CTADE Cases	1
D STAKS_CORE	Off. Contra Chatching	S - 🖷 10 -	M aditanti	
 D Starscalloac C StarsCalloac 			To text/iew2 - "1/0 Client parts	
Starsteterfa	CheckedTextView Spinner •		T editText2	
StarsMessan	00	V 🚺 tot aft	* ex button2 - "Disconnect"	
> an iGenerated Jay		STARS_Control_Panel align®	os button1 - "Connect"	
EDK 2.0 [Android 2		STARS Server Host Name aligned	Ab textView3 - "Present"	
> Android Dependenc	CuickContactBadge		A textView4 - "Present"	
assets	000	I/O Client name	An textView5 - "Absolute"	
🖻 🍰 bin			T editText3 +	
🔺 😓 lbs	TTTTT		* +	
android-support		Conpect Discopnect	📼 Properties 🏇 🍂 🔣 🗃 🖻	
# 🍰 res			Height wrap_content *	
o 😂 drawable-hdpi		Present Present	B Margins []	
o 🗁 drawable-ldpi		Abrolute	Style buttonStyle	
b 🗁 drawable-mdpi	Text Fields	Set .	Text @string/button_c	
o 🦲 drawabie-xndpi			Conten	
a tayout	Composite	Relative	TextView	
to Chi manu	Composite	- +>	Text @string/button_c	
> so values	Co Time & Date		Hint	
AndroidManifest.xn	Time & Date	Stree	Text @android:color/pr	
ic_launcher-web.pn	Transitions		Text III @android:color/	
proguard-project.to	Advanced		 Text ?android:attr/text 	
project.properties	Custom & Library Views		Text	
STARS_Control_Pai	🔄 Graphical Layout 📡 activity_	stars_control_panel.xml		
TestPanel				
	- Documentation III Properties	Console 33		
	DDMS			
- H	4		b	٠

Figure 4: Design view of the Motor Control Panel.

Figure 5 shows a photograph of the Motor Control Panel running on a smart phone (Sony Ericsson Xperia). The Panel connects to the STARS server through a wireless LAN and communicates with clients using a stepping motor controller. Beamline users can remotely control the stepping motor (Fig. 6).

STARS Server Host Name spinet office2.kek/p pm16:08.dth1 Context and pm16:04.dth1 Present 23456 Absolute 23456 server 23456 server 2345	STARS O	ontrol Pane	8.4	a 🗰 15:4
spinet-office2.kek/p 2/0 Client name pm16c04.dth1 Correct 2/3456 Absolute 2/3456 ar Absolute 2/3456 ar Absolute 2/3456 Absolute 2/3456 Absolute 2/3456 Absolute 2/345	STARS S	erver Hos	it Name	
VO Client name pm16c04.dh1 Connex Durdment Present 23456 Absolute 23456 Relative Stop	spinet	office2.k	ek.jp	
pentleck4.dth1 Centre: Bucketer Present 23456 23456 set 23456 set 2356 set	I/O Clier	nt name		
Conco Decement Present 23456 Absolute 23456 se Relative and an	pm160	04.dth1		
Present 23456 Absolute 23456 5 Relative Stop			Connect	Disconnec
Present 23456 Absolute 23456 Relative Stop				-
Absolute 23456 se Relative Stop	Present	2	3456	
23356 se Relative Stop	Absolute			
Stop.		23456		Se
en e	Relative			
Stop	e.			42
Stop				_
Stop				
			Stop	
9 6 E	9	4-	G 	1

Figure 5: Motor Control Panel on a smart phone.



Figure 6: Stepping motor controller operated by the Motor Control Panel through STARS.

CONCLUSION

We ported the STARS Java interface to Android, which was demonstrated to work satisfactorily. We then developed a Motor Control Panel for smart phones. The development of a GUI for an Android tablet is also possible using a similar method.

The STARS Java interface for Android and the GUI for smart phones or tablets will be useful tools on STARSbased control systems.

- T. Kosuge, Y. Saito, "RECENT PROGRESS OF STARS" Proceedings of PCaPAC2005, Hayama, Japan, 2005.
- [2] http://stars.kek.jp/
- [3] T. Kosuge, "STARS .NET INTERFACE FOR WINDOWS CE", Proceedings of PCaPAC2008, Ljubljana, Slovenia 2008.
- [4] T. Kosuge, K. Nigorikawa, "STARS ON PLC" Proceedings of PCaPAC2010, Saskatoon, Canada, 2010.

DEVELOPMENT OF EPICS CHANNEL ACCESS EMBEDDED ACTIVEX COMPONENTS FOR GUI DEVELOPMENT

A. Roy[#], R. B. Bhole, S. Pal, VECC, Kolkata, India

Abstract

The paper describes the integration of Experimental Physics & Industrial Control System (EPICS) Channel Access (CA) protocol and Microsoft ActiveX technology towards developing a generalize operator interface (OPI) building facility for Windows platform. EPICS is used as the development architecture of the control system in Superconducting Cyclotron (SCC). Considering the operators' familiarity and compatibility with third party software, it was decided to use MS-Windows platform at operator interface level in SCC during commission. Microsoft Visual Basic (VB) is used on trial basis as OPI building platform to incorporate user specific features e.g. file system access for data storage and analysis, user authentication at OPI level etc. A set of EPICS Channel Access embedded ActiveX components is developed to the programming complexity and reduce ease developmental time of the OPI for Windows platform. OPIs, developed using these components and containing hundreds of process parameters, are being used reliably over a considerable period of time.

INTRODUCTION

The K_{bend} =520 Superconducting Cyclotron, under commissioning activity at the centre, is expected accelerate heavy ion beams to energy up to 80 MeV/A for fully stripped light heavy ions and about 10 MeV/A for heavy ions. The Microsoft Windows XP was considered as the best suited platform at operation console level during the commissioning phase due to its compatibility with third party software, commonly used Microsoft tools and operators' familiarity. EPICS [1], a standard opensource dual layer software tool for designing distributed control system, is adopted to implement the in-house supervisory control software in SCC. The lower layer of EPICS i.e. Input Output Controllers (IOC), are realised mostly on Linux platform of various flavours to minimise compatibility issues and maximise operational reliability against malwares. There are standard EPICS tools e.g. EDM, MEDM etc. for developing OPIs in general for Linux platform. But porting these X windows based OPIs to windows platform require third party X server e.g. Exceed, Xming etc with expertise during installation. The users, ranging from operators to physicists to system personnel, have the requirements e.g. data archiving into local file for offline / online analysis, authentication based service, integrity with third party system etc. during commissioning phase. As the overall control system was not grown up to its fullest and there are no standard facilities/methods available in the standard OPI tools to meet those custom requirements. Hence a methodology was required to be devised to meet specific requirements of this heterogeneous system. Developing OPIs for Windows platform integrated with standard Windows facilities, meeting user requirements, was the best option. The development of OPIs for Windows platform involved selection of developmental tools among the popular ones e.g. Microsoft Visual C or Visual Basic or National Instrument's LabVIEW, and Borland's Delphi and C++ Builder etc. The CA functionalities were to be incorporated for data access by integrating the CA library with developmental tool. This method, however, required us to understand each development tool's requirements for accessing C language function calls and maintaining this extra layer of code [2]. This results into longer developmental time and larger coding with associated complexity, efforts for testing and debugging for individual OPL

Microsoft ActiveX technology is chosen to creating reusable, platform-independent, distributed, objectoriented binary software components with encapsulated CA functionalities. Microsoft Visual Basic (VB) is chosen as the OPI development platform considering its object oriented structure, rich GUI library and less complicated coding style. A number of different options currently exist for building CA clients on the Microsoft Windows platform [3]. These include Easy Channel Access (EZCA) [3], ActiveX CA, Simple CA (SCA) [2], Java CA (JCA), CA Java (CAJ) [4], and calling native code in CA via C++. EZCA is chosen to implement CA functionalities. Several CA ActiveX components, commonly used in OPI, are developed and are being used to develop OPIs in considerably reduced time frame.

EASY CHANNEL ACCESS (EZCA[†])

The EPICS channel access API was designed to implement a high performance network protocol including such features as data and connection call-backs, event notifications and smart aggregation of data requests [2]. EZCA is designed to provide an easy to use interface to CA [3]. The library composed of around twenty five API functions supporting CA functionalities e.g. data access (process value, precision, control limits, graphical limits etc.), error handling, grouping, monitors, and tuning. The data types supported by the library are byte, string, short, long, float and double. As VB is selected as developmental tool, EZCA is chosen due to its readily available VB interface supporting, not full but, necessary and sufficient CA functionalities. Although the tuning parameters of EZCA i.e. timeout and retry-count, are provided for improving reliability, the default parameter values are found to be suitable for our purpose.

[#]r_ani@vecc.gov.in

[†]Thanks to Nicholas T. Karonis, ANL for EZCA support.

ACTIVEX TECHNOLOGY

ActiveX control has a number of advantages for Microsoft Windows based systems. Among other well developed tools for building them, we use VB in VECC. These components are registered with the OS. This registration tells a client application, in a standardized manner, where they are and what features they have and which makes *late binding* possible. [2]. Due to late biding feature, once an ActiveX component is selected from the list of registered components, all methods and features of the component are readily available to the developer during coding. These components can be installed / uninstalled using standard Windows facility for adding and removing programs.

Since the ActiveX components make calls across process which may take more time compared to library function call. Hence the performance issues using ActiveX components may not be suitable for some time critical applications. In our case, mostly the OPIs are developed for systems e.g. Vacuum, LCW plant, trim coil cooling system, trim coil interlock monitoring etc. involving PC based soft IOC with minimum scan period of 100 msec. Hence the OPI response time for monitoring satisfies the purpose. There are control parameters e.g. set points, binary operations (ON/OFF or OPEN/CLOSE type) which require time criticality of the order of 100 msec. Thus the performance of ActiveX control is found to be satisfactory.

COMPONENT LIBRARY

After surveying the types of control components commonly used in Graphical User Interfaces (GUIs), it is found that the components can be divided into six categories comprising of display of textual values, display of status using colours & images, display of alarms in text, set-point modifier, command issuing button and trending w.r.t. time. Therefore six ActiveX components are developed. The CA interface is minimised to a property called, pvName, for attaching an EPICS process variable to the component. A function called, init, is required to be called only once per component to initialise CA connection management, while loading the GUI for execution. As these components are developed using MS Windows components, hence other properties e.g. visibility, colour, font, border style, tool tip text etc. are incorporated by default. A detailed description of the components with embedded features is described below.

CA Text

The *CA Text* control component is used to display a channel's value textually. There are properties to specify *HiLimit*, *LoLimit* and corresponding colours for specifying alarm conditions. The display value can be formatted either in *scientific* or in *general* format with *precision* and *engineering unit* specified by user.

CA Alarm

The CA Alarm component is used to display colour

coded text information corresponding to *HiLimit* and *LoLimit* of a channel specified by user. It is developed to replicate the conventional alarm window used in control panels.

CA Image

The *CA Image* component is used to display the status (ON/OFF or OPEN/CLOSE) of field components e.g. valve, pump etc. with colour coded industry standard symbols. Each display state is represented by an image file. These may be bitmap, jpeg, png etc. The standard Windows Mouse actions are incorporated in this component for allowing user to take action.

CA Set-point

The *CA Set-point* component is used for controlling numeric value of a channel in textual format. This is required for modifying set-points in control system.

CA Button

The *CA Button* component is used for controlling state of a binary channel. This is required for ON/OFF or OPEN/CLOSE type operation of field components e.g. vales, pumps etc.

CA Trend

The *CA Trend* component is used for trending of process variable. This component has features e.g. *Auto scale, time span etc.*

SOME EXAMPLES

The ActiveX components, developed above, are used extensively for building GUIs for various subsystems of SCC, RTC and other projects. These GUIs are being used for a considerable period of time and have earned wide acceptance among the users. Some of the GUIs are shown in Fig. 1, 2, & 3 below.



Figure 1: GUI for RTC Vacuum control system.

CONCLUSION

The use of the CA enabled ActiveX components results into more user friendly, colour & textually more rich

GUIs with less development time. As the GUIs are developed using VB environment, hence other Windows feature e.g. user authentication, message box, file handling for data storage, multiple windows can be incorporated without any extra effort.



Figure 2: GUI for SCC Vacuum control system.



Figure 3: GUI for SCC LCW plat monitoring system.

Of course, an ActiveX control is not installable on other operating systems. ActiveX has now been subsumed by Microsoft .NET which defines a new object model. However, an ActiveX control can still be used directly in the .NET environment [2].

- [1] L.R. Dalesio, et al., "The Experimental Physics and Industrial Control System Architecture", ICALEPCS, Berlin, Germany, 1993.
- [2] C. Timossi, et al., "Experience with ActiveX Control for Simple Channel Access", 2002.
- [3] EZCA Primer, http://www.aps.anl.gov/epics /EpicsDocumentation /ExtensionsManuals /EzCa /EzcaPrimer.html
- [4] M. Sekoranja, "Native Java Implementation of Channel Access for EPICS", ICALEPCS, Geneva, October, 2005.

DEVELOPMENT OF FAST CONTROLS FOR BEAM WIRE SCANNER FOR SuperKEKB

A. Roy[#], VECC, Kolkata, India T. Okazaki, EJIT, Tsuchiura, Japan N. Iida, K. Furukawa, KEK, Tsukuba, Japan

Abstract

Recent development towards the data acquisition system of the wire scanner (WS) systems of the SuperKEKB injector linac (LINAC) and beam transport lines (BT's) is described. A VME based system, comprised of charge sensitive ADC (CSADC) board, scaler board, DAC board and Event receiver board, has been installed. The primary aim of the system is to utilise global linac event timing system for synchronized and mode-dependent data acquisition. A set of EPICS device driver has been developed for new hardware e.g. CSADC, scaler and DAC boards. The combination of latest versions of firmware and EPICS device driver for Micro Research Finland (MRF) Event receiver board is also evaluated and further incorporated in this system. The application software is developed for simultaneous acquisition of multiple beam mode data during multimode injection of the LINAC. The developed system is tested successfully after integrating with the existing wire scanner driving mechanism. The system enables the beam size measurements at four consecutive locations that derive Twiss parameters and ensure the reliable beam transport to four downstream storage rings.

INTRODUCTION

The KEK 8-GeV linac injects electron and positron beams with different characteristics into four storage rings: KEKB high-energy ring (HER), KEKB low-energy ring (LER), Photon Factory (PF) and PF-AR [1] [2]. The distance from the linac to the injection points of various storage rings is about 1km. A well controlled stable operation is required to maintain high luminosity. A wire scanner (WS) is useful for non-destructive monitoring of the beam profile for such long beam lines. A set of four WS are used for beam emittance and Twiss parameter calculation in optics matching. There are seven such matching sections in LINAC and BT's. The design of WS's and its measurement software were reported elsewhere [1, 3, 4].

At present, the WS's data acquisition system is comprised of CAMAC based front-end hardware e.g. CSADC, Scaler and DAC. A VME based supervisory system, running EPICS IOC on VxWorks 5.5, is used to control the pulse motor based WS system and Photomultiplier Tube (PMT) high voltage. An independent application program is used to acquire data from CAMAC hardware and save it in a memory based table. The supervisory system acquires the data from the table. The data acquisition process is synchronised with

#r_ani@vecc.gov.in

the beam pulse by an independent gate generation system. Since LINAC is used for injecting beam of different characteristic into three storage rings (HER, LER and PF) simultaneously [5], hence a system synchronised with LINAC timing system, may be useful to acquire various beam mode data simultaneously. A VME based system, as shown in Fig.1, is developed to utilise timing system events for WS data acquisition. The speed of the wire is also changed while scanning the beam for obtaining maximum data points in minimum scanning time.



Figure1: Hardware architecture of the system.

HARDWARE

The new system is comprised of Motorola MVME-5500 CPU, Hoshin V004 Scaler, Hoshin V005 CSADC, PVME DAC, Agilent LAN/GPIB converter and Micro Research Finland Event Receiver (EVR). The system is connected to LINAC timing system using single mode optical fiber through EVR. The EVR is used to generate gate pulse for CSADC synchronised with incoming events from global event generator of the timing system. The movement of the wire is controlled through pulse motor controller using LAN/GPIB converter. A GPIB based multi-channel digital voltmeter is used measure the absolute wire position through potentiometric arrangement. The DAC is used to control the PMT bias voltage.

SOFTWARE

The Experimental Physics & Industrial Control System (EPICS), a standard open-source dual layer software tool for designing distributed control system, is adopted to implement the supervisory control software in LINAC as well as in all four storage rings. An EPICS Input-Output Controller (IOC), running on VxWorks 6.8, is developed for control and data acquisition for the WS system.

Since the user interface panels are developed and tested for the existing IOC of the WS system. Hence it is decided to retain the same software interface with user interface panel while developing the new system. The foregoing requirement is resolved by designing a new EPICS record, Wire Scanner (WS) record. In this record, data from various sources e.g. CSADC, Scaler, Beam Position Monitor (BPM) etc are acquired through input links and stored in a predefined array format while processing of the record. The maximum number of such array is to be specified while defining the record. To incorporate BPM data, options are kept for obtaining BPM signals and further calibration of the BPM signals. The Beam mode data is appended dynamically to the array through an additional field. A field is also provided to specify the delay in processing the record to fine tune the hardware e.g ADC with event signal. The record is designed as a ring buffer of the data array. A header is provided to specify total size of the buffer and the latest data position in the buffer. In this system, the buffer length is kept for storing 2048 events. On each event, the record acquires data from four scaler channel, three BPM signals and twelve CSADC channels.

In LINAC and BT, the simultaneous top-up injections to three rings, KEKB-HER, KEKB-LER, and PF is realised by a global fast event-based control system [6]. In this system, various beam line equipments are controlled through a set of event codes, broadcasted sequentially for each beam mode. An event generator, clock synchronised with RF, is used to sent event codes along with clock to various event receivers distributed along the beam line through optical link. The event receivers decode the event codes and generate timing signals for the related IOC's to control subsequent beam line equipments e.g. klystrons, magnets etc. A set of Micro Research Finland event generator and event receiver modules are used for timing system in LINAC and BT. There are maximum six event codes e.g. preparation, klystron-1, klystron-2 etc. for each beam mode in the present configuration. In the wire scanner system, the event signals for klystron 1 & 2 are used for data acquisition. The event signal for klystron-1 is used to set the gate width and gate delay value in EVR depending on the beam mode. The subsequent klystron-2 signal is used for gate generation and data acquisition.

A set of EPICS device drivers are developed for CSADC, Scaler and DAC hardware, while 'asynDriver' is used for LAN/GPIB and existing latest driver is used for EVR. Since there are three distinct positions of the wire scanner for maximum signal, peak, in PMT. Hence a combination of slow and fast speed is used during scanning to get maximum data at the desired wire position keeping overall scanning time to a minimum. The options are provided for specifying high speed, slow speed, peak positions and width around the peaks by the user.

TESTING & TEST RESULT

The new wire scanner system is installed in the Sector -5 of the LINAC. After installation, the gate delay and width are adjusted with PF, PF Study, AR and AR Study modes. The wire speed, peak positions and width are optimised after multiple scanning. Some minor changes are also done in the user panel to adopt software interface of the new system. The final system is tested with PF and PF Study modes. The test results are shown in Fig. 2 & 3.



Figure 2: Test results.



Figure 3: Test results.

It can be seen that the wire speed is fast at the beginning of scan and becomes slow down around the three wires. Owing to that, we can speedily take the data points enough for obtaining the width of the peak although the beam repetition rate is constant, e.g. 10 pps. A user panel is developed using EPICS MEDM tool for fine tuning of the system by specifying various parameters e.g. wire speed, peak position, width, beam mode etc. It is shown in Fig.4.



Figure 4: User panel.

CONCLUSION

The new system is developed to acquire wire scanner data of multiple beam modes simultaneously. This system will contribute significantly for beam tuning during SuperKEKB commissioning and subsequent stages.

ACKNOWLEDGMENT

Authors would like to acknowledge their kind regards to Dr. T. Nakamura, KEK and Dr. R. K. Bhandari, Dr. D. Sarkar and Dr. S. Pal, VECC for their kind support and encouragement towards this development.

- N. Iida et al., "The First Results of Optics Matching Using Wire Scanners for the KEKB Beam Transport Line", Proc of PAC99, New York City, USA., March 1999, pp.2108-2110.
- [2] I. Abe, el al., "The KEKB injector linac", Nucl. Instrum. Methods A 499 (2003) 167-190.
- [3] N. Iida et al., "Recent Progress of Wire Scanner Systems for the KEKB Injector Linac and Beam Transport Lines", Proc of EPAC2000, Vienna, Austria, June 2000., pp.1738-1740.
- [4] Y. Funakoshi et al., "Beam Tests of a Wire Scanner for the KEKB Injector Linac and Beam Transport Line", Proc of APAC'98, Tsukuba, Japan, March 1998, pp.537-539.
- [5] N. Iida, el al., "Pulse-to-pluse Switching Injection to Three Rings of Different Energies from a Single Electron LINAC at KEK", Proc. of PAC09, Vancouver, BC, Canada, WEPFP110.
- [6] K. Furukawa et al., "New Event-based Control System for Simultaneous Top-up Operation at KEKB and PF", Proc of ICALEPCS2009, Kobe, Japan, Oct.2009, pp.762-764.

GRAPHICAL USER INTERFACE (GUI) FOR TESTING CAMAC MODULES

S.G. Kulkarni, J.A. Gore, A.K. Gupta, P.V. Bhagwat, S. Kailas, BARC, Mumbai, India

Abstract

A new program (GUI) for testing CAMAC modules (CAMAC ADC, DAC, Input Gate, Output Register) is developed using LabVIEW and dynamic link libraries (DLLs). On start-up, the program initializes the CAMAC Controller via PCI bus interface, thus enabling communication with CAMAC modules. It can test CAMAC modules through different controls like slider bars, buttons etc. and display status of individual channels with soft panel meters and LEDs. The GUI is extremely useful in troubleshooting hardware problems of CAMAC modules and also in developing new modules.

CAMAC TEST SET-UP DESCRIPTION

The CAMAC modules test program (GUI) is written in LabVIEW and has one page for testing CAMAC ADC/DAC modules and other page for testing CAMAC Input/output Gates. Figure 1 shows the basic block diagram of CAMAC Test set-up. The GUI communicates with the dynamic link libraries (DLL) which in turn call the PCI driver (associated with PCI Interface card). Separate DLLs are used to implement CAMAC read command and CAMAC write command. The DLL initiates the PCI Interface card to communicate with CAMAC controller via parallel bus interface. The CAMAC controller accepts command from PCI Interface card and passes that command to CAMAC modules via CAMAC bus. If it is a "CAMAC WRITE" command then the module receives data from PC in appropriate register (channel number called sub address). If it is a "CAMAC READ" command then the module sends data (from appropriate channel number) to PC via CAMAC Controller.

The CAMAC ADC/DAC page has user interface to enter station number N (where the module to be tested is present). The DAC has 8 channels and hence 8 slider bars are used to control each channel's voltage output from 0 to 10 V. If it a 12-bit CAMAC DAC module located in slot number 12 and we want to set 5V at channel number 6, then station number entered into GUI is 12 and sixth slider is used to set 5V. 0 to 10 V is converted into 0 to 4095 (as DAC is 12 bit). 5 V is converted into 2047 by the slider control. This data is passed to CAMAC Controller which passes this to station number 12 (it also sets sub-address as 5 for channel number 6). The ADC has 16 channels and hence 16 meters are used to display individual channels voltage. The range of each meter is from 0 to 10 V. Each channel converts analog input (range 0 to 10 V) into digital number (0 to 4095) and this number is read by the CAMAC controller from ADC module's appropriate channel register and send to PC via PCI Interface card. The LabVIEW program will convert this digital number into analog voltage (0 to 10V) and display it on respective channel meter. The page for CAMAC ADC/DAC module testing can test one ADC and one DAC module simultaneously.

The CAMAC Input/output gate page has user interface to enter station number (N). The Output Gate has 24 channels with each channel controlling an isolated relay contact. Whenever the channel receives logic '1' i.e 5 V input the relay operates and connects the output contact. Hence if we want to turn on the contact at 8th channel number, we have to press the eighth button (which when pressed turns green). Internally the GUI program converts this information into binary 24 bit word in which the eighth bit will be '1' and all other bits will be '0'.

Figure 2 shows us the snapshot of GUI program in LabVIEW which controls Output register and displays status of Input Gate. It also displays the internal logic value of Output Register which the user has changed. This internal register is first written by CAMAC Controller and then read by the LabVIEW program and displayed as 24 LEDs (instead of 24 bits).



Figure 1: CAMAC Testing Set-up Block Diagram



Figure 2: Snapshot of LabVIEW program testing CAMAC Output Register's 8th channel (turning 8th channels relay ON/OFF) and also watching the status of all channels.

LabVIEW GUI PROGRAM

The reason for choosing LabVIEW was ease of programming, easy interface to DLLs (active X controls), having many instrument palettes etc. [1]. The graphical way of programming is easier to understand and implement. The call library node is used to call DLL files. Separate DLL files are used for read and write operation. The ADC status is displayed continuously by using a Timer in the LabVIEW program which executes CAMAC read function every one second.

The program takes the input (station number) N and then makes a CAMAC NAF (station Number, Sub-Address, Function) command internally for each channel. This command is then passed on to appropriate call library node which executes the CAMAC command and writes/reads CAMAC data to/from DAC/ADC respecttively. For DAC module plugged in station number 20, there are eight channels and hence eight sub-addresses (0000 to 0111) whereas the function is write (CAMAC function code = 10000). Hence the CAMAC NAF (station number, Sub-address, Function) command for channel 1 is "10100000010000". For ADC module situated at station number 20, there are sixteen channels and hence sixteen sub-addresses (0000 to 1111) from where the ADC data is read (CAMAC function code = 00000). NAF command for reading from channel 1 is "101000000000". While reading data from CAMAC Controller even parity is checked and data is considered valid if parity of all data bits is even. This parity checking logic is also implemented in LabVIEW program. Timer control is used to trigger measurements every one second for ADC and Input gate modules.

ADVANTAGES

The LabVIEW program is useful for troubleshooting different CAMAC modules and tested modules are used for control of Pelletron Accelerator Facility, TIFR. This program has helped in preventive as well as breakdown maintenance of CAMAC electronics. Development of new modules is also possible due to this test program.

REFERENCES

[1] Jeffrey Travis, Jim Kring, "LabVIEW for Everyone", Pentice Hall.

RE-ENVISIONING THE OPERATOR CONSOLE FOR DHRUVA CONTROL ROOM

S. Gaur, P. Sridharan, P. M. Nair, M. P. Diwakar, N. Gohel, C. K. Pithawa BARC, Mumbai, India

Abstract

Control Room design is undergoing rapid changes with the progressive adoption of computerization and Automation. Advances in man-machine interfaces have further accelerated this trend. This paper presents the design and main features of Operator consoles (OC) for Dhruva control room developed using new technologies. The OCs have been designed so as not to burden the operator with information overload but to help him quickly assess the situation and timely take appropriate steps. The consoles provide minimalistic yet intuitive interfaces, context sensitive navigation, display of important information and progressive disclosure of situation based information. The use of animations, 3D graphics, and real time trends with the benefit of hardware acceleration to provide a resolutionindependent rich user experience. The use of XAML, an XML based Mark-up Language for User Interface definition and C# for application logic resulted in complete separation of visual design, content, and logic. This also resulted in a workflow where separate teams could work on the UI and the logic of an application. The introduction of Model View-Model has led to more testable and maintainable software.

INTRODUCTION

With years of Dhruva reactor operation, a need for up gradation of some of the instrumentation was felt in the data acquisition and processing systems, due to either obsolescence or for augmenting the facilities provided by the existing systems. Hence, taking due care of the retrofitting problems, some PC based systems have been implemented and others are being implemented in Dhruva. The Operator Consoles for these systems in Dhruva Control room have been upgraded with the latest UI design trends and technologies and created intuitive and flexible UX models. The computer based user interfaces with consistent and large screen overview displays have been introduced which has replaced the existing recorder-based displays.

DHRUVA OPERATOR CONSOLES

The Operator Consoles has been developed for continuous monitoring of nuclear, process parameters, radiation parameters and relevant status signals of Rector Trip Logic System, Alarm Annunciation System, Emergency Core Cooling System and Start-up Logic System. The OCs acquire the data from the corresponding embedded systems via dual redundant Ethernet links. The primary function of OCs is data reception, storage and display in various for-🕘 mats.

They provide:

- Multiple views of plant state for quick assessment of situation.
- Alarm Visualization(Analysing, organizing, filtering, viewing alarms)
- Archival of Periodic Data, Alarms, Diagnostics with a data life cycle of 5 years.
- Logging and reporting with data export to Excel.
- Real-time and historical data trending.

DESIGN APPROACH

The OC has been designed keeping in mind the common requirements across systems of similar categories while allowing the UI to be tailored to specific requirements of the system. This has resulted in a common OC platform. Variability between these applications were analysed and addressed through configuration points. A flexible XML schema was designed to describe configuration information related to UI. Windows Presentation Foundation (WPF) helped in separating design and code, and provided comprehensive binding framework, command infrastructure, specialized layout builders for flexible UI creation. Loose coupling of the View from the supporting logic and data (Model) was achieved through the Model-View-View Model (MVVM) pattern. Asynchronous invocation, thread pooling and object pooling resulted in efficient use of multi-core hardware to achieve the required performance. Use of lightweight embedded relational storage engine resulted in high performance concurrent storage and retrieval system.

The design of UI was mainly focused to achieve visually appealing, feature oriented, intuitive, and less cluttered graphical user interface. The rest of the paper describes the UI concepts that were used for building the user interface of Operator Consoles.

Separation of Concerns

Model-View-ViewModel (MVVM), a design pattern for UI development consists of three conceptual parts - Model, View and ViewModel. The Model is the data, completely UI independent that stores the state and does the processing. The View consists of visual elements. The ViewModel is an abstraction of the view, but it also provides a specialization of the Model that the View uses for data-binding. This resulted in complete separation between visual design, content, and logic. The separation of concerns enabled UI Unit-testing, designer-developer workflow and decoupling. The layers could be developed and changed independently of one another, resulting in parallel development. View-Model classes are easy to unit-test since they have no specific dependencies on visual elements.


Figure 1: Alarm Windows for RTLS.

Configurability

The data model for plant data acquisition systems was captured using XML technologies such as XML schema and XPath. With its flexible tree-like data structure framework, XML provides more natural alternative for knowledge representation as compared to traditional databases. XML schema with its strong type sensitivity and capability to define new complex types allowed capturing the data model accurately. Structured hierarchal in-memory representation of XML data and use of Language Oriented Query technology (LINQ) resulted in a declarative, query processing.

Modernization of User Interface

The Operator Console is built around the minimal interface principle such that it contains only features that are absolutely necessary for users to complete the activity the application is meant to support; supports the user's mental model of what it does. It contains uniformly designed interface elements, but leverages on irregularity to create meaning and importance.

Info Graphics

Visual metaphors have been provided to traditional and familiar objects to make the interface more intuitive. The OCs use monochromatic icons and simple solid colours for its new interface. The alarm windows from the Alarm Annunciation System that consists of alarms for the particular OC are represented in the same format and colour definitions as its available to the operators in the Control Room (see Fig. 1).

Laser Focus

Laser focussed interfaces helped to put visual focus on the most important details. The most valuable information like current alarms remains permanently available, while there is a shared area which is used for viewing specific in-detail data and for executing commands. The key benefit of this approach is simplicity. The Ribbon framework has been used to implement a command UI that is an alternative to the layered menus, toolbars, and task panes of traditional Windows applications. The tabs are used for displaying different peer groups of content or functionality. The application features are organized into a series of ribbon tabs at the top of a window for each kind of visualization: Trend, Alarm, Configuration, Health Messages and Snapshot. This has increased discoverability of features and functions, enabled quicker learning of the program as a whole, and made users feel more in control of their experience with the application (see Fig. 2).

Context Sensitive Navigation and Collapsed Content

The OC implements Context sensitive navigation and collapsed content to de-clutter the design. Thus, only the required navigation elements are present on screen all the time and others are shown only in certain situations. Figure 3, which shows the Trend Page where group selection panel automatically collapses, providing more space for trend viewing.

Content Chunking

Content chunking has been provided for presenting a large number of alarms in smaller visual chunks so its easier for operators to understand and interpret.

- The current alarms display is designed to provide a consolidated highest priority status (ORing of three channels alarm status); so that any signal causing system disturbance can be easily identified.
- Trend view supports "Recently viewed trends" along with the signals selected for trending. This feature provides the last three recent trends that have been selected to be viewed at a coarser resolution.

Attachment Boxes

They are an alternative to popup boxes with certain benefits. The Attachment boxes are tied to the form/ button and can be located on of its edge; as against pop up boxes that open up anywhere on the screen. This retains the context on which it was opened. It supports Hidden Navigation (Dont Leave the screen principle). The pop-up boxes have to be closed explicitly whereas the attachment boxes are hidden as the cursor is moved out. The Date Time Change, user login and shut down are provided in the form of Attachment Boxes (see Fig. 3).

Windows Presentation Foundation

Windows Presentation Foundation (WPF) has been used for UI design and development and provides the following advantages:

• Hardware acceleration: The OC workstations are provided with high end graphics that incorporates parallel computing architecture and supports Microsoft DirectX. WPF built on top of DirectX allowed fast data acquisition rate, archiving and responsiveness to the

Reactor Trip Logic System			0 🕑 🌔	01/02/2011 17:44:41
Trend Alarms Configurations Health Messages Snapshot			UserID	
Sample Instantaneous ShowSubGroups CTGroupA11_1 CTGroupA11_2 100 Source State Stat	Sample Instantaneous 🔹 🚘 Online	From 01/02/2011 17:14:11 - Mutip	eY-A Password	
Interval 200msec CoomSubGroups III CTGroupA11_3 III Signals	Interval 20msec • 👗 Offline		ck	Cancel
Group Resolution SubGroup Selection	Resolution Options	Duration Y-Axi	Options	C 🕤 📮 🂈



Figure 2: Ribbon View/Attchment Box for Login.

Figure 3: Operator Console for RTLS.

operator, permitting smoother graphics and enhanced performance.

- Resolution independence: WPFs vector graphics allowed the same user- interface to be designed for different monitor sizes of the control room(20 inch and 30 inch)
- Declarative programming: Extensible Application Mark-up Language (XAML) declarative programming was used to define the layout of application objects. This has now become a common trend in UI development for the design / code separation. This allowed parallel workflow of UI design and logic by different team members. Also, tool support helped complete an iterative lifecycle of design development and testing.
- Data Binding: OCs use the data binding capability of WPF, look-less control model and data templates, to

achieve strong separation of display from state and behaviour promoted by MVVM.

• Rich composition and customization: The OCs for different systems of Control room featured the same design and layout but were customised through themes and skins to create a radically different look.

CONCLUSION

OCs have been successfully developed for Reactor Trip Logic System, Alarm Annunciation System, Start-up Logic system and Emergency Core Cooling System for Dhruva Reactor. OCs were designed with continuous discussion and suggestion with the operators and maintenance staff, so as to provide user-centric visualization to effectively cope up with the increasing complexity of the processes to be monitored.

EMBEDDED PC BASED CONTROLLER FOR USE IN VME BUS BASED **DATA ACOUISITION SYSTEM**

G. Verma*, M. Kalra[#], S. K. Jain, D. A. Roy, B. B. Biswas BARC, Mumbai, India

Abstract

An embedded PC based Controller module, named System Controller Module (SCM), has been developed at Reactor Control Division (RCnD), BARC. This module uses standard PC-104 bus based CPU module integrated with a protocol translator card to provide an interface between the CPU module and VME bus. The signal interface between PC-104 bus of CPU module and translator card is achieved through stackable connectors. SCM can be interfaced with 16- bit slave I/O modules on VME bus for Data Acquisition and Control. This development provides low cost PC based platform for developing I/O intensive embedded system requiring high processing power. SCM module is fully compatible with PC architecture and is available in Double Euro modular form factor. Module has self diagnostics features to test software integrity using onboard watchdog timer. The module provides dual Ethernet link for communication. The SCM has been assembled, integrated and successfully tested along with VME based high speed data acquisition system (Machinery Protection System), which has been developed in RCnD for condition monitoring of rotating machines. SCM acts as a configuration controller and data manager for this system.

INTRODUCTION

There are many different industrial buses and technologies used for different applications. VME is one of the popular 16/32/64 bit backplane bus [1] which has been used in many applications like control, aerospace, military etc. Embedded systems utilizing PC-104 bus based modules are also widely used in many data acquisition applications because of its features like small structure size, self-stacking, PC platform, high quality and low cost. Using these two widely used technologies, System Controller Module has been developed, which uses PC-104 bus based CPU module along with VME interface with standard VME based I/O modules for data acquisition and control. The VME interface is achieved by using a Protocol Translator Card, also known as Protocol Interface Card, which provides protocol interconnection between PC-104 bus signals and VME bus signals. The SCM has been designed, assembled, and successfully tested along with a VME based data acquisition system (Machinery Protection System).

VME BUS STRUCTURE AND **CHARACTERISTICS**

VME bus is one of the most popular standard bus because of its mechanical and electrical robustness. A large number of commercial products are available for this open standard. A product designed using the VME bus standard can be easily interfaced with off the shelf standard products. The VME bus standard specifies a high-performance backplane bus. The VME bus supports 8-, 16-, 32-or 64 bit data transfers over a non-multiplexed data and address highway. The transfer protocols are asynchronous and fully handshaken. The mechanical specifications of boards, backplanes, subracks, and enclosures are based on IEC 297 specification, also known as the Eurocard form factor [1].

The mechanical structure of a VME bus system consists of a backplane on which system bus resides. The backplane has 21 slots where CPU boards, memory boards or I/O boards connect to the system bus.

PC-104 BUS BASED MODULES

PC architectures are popular in both general purpose and dedicated (embedded) applications [2]. But because of the large form factor of PC motherboards and I/O cards, its application in embedded systems is limited. The embedded PC-104 bus uses the PC card specification but changes the form factor. The specification defines a new mechanical foot print and card power requirements. The PC-104 bus is an adaptation of the ISA bus for embedded computing use. It uses the same signals as ISA, but uses a smaller connector and cards that are stackable, which eliminates the need for a backplane. Use of PC-104 busbased CPU modules in embedded systems provides PC hardware and software platform and ensures upgradability and protects against chip obsolescence [3]. PC-104 bus supports 8- and 16-bit data transfers.

The PC-104 bus based CPU module used in System Controller Module consists of 300 MHz Geode processor and 256 MB of SDRAM system memory. It provides 10/100 Mbps dual Ethernet port which can be used for high speed data communication. It also provides standard interface for I/O devices like USB, CRT, TFT, Mouse and Keyboard. It implements a Watch dog timer for software and hardware integrity. PC-104 module provides PC-104 expansion interface which is utilized for development of a Protocol Interface Card to facilitate VME interface.

DESIGN OF INTERFACE

The PC-104 bus based CPU module along with the Protocol Interface Card forms the System Controller

^{*}gaurav@barc.gov.in

[#]kalra@barc.gov.in

Module (SCM) as shown in Figure 1. The Protocol Interface card provides interfacing between PC-104 bus and VME bus. The interconnection between the CPU module and Protocol Interface Card is through stackable connector. The Protocol Interface Card has Eurocard formfacor. Figure 2 shows the block diagram of the SCM. SCM acts as a master module and can control slave I/O modules on the VME bus. The Protocol Interface card utilizes a CPLD which translates signals from one bus type to the other. VHDL language is used for the implementation of CPLD logic. The Algorithm implemented in CPLD for data transfer between the PC-104 card and VME slave module is as follows:



Figure 1: System Controller Module.

- Assert address strobe on VME bus on valid address of PC-104 bus.
- Wait for *write signal* from the PC-104 and assert *data* strobes and write signal on VME bus and de-assert ready signal on PC-104 bus on reception of write signal on PC-104 bus.
- Wait for acknowledgement on VME bus from I/O slave and assert ready signal on PC-104 bus after reception of acknowledgement.
- Wait for write signal to de-assert on PC-104 bus and then de-assert data strobes, write signal on VME bus.
- De-assert address strobe on VME bus after deassertion of acknowledgement.





EXPERIMENTS AND RESULTS

SCM has been tested and validated on two experimental setups. In the first experimental setup, the SCM is validated by integrating and testing it with standard VME bus based Analog Input Board (AIB). VME based AIB is a high performance analog I/O board designed for safety applications. Application software for SCM has been developed on Borland C. In this setup, one AI board and SCM sit on the VME back plane. A reference analog input is connected to the AI board. The SCM configures the AI board and periodically reads the analog input data. Reception of correct data validates the VME interface of the SCM. Figure 3 shows the timing diagram of the VME and PC-104 signals taken through oscilloscope for one cycle of data transfer. Diagram shows valid signal generation on the VME and PC-104 bus.



Figure 3: Timing diagram of VME and PC-104 Signals.

In the second setup the SCM has been assembled, integrated and successfully tested along with VME based high speed data acquisition system known as Machinery Protection System (MPS) [4]. This system has been developed in RCnD for condition monitoring of rotating machines. SCM acts as a configuration controller and data manager for this system. In this setup, processing module of MPS known as Machinery Protection Module (MPM), having VME interface, is mounted on the VME back plane along with SCM. MPM acts on the configuration data received from the user through Engineering Console. In this setup SCM provides configuration data to the MPM through VME interface. SCM receives configuration parameters from the Engineering Console through Ethernet communication. Reliable Ethernet communication between SCM and Engineering Console is achieved via TCP/IP over Ethernet. Open source TCP/IP stack is used to facilitate TCP/IP communication. In this experiment SCM is able to configure the system and the required data transfer rate of 2Mbytes/second is achieved.

CONCLUSION

This paper has presented an embedded PC based Controller module which provides interface between the

CPU module and VME bus. The design provides a low cost PC based processing platform for VME bus based I/O boards.

ACKNOWLEDGEMENT

The authors express sincere gratitude to Shri G. Ganesh, OS, Reactor Control Division, BARC and Shri G. P. Srivastava, Director, E&I Group, BARC, for their encouragement and support for this project.

- [1] American National Standards Institute, IEEE standard for a versatile backplane bus: VMEbus, ANSI/IEEE Standard 1014-1987.
- [2] PC/104 Specification, Version 2.3, June 1996.
- [3] M. Kalra, D. A. Roy, C. K. Pithwa, "Interface Board for Interfacing PC-104 Bus Based SBC Module With Euro Bus Based Boards", BARC Newsletter, December 2004.
- [4] D. A. Roy, P. K. Kavalan, Sanjay Jain, Mohit Kalra, B. Kannaiah, C. K. Pithawa, B. B. Biswas, "Development of Machinery Protection System", ICOVP'07, BESU, Shibpur, Jan-Feb 2007.

A LOW-COST HIGH-PERFORMANCE EMBEDDED PLATFORM FOR ACCELERATOR CONTROLS

Stefano Cleva, Alessio Igor Bogani, Lorenzo Pivetta Elettra-Sincrotrone Trieste S.C.p.A., Trieste, Italy

Abstract

Over the last years the mobile and hand-held device market has seen a dramatic performance improvement of the microprocessors employed for these systems. As an interesting side effect, this brings the opportunity of adopting these microprocessors to build small low-cost embedded boards, featuring lots of processing power and input/output capabilities. Moreover, being capable of running a full featured operating system such as GNU/Linux, and even a control system toolkit such as Tango, these boards can also be used in control systems as front-end or embedded computers. In order to evaluate the feasibility of this idea, an activity has started at Elettra to select, evaluate and validate a commercial embedded device able to guarantee production grade reliability, competitive costs and an open source platform. The preliminary results of this work are presented.

INTRODUCTION

During the last years the requirements of particle accelerator control systems moved from the traditional distributed architecture, based on modular but complex platforms, such as VME, to even more distributed systems based on simple embedded devices [1,2,3]. The ever-growing performances of the modern hand-held oriented system-on-chip devices allow nowadays to fulfil these requirements. Desirable characteristics are:

- a large set of Input Output (I/O) subsystems (GPIO, SPI, UART, PWM, ...);
- remote control/communication interfaces;
- multiple communication protocols (UDP, TCP/IP, field-bus based);
- Operating System (OS) support, • full with multitasking, multi-user, real-time capabilities;
- hardware, software and documentation support by either the manufacturer or a dedicated third part player;
- long term commercial availability and support;
- flexibility and modularity to cover a wide range of different fields of application;
- competitive cost-performance ratio;
- competitive development and maintenance costs;
- deterministic (real-time) capabilities.

Commercial-off-the-shelf (COTS) products that can a match most, even if not all, of the described features are now available. Adopting the "system integrator" point of view, the available devices can be classified into the ☑ following levels:

• SOC (system on chip) level;

- SOM (system on module) level;
- SOB (system on board) level.

Design complexity, available resources and manpower are practical reasons that lead to choose the easiest, fastest and most effective solution: find the most suitable commercial SOB and customise it in order to fit the specific requirements.

MARKET SURVEY

A market survey has been carried out to find COTS products to compare, taking into account performance, reliability, lifetime and, last but not least, cost. This preliminary step has led to three product platforms: the iMX by Freescale [4], the OMAP and the Sitara by Texas Instruments (TI) [5].

The Freescale iMX53 "QSB" board has turned out to be a clean design, quite powerful for an embedded board, allowing the native porting of the Tango control system framework [6] in use for the FERMI@Elettra project [7]. On the downside, lacking a simple way to reach the I/O connector, the flexibility of the platform hasn't been satisfactory.

On the TI side, the "Beagle" family boards have been evaluated with good and encouraging results .

THE "BEAGLE" PLATFORM

Formally announced by Digi-Key [8] on July 28th, 2008, the "Beagle" project has been immediately able to collect the attention of a large community of developers whose main interest is focused on real and complete open source projects [9]. The Beagle family includes, at present, three different SOB's:

- the BeagleBoard, the first member of the family, a fully featured, low-power, high-performance Single Board Computer (SBC) based on TI OMAP3530 SOC, designed for user-oriented applications;
- the BeagleBoard-xM, the evolution of the BeagleBoard, based on OMAP compatible TI DM3730 SOC;
- the BeagleBone (Fig. 1), the most recent board of the family, based on TI Sitara AM335x SOC, designed for machine and/or industrial-oriented applications.

The BeagleBoard and the BeagleBoard-xM are, in principle, very similar to the iMX53 QSB board. This means that they are powerful devices from the computational point of view but, due to design decisions, they feature a small expansion connector, too limited to be used in an industrial oriented platform.

respective authors

the



Figure 1: BeagleBone board.

On the other hand, the BeagleBone has some key features as:

- compact form factor;
- robust, accessible expansion connectors;
- large number of exported I/O pins;
- enough computational power;
- deterministic execution hardware support by means of a dedicated processing unit;
- 256 MB RAM and microSD card slot:
- native Ethernet interface;
- open source approach, on both hardware and software;
- manufacturer board support packages (BSP) for Linux and Android;
- large community of developers and users.

Furthermore, a number of interesting projects have already been based on the BeagleBone platform.

THE AM335X PROCESSORS FAMILY

The core of the BeagleBone is presently the AM3359 SOC, a member of the AM335x family. The general architecture of this family is shown in Fig. 2.



Figure 2: AM335x SOC architecture.

The main features of the AM3359 device are:

• ARM Cortex-A8 core running at up to 720 MHz clock frequency;

- programmable real-time subsystem unit (PRUSS) [10], made of a couple of deterministic RISC cores able to access the whole SOC memory space;
- rich set of peripherals accessible from both ARM and PRUSS cores;
- native Ethernet interfaces supporting PTP (IEEE 1588) and Ethercat.

Currently the AM3359 is the top level device of the AM335x family and supports the whole set of functionalities depicted in Fig. 1. The AM3358, AM3357, AM3356, belonging to the same family but less feature-rich, could be, nonetheless, suited for specific applications where Ethercat and PTP capabilities are not required. Due to the lack of the PRUSS engine, the entry level devices AM3354 and AM3352 have been discarded.

PLATFORM EVALUATION

The board evaluation process has been split in three phases. The first phase has been dedicated to testing of both the native and cross development tools (GCC tool chain) and building and running a number of sample applications. A Ubuntu [11] distribution flashed on a bootable microsSD card has been used. These tests have shown that it is possible to compile and run natively OmniORB, Tango and Tango based device servers, both locally on microSD and remotely on an NFS mounted root file system. A number of minor issues have still to be fixed to successfully cross compile Tango based applications.

The second phase has been focused on testing the Linux kernel and platform BSP. The device drivers for specific peripherals such as SPI, PRUSS and the support for kernel subsystems have been studied and modified to fulfil the required performance figures. Starting from the official TI Linux BSP, an extended kernel has been obtained by patching and cross compiling the original one. The original Ubuntu distribution root file system has been kept on the microSD card whereas the new kernel has been mounted via NFS. This set up has been extremely helpful because it has allowed to share the same remotely stored kernel and applications under test between multiple boards, leading to a simple, quick and effective bug fix activity, especially for the PRUSS.

Finally the original BSP kernel on the microSD card has been replaced by the renewed one, tested and validated. In this way, the device has become a complete stand-alone system that boots from the local removable storage and then loads the specific drivers and applications.

THE FIRST APPLICATION

Within the context of particle accelerators control systems the set of possible applications of the BeagleBone ranges from low to middle level complexity. It could be used as an intelligent hub for networks of sensors directly connected to the machine, as a gateway between different interface field-buses or as a controller embedded into a device or equipment.

A concrete example is the "TipTilt Controller" (TTC), a real-time mirror mover designed in house. In order to perform "pump and probe" experiments at the experimental stations, a laser transport system (LTS) is currently under development at the FERMI@Elettra Free Electron Laser. The LTS is made by a multiple-mirror 170-meter long optical path wherein the laser beam trajectory must be kept stable within few microradians. Each mirror is controlled by a dedicated TTC that, after receiving an UDP packets containing the X/Y offset information, issues the required voltages by a piezoelectric drive. The control system generates and distributes the UDP packets at up to 1 KHz repetition rate, and every TTC must be able to apply the driving signal at the same rate.

The final release of the TTC, i.e. the assembly of BeagelBone plus the expansion piezoelectric driver board, depicted in Fig. 3, is in the final stage of development, and all the main functionalities have already been tested and validated in laboratory.

The laboratory set up consists in a crate hosting an adapter board that carries a couple of SPI DACs driven by the PRUSS, and a variable rate UDP packet generator that simulates the control system.



Figure 3: TipTilt board (Courtesy P. Sigalotti).

The UDP packets collected by the TTC are managed in user space by a routine that extracts the set points and sends them to the SPI control loop running in the PRUSS subsystem. Inside the PRUSS a smooth ramp is calculated and the mirror is moved to the new X/Y position before a new UDP packet arrives.

It is worthwhile to observe that, in principle, the SPI channel could be driven by Linux itself, without using the PRUSS. However, dedicated tests have shown that the performance that the PRUSS can guarantee is at least one order of magnitude higher and, even more important, the PRUSS has a deterministic behaviour.

Moreover, the use of a real-time capable kernel [12] or of some real-time extensions [13] is foreseen to guarantee the determinism of the whole system.

CONCLUSIONS

The tests for the final validation of the BeagleBone platform is not yet complete, but the already obtained results are very encouraging and no important difficulties have been encountered so far. The AM335X family has the potentiality to cover both low-end embedded system and high-end demanding applications, where the determinism of the PRUSS can help. The design of the TipTilt controller has been the first BeagleBone based project at Elettra, and is the starting point for future applications, such as the ones in the field of power supply and motion control.

ACKNOWLEDGEMENTS

The authors would like to thank Paolo Sigalotti (Laser Team) and Giulio Gaio (Controls Team) for their contribution and support during the tests.

The authors also want to thank all the people who, day by day, contribute to make the "Beagle" an even better project.

- [1] G. Bassato, W. Zou, "Experience in using credit card size boards based on Coldfire microprocessors and running under µClinux", PCaPAC 2008.
- [2] J. Yan, D. Sexton, W. Moore, A. Grippo, and K. Jordan, "Ethernet based embedded system for FEL diagnostics and controls", PCaPAC 2006.
- [3] C. Scafuri, V. Forchi', G. Gaio, N. Leclercq, "Embedding a TANGO device into a digital BPM", PCaPAC 2006.
- [4] http://www.freescale.com
- [5] http://www.ti.com
- [6] http://tango-controls.org
- [7] E. Allaria et al, "Highly coherent and stable pulses from the FERMI seeded free-electron laser in the extreme ultraviolet", Nature Photonics 6, 699–704 (2012)
- [8] http://www.digikey.com
- [9] http://beagleboard.org
- [10] BeagleBone PRU samples package, https://github.com/beagleboard/am335x_pru_package
- [11] Ubuntu on the BeagleBone with Userland SPI,
- http://waxpraxis.tumblr.com/post/19277633509/ubuntu-onthe-beaglebone-with-userland-spi
- [12] https://rt.wiki.kernel.org/index.php/Main Page
- [13] http://www.xenomai.org

A WIRELESS CONTROL SYSTEM FOR THE HTS-ECRIS, PKDELIS AND LOW ENERGY BEAM TRANSPORT

R.N. Dutt, Y. Mathur, P.S. Lakshmy, U.K. Rao, G. Rodrigues, D. Kanjilal, IUAC, New Delhi, India

Abstract

The 18 GHz High Temperature Superconducting ECR ion source at the Inter University Accelerator Centre (IUAC), New Delhi is the first High Temperature ECR Ion source designed for operation on a 400 kV high voltage platform as part of the High Current Injector development programme. For automation of source operation and control of source parameters, a wireless control system has been developed. Connection to the Ethernet based central control system has been provided.

INTRODUCTION

The 18 GHz High Temperature Superconducting ECR ion source (ECRIS), PKDELIS[1, 2], is designed to inject multiply charged ion beams into the superconducting linear accelerator. The HTS-ECRIS is driven by an 18 GHz, 1.8 kW klystron and has been recently upgraded with an additional frequency by incorporating a TWT amplifier for alternate frequency injection and for double frequency operation [3]. The low energy beam transport section consists of the HTS -ECRIS, beam extraction system and a large acceptance mass analyser followed by a diagnostic system consisting of double slits, beam profile monitors and faraday cups [4]. The control system has been designed keeping standardization and reliability of the complete system. Module backplane, isolation channel, IO hardware and embedded hardware are industrial strength, fail-safe, reliable and widely supported. Speeds of module backplane, IO hardware and isolation channels are sufficient for feeding the control and automation requirements. Incorporation of spark protection and safety interlock systems are additional requirements for smooth operation of the complete system. Fieldbus technology offers several benefits in an application. ion source control An **RS-485** MODBUS/RTU module backplane can be implemented using a single twisted pair and connect seamlessly to a wireless isolation channel, forming an industrial strength system. Simplicity of this bus helps ruggedness and easy maintenance. Multiple sourcing of industrial strength ADCs, DACs, DIO, relays and thermocouple modules is supported. Large amount of software development bus has been implemented as module backplane. This network, spread across the isolation channel (Radio Modems), appears as a local RS-485 MODBUS to the control computer on the ground. Such direct integration support for both Windows and LINUX systems is available. An RS-485 MODBUS/RTU twisted pair local of the isolation channel makes control parameters available in real time. The system has proven to be simple, rugged, robust economical and reliable. Common electrical isolation channels in ion sources are fibre optic links. However, a radio channel can maximize reliability and minimize downtime due to total absence of any mechanical contacts and make source operation easy and convenient. Commercially available ISM band 2.45 GHz CDMA radio modems have high data integrity and reliability. The TWT amplifier and associated function generator provide only GPIB channel for control. These systems are connected using a GPIB to RS-232 conversion and connection via a Radio Modem. An additional software module for interfacing to the GPIB functions via the converter is then required. The central control system of IUAC runs a high level protocol using an Ethernet TCP/IP based central control scheme. A distributed control topology is used to enable control from the control room of IUAC. Several enhanced control functions are available in the local mode. Local control also supports enhanced graphics and GUI based control functionality. The system has been running for more than five years with reliable operation.

DESCRIPTION

A block diagram of the system architecture is shown in Figure 1. Radio modem links RM1a-RM1b and RM2a-RM2b provide the basic isolation channels for systems of the source on a 400 kV high voltage platform. The RM1a-RM1b link provides control of all the parameters. The RM2a-RM2b link is for the GPIB based second link for the TWT amplifier and the function generator associated with it. The ground potential systems have more flexibility due lack of the limitation of the isolation channel. A non-isolated RS485 MODBUS serves this purpose. The MODBUS parameter scan program scans the bus for all the parameters at a regular interval. The scaling and linearization is done individually and the parameters are displayed graphically as well on GUI objects.

Interlocks are provided via PLCs as they provide the most modern and reliable mode of interlock. They can also be connected to the control system to provide the interlock status. They also provide connectivity using standard protocols.

SOFTWARE SYSTEM

A simplified block diagram of the software is shown in Figure 2. The program uses GUI for display of control and read-back parameters. Graphical plots provide histories of vital parameters like gas pressure, vacuum levels at various positions in the beam-line, drain currents from power supplies etc. Addition of auto-scan and interlock features has been implemented. To support Ethernet based control protocol, a service interface has



Figure 1: Schematic of the wireless based control system.

been implemented on Ethernet for connection to main control after parsing are passed on to the front end for IO access. Parameters are looked up from database file containing local MODBUS addresses and parameter scaling information.

INTERLOCKS

Interlocks are provided for high voltage power supplies, magnet power supplies, RF amplifier, vacuum pumps and other sensitive systems. To maximize reliability, a PLC system has been incorporated with interlocks set at the hardware level.

SERVICE, MAINTENANCE AND UPGRADATION

All the control hardware including the control modules, the isolation channel, the server hardware, the control hardware and other systems are devised as industry standard systems. A standard RS-485 has been used as the standard backplane and MODBUS as a standard protocol sourcing of control hardware is standardized. The Ethernet server and hardware are built using standard Intel based fan-less systems. A strong support exists for MODBUS libraries for all the platforms including UNIX, LINUX, Windows(TM) and LabVIEW (TM).

CONCLUSION

The wireless based control system for the HTS-ECRIS and low energy beam transport has proven to be a reliable workhorse in terms of smooth operation [5] inspite of the severe spark environment. Interference due to sparks during the communication and control of various parameters has not been observed, since the spark frequency spectrum is way below the control frequency of 2.45 GHz.

CONTROL PROGRAM / SERVER
MODBUS / RTU RS485
IO MODULES

Figure 2: Control program and interface.

- [1] D. Kaniilal, G. Rodrigues, C. Beith, S. Kantas, P. Sortais, C. P. Safvan, P. Kumar, U. K. Rao, A. Mandal, A. Roy, Proc. Indian Particle Accelerator Conf. 2003, p 144.
- [2] D. Kanjilal, G. Rodrigues, P. Kumar, A. Mandal, A. Roy, C. Beith, S. Kantas, P. Sortais, Rev. Sci. Instrum. 77, (2006).
- [3] G. Rodrigues, P. S. Lakshmy, Y. Mathur, U. K. Rao, R. N. Dutt, P. Kumar, A. Mandal, D. Kanjilal, A. Roy, Proceedings of ECRIS 08, Chicago IL, USA, P 107.
- [4] C. Beith, S. Kantas, P. Sortais, D. Kanjilal, G. Rodigues, S. Milward, S. Harrison, R. McMahon, Nucl. Instr. Meth. B235 (2005), 498.
- [5] R. N. Dutt, G. Rodrigues, Y. Mathur, U. K. Rao, A. Mandal & D. Kanjilal, Proceedings of InPAC 2011, Feb 2011, New Delhi, INDIA.

DEVELOPMENT OF AN ETHERNET ENABLED MICROCONTROLLER BASED MODULE FOR SUPECONDUCTING CYCLOTRON ECR BEAM LINE CONTROL

M. Chatterjee[#], D. Koley, P.Y. Nabhiraj, VECC, Kolkata, India

Abstract:

An Ethernet enabled control and data acquisition module is developed for remote control and monitoring of the ECR beam line equipment of the Superconducting Cyclotron. The PIC microcontroller based module supports multiple general purpose analog and digital inputs and outputs for interfacing with various equipments and an embedded web server. The remote monitoring and control of the equipment are achieved through the web based user interface. The user authenticated access to control parameters and module configuration parameters ensures the operational safety of the equipment under control. This module is installed in Superconducting Cyclotron ECR beam line for the control and monitoring of vacuum pumping modules, comprising of pumps, gate valves and dual vacuum gauges. The installation of these modules results in a distributed control with localised field cabling and hence better fault diagnosis.

INTRODUCTION

The Electron Cyclotron Resonance Ion Source (ECRIS) plays a major role in generation and injection of the different ions to be accelerated by the Superconducting Cyclotron. ECR ion source develops multiply charged ions which are transported to the cyclotron through the injection line.

A large number of beam line equipments including various types of magnets, beam diagnostics elements, vacuum system components etc are required to be monitored and controlled for efficient transportation of the ion species. Among these elements, the vacuum pumping modules, though crucial for maintaining high level of vacuum inside the ion source and the injection line, are operated less frequently. Each pumping module again comprises of a scroll pump, a turbomoleculer pump, various valves and vacuum gauges which altogether can be considered as a complete unit of one pumping station. Hence a generalised, modular, distributed control and data acquisition system for the pumping stations as well as for other diagnostic elements result into a compact, less complicated and cost effective solution.

Microcontroller is the obvious choice considering its cheaper cost, faster development, and rich in built external interfaces. But the connectivity for data communication to PC is limited to serial interface for the microcontrollers.

#mou@vecc.gov.in

In recent time, the Ethernet protocol has been widely adapted for data communication over wide area. Therefore a microcontroller based control and data acquisition module with embedded Ethernet connectivity is the optimum choice for the above mentioned control system.

In this paper, the development of an Ethernet enabled control and data acquisition module with embedded web server is explained along with the hardware and software environment detail. The schematic in Figure. 1 shows the basic architecture of the module.



Figure 1: Block schematic of the control architecture.

HARDWARE OVERVIEW

The Ethernet enabled control and data acquisition module is developed for remote control and monitoring of these ECR beam line equipments. An embedded web server along with the other essential hardware makes the module suitable for control and monitoring of the field devices over LAN. The schematic in Figure .1 shows the basic architecture of the module.

Microcontroller

The microchip PIC controller 18F4620 is used as the main controller. It is an 8 bit microcontroller with 64kbytes of Flash, 10 bit, 13 channel ADC, 31 level stack and 36 Inputs and outputs (I/O). The 1Kbyte of EEPROM data memory available in this IC is used to store the configuration data. The Master Synchronous serial port (MSSP) module in this microcontroller supports 3 wire SPI communication, used for communicating with Ethernet controller.

Ethernet Controller

The ENC28J60 is used as Ethernet controller. It is a IEEE802.3 Compatible, stand alone Ethernet controller with integrated MAC and Phy with SPI interface. The operating frequency of the ENC is 25MHz. This controller supports one 10Base-T port with automatic polarity detection. The MAC layer in this controller supports Unicast, Broadcast and multicast packets, whereas the Physical layer supports the loopback mode. There are two programmable LED outputs for LINK, Tx/Rx, collision and Full/Half duplex status. The proper connectivity to magnetic and terminations are important for the completion of Ethernet interface. More information about these two ICs can be found in [1].

This embedded Ethernet module works as a web server and is placed in the local area network through which it communicates with the client PC. The hardware of this module is developed in house.

All the digital Inputs and Outputs (I/Os) from the field devices are interfaced to the controller card via a relay interface card. Each module with Ethernet connectivity can support three numbers of digital inputs, three digital outputs and two analog inputs. The IP (Internet Protocol) address of this module can be configured remotely over the LAN from the client PC and the module can be booted either with the default IP or the IP configured by the user by using jumper selectable switches. A temperature sensor (AD22100KT) with inbuilt ADC for monitoring temperature in the range of 0°C to 100°C is also provided in this module. The analog field signals e.g. pump speed, vacuum reading etc, are converted to digital by the internal ADC of the microcontroller. A LCD provided on local panel displays the IP address and the ON/OFF status of the field devices besides other parameters.

FIRMWARE FEATURES



Figure 2: TCP/IP stack and the protocols used

The Microchip TCP/IP stack [1] has been implemented in this module with minor modifications. The stack and the protocols used are shown in Figure 2. This stack mainly includes the ARP, IP, ICMP, TCP and HTTP. The ARP (Address resolution protocol) converts the IP address to physical address. IP (Internet protocol) provides the routing information for the packet. ICMP (Internet control message protocol) used for network © management. ICMP supports ping, by which the modules presence in the network can be detected. TCP (Transfer control protocol) provides reliable data flow over the network and the HTTP (Hyper text transfer protocol) is responsible for the Data transmission and reception from the Webpage.

The CCS C compiler [2] with MPLAB [1] environment is used to develop the code and Ethereal [3] software has been used extensively during the development for trouble shooting. The program flow is shown in Figure 3.



Figure 3: Program flow.

WEBPAGE FEATURES

The control and monitoring of the equipments is achieved through a web based user interface. The web page is developed using hyper text markup language (HTML). Total six numbers of pages are hosted by this module. These are Default (Feedback) page, Log In page, Access verification success page (shown in Figure 4), Access verification fail page, Control page, and the Configuration page. On connection, the default page is hosted by the module. The default page shows the digital and analog feedbacks from the associated devices e.g. ON/OFF indication, vacuum information, and pump speed etc. The automatic refresh time of this page is kept 5 second. The access to the control page is restricted by user ID and password to ensure operational safety.

Similarly while configuring the module, one has to go through the successful log in and enter into the configuration page. The network configuration of the module like IP address, subnet mask, and gateway can be changed or configured from a remote PC over the LAN. Entering any wrong user ID and password, the 'Access verification fail' page appears. The newly configured IP address along with the subnet and gateway are saved in the internal EEPROM of the microcontroller. The device can be rebooted either with the default IP or the new one depending on the jumper set in the hardware at the time of booting.



Figure 4: Snapshots of Feedback, Log in and Access verification success web pages.

FUTURE ENHANCEMNETS

The future scope of this module is to make it more generalised with more number of I/Os available at user end. The data collected from the field device can be stored in either the internal or external EEPROM for future reference which can be recollected whenever required.



Figure 5: Complete assembled module.

CONCLUSION

One set of this embedded Ethernet module has been installed in the SCC ECR beam line and working reliably over a considerable amount of time. Figure 5 shows the picture of an assembled module. This module is used for control and monitoring of one pumping station comprising of Scroll pump, Turbo molecular pump, Angle valve, Isolation valve and the adjacent vacuum gauge. It also reads the pump temperature and speed and displays it on the web page. Installation of this module made it possible to control the pumping modules from a client PC anywhere on the local area network with a minimum of field cabling.

- [1] www.microchip.com
- [2] www.ccsinfo.com
- [3] www.ethereal.com

A NEW SCHEME FOR DIRECT ESTIMATION OF PID CONTROLLER PARAMETERS

S. Srivastava[#], V. S. Pandit, VECC, Kolkata, India

Abstract

This paper presents a novel scheme for the direct estimation of a PID (Proportional Integral Derivative) controller parameters (K_p, T_i, T_d). The proposal discussed here is only applicable to first and second order stable systems. The formulation begins with system parameter identification (Transfer function of the process), which has been obtained using system identification toolbox of MATLAB. The pole zero cancellation technique is applied to estimate PID controller parameters which inturn results into the matched coefficients of the system parameters to the Controller parameters. An additional tuning parameters α is proposed in our method, which provides an additional flexibility of tuning the response time of the controller without disturbing the controller parameters. The proposed scheme is bench marked using real time case of dc motor speed control. The effectiveness and robustness of the proposed auto tuning algorithm are verified by the simulation results.

INTRODUCTION

A Proportional-Integral-Derivative (PID) controller is the most widely used controller in the industry today. The popularity of PID controller is due to its simplicity which uses only three, parameters to tune. Proportional (K_p) term which controls the plant (system) proportional to the input error. Integral (T_i) term which provides the change in the control input proportional to the integral of the error signal and the last one is the Derivative term (T_d) that control the system by providing control signal proportional to the derivative of the error signal. Derivative action is used in some cases to speed up the response time and to stabilize the system behavior [1]. Error value is the difference between the set Reference value and the output value.

Although the PID controller is known to be the simplest and efficient controller, but it requires effective and optimized tuning of the control parameters (K_p , T_i , T_d). Many PID controller tuning methods have been proposed in the literature, some of the popular methods are Ziegler– Nichols tuning ,Cohen–Coon tuning , internal model control , direct synthesis method, neural networks based methodologies, relay based auto tuning method and may be much more [2]. In this paper a simple auto-tuning method is proposed which can be applied to most of the stable first and second order systems without having time delay. The proposed tuning method comprises of two parts. First part is used for finding the system transfer function identification. Second part is used for obtaining PID parameter by arranging the parameters in such a way

#saurabh@vecc.gov.in

that the pole of the second order transfer function can be cancelled by the zeros of the PID parameters. In this way by introducing an extra parameter that is gain ' α ' we can optimize the rise time and settling time of the system. This technique basically tunes the PID parameter in a single iteration and thus it is fast and easy to implement. The paper is organized as follows. In section 2 we have discussed how to obtain the transfer function of an unknown system whereas in section 3 the PID tuning methodology is discussed. In section 4 we have presented the real time simulated case study of DC motor speed control that is done in MATLAB Simulink environment. A brief conclusion is presented in section 5.



Figure 1: Auto Tuning Scheme of PID Controller.

SYSTEM IDENTIFICATION

The system identification is the method of finding the transfer function of an unknown system by observing the input and output sequences of the system as shown in the figure 1. The transfer function (TF) identification was carried out by using system identification toolbox of the MATLAB. Let the measured input and output sequences are given by matrix:

$$\Psi(k) = \{-y(k-1), -y(k-2)\cdots - y(k-n), \\ u(k-1), u(k-2)\cdots u(k-n)\}^T$$
(1)

The dynamics of the system is given by:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}) \tag{2}$$

where \mathbf{x} is the state variable, \mathbf{u} is the input vector and $\boldsymbol{\theta}$ is the parameter vector.

$$\boldsymbol{\theta} = \begin{bmatrix} a_1 \ a_2 \ \cdots \ a_n \ b_1 \ b_2 \ \cdots \ b_n \end{bmatrix}^T \tag{3}$$

We have to choose the value of parameter vector $\boldsymbol{\theta}$ in such a way so that the difference between $\dot{\mathbf{x}}$ and $\mathbf{f}(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta})$ will impose minimum error. There are several techniques available in literature. MATLAB has in build procedure to find out the TF of the system by measuring input and output sequences using system identification toolbox. The input and output sequences are observed in a discrete domain with a sample time T_s after that the system identification toolbox uses parametric system identification technique and gives the TF in discrete domain (z domain) that is given by:

$$H(z) = \frac{b_n z^n + b_{n-1} z^{n-1} \cdots b_1 z^1 + b_0}{z^{n+1} + a_n z^n \cdots a_1 z + a_0}$$
(4)

After getting the plant transfer function in z domain it is finally converted into continuous domain by using the analogy $z = e^{sT_s}$. In the MATLAB we use the function d2c (discrete to continuous) which converts the z domain TF to s domain using Tustin approximation [3].

PID TUNING

PID tuning comprises the selection of best value of K_p, T_i and T_d of the PID controller so that the system performance can be increased. As we already discussed in introduction that there are various schemes involved in tuning the PID controller parameters. In this section we would form a simple and effective PID tuning method that can be applied to any stable first and second order systems. The tuning basically based on the logic that if we can be able to make the closed loop transfer function of the system in such a way that the poles of the open loop TF of the system exactly cancelled by the zeroes of the PID controller. Than we can make the closed loop transfer function (CLTF) of first order and then there should be no overshoot. Rise time and settling time can be tuned by introducing an extra parameter ' α ' in cascade with the input of controller as shown in figure 2. Let the second order system can be represented by:

$$G(s) = \frac{K}{s^2 + as + b}$$
(5)

The transfer function of the PID controller is given by:

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) e(s)$$
(6)

So the open loop TF (OLTF) of the system with PID is G(s)C(s). Now if we tune the parameters K_p , T_i and T_d in such a way so that the poles of the system TF (eq. 5) can be cancelled by the zeroes of the PID parameter than the closed loop TF (CLTF) becomes of first order type. By comparing the coefficients of the numerator of PID controller TF with the denominator of the system TF we can find that the value of PID parameters that make the CLTF of first order type are:

$$\begin{bmatrix} K_p \\ T_d \\ T_i \end{bmatrix} = \begin{bmatrix} a/K \\ 1/a \\ a/b \end{bmatrix}$$
(7)

When these values are put in the PID parameters then the CLTF becomes first order type. In order to have some control over rise time and settling time we introduced another parameter ' α ' at the input of controller as shown in the figure 2. This way the CLTF become in the form of:

Г-- Л

F (--7

$$\frac{Y_{o/p}}{R_{ef}} = \frac{\alpha}{s + \alpha}$$
(8)

Finally, we have a control over the rise time and settling time with the parameter alpha α . The Rise time is given by $T_r = 2.2/\alpha$ and the settling time is given by $Tss = 4/\alpha$ in case of first order system.



Figure 2: Block Diagram of the Closed Loop system with PID and parameter ' α '.

CASE STUDIES

The effectiveness of the proposed tuning method is demonstrated on speed control of the dc motor [4]. The block diagram of the DC motor is shown in figure 3:



Figure 3: DC Motor

The motor transfer function was calculated as

$$\frac{speed(\dot{\theta})}{Voltage(V)} = \frac{K}{JLs^2 + (JR + Lb)s + bR + K^2}$$

where,

Moment of Inertia of rotor (J) = $0.01 Kgm^2/s^2$ Damping ratio of mechanical system (b) =0.1 NmsElectromotive force constant (K) =0.01 Nm/AElectric resistance (R) = 1Ω Electric Induction (L) = 0.5 H

Output velocity $(\dot{\theta}) = m/s$

So, TF of the DC Motor Speed Control will become

$$G(s) = \frac{\dot{\theta}}{V} = \frac{2}{s^2 + 12s + 20.02}$$
(10)

where θ is the position and its derivative is the speed of the DC Motor, V is the input voltage given to the motor. Eq. (10) gives the original calculated transfer function of the DC motor. When we apply auto tuned PID controller method to this transfer function than the discrete estimated TF identified by using system identification technique given in MATLAB with sampling time 0.01s is:

$$G(z) = \frac{\dot{\theta}}{V} = \frac{-6.882e^{-9} + 0.0001705z^{-1}}{1 - 1.896z^{-1} + 0.8977z^{-2}}$$
(11)

after converting the z domain TF to s domain, the resultant transfer function comes out to be:

$$G(s) = \frac{\dot{\theta}}{V} = \frac{1.8}{s^2 + 10.8s + 18}$$
(12)

So we obtain the values of K=1.8, a=10.8, b=18. When these values are put in eq. (7) then the PID parameters come out to be:

$$\begin{bmatrix} K_p \\ T_d \\ T_i \end{bmatrix} = \begin{bmatrix} 6 \\ 0.0926 \\ 0.6 \end{bmatrix}$$
(13)

Figure 4 shows the comparison of step response between the PID parameters calculated for original and istimated transfer function. Figure 5 shows the step response of the DC motor speed control with the value of PID parameters given by eq. (13). Figure 6 shows the effect of parameter α over the rise time and settling time.



Figure 4: Comparison of step response between the PID parameters calculated for original and Istimated transfer function (TF).



Figure 5: Step Response of the DC motor speed control.



Figure 6: Effect of α over the rise time and settling time.

CONCLUSION

A PID auto tuning method has been formulated which directly estimates the PID parameters (K_p, T_i, T_d) . The proposed tuning method is fast and does not require complex algorithm to tune. An extra parameter α was used in series with the controller in order to have flexible control over rise time and settling time without affecting the PID controller parameters.

- [1] Sigurd Skogestad, "Probably the best simple PID tuning rules in the world" Journal of process control july3, 2001.
- [2] J.G. Ziegler, N.B. Nichols, "Optimum settings for automatic controllers", Trans. ASME 64 (1942) 759-768.
- [3] G.F.Franklin, J.David Powell, Michael Workman, "Digital control of dynamics systems" 3rd edition, Pearson education.
- [4] Carneige Mellon University control tutorial for MATLAB, http://www.engin.umich.edu/group/ctm/examples/motor/ motor.html

FPGA DATA BLOCK FIFO FOR THE APS ID MEASUREMENT SYSTEM*

J. Z. Xu, R. Farnsworth, I. Vasserman Advanced Photon Source, ANL, Argonne, IL 60439 USA

Abstract

A Hall probe insertion device (ID) measurement system has been developed to characterize the IDs at the Advanced Photon Source (APS). The system uses the latest state-of-the-art field-programmable gate array (FPGA) technology to synchronize the position and Hall voltage measurements. Data block first-in-first-out (FIFO) has been implemented to transfer the data from the FPGA to the host computer during measurement. The system is capable of continuous scanning measurements on a full 6meter bench at 1 msec per data point with a position resolution of 1 micron and Hall voltage precision of 5-1/2 digits.

INTRODUCTION

Insertion devices (IDs) at the Advanced Photon Source (APS) are measured and fine-tuned against their design specifications at the magnetic measurement laboratory [1] before installation into the storage ring. The APS Hall probe measurement system is designed to measure the magnetic field distributions of the IDs. The system has been upgraded with the latest state-of-the-art field-programmable gate array (FPGA) technology to synchronize the measurements of position and Hall voltage.

Since the FPGA hardware has limited onboard memory capacities, data block FIFO has been implemented to transfer measurement data to the host computer efficiently during measurements. The new system is capable of continuous scanning measurements of the IDs at the speed of 20 cm/sec with a spatial resolution of 0.2 mm reliably.

SYSTEM DESCRIPTION

The Hall probe measurement system consists of the following subsystems shown in Figure 1.



Figure 1: Hall probe measurement system.

A rotary stage carries the Hall sensor holder. A twoaxis (X and Y) linear stage with digital linear encoders supports the rotary stage. All the stages are remotely controlled by servo motors. A (Z) linear carriage with air bearings carries the stages as well as all the preamplifiers and electronics moving along the Z direction. The linear carriage is remotely controlled by a linear servo motor. A 7-meter-long granite table with a 6.8meter linear digital encoder hosts the carriage as well as the cables and air tubes.

A high-precision Hall probe sensor is mounted on the sensor holder. A PXI shelf with a control card hosts the FPGA card and digital multimeter (DMM) card. Software programs are hosted on the control card.

SYSTEM CONTROL, DATA ACQUISITION, AND ANALYSIS

LabVIEW-based system software has been developed to coordinate the stage control and data acquisition. Figure 2 shows the schematic layout of the system control and data acquisition architecture. The system can be accessed via the Internet from anywhere at any time, wired or wireless, through the remote desktop interface.



Figure 2: System control and data acquisition architecture schematic layout.

The system software sets the FPGA firmware to monitor the stage digital encoder positions and commands the stage scan across the measurement region. For each measurement point, the FPGA checks if the DMM is ready for measurement. If so, it trigs the DMM to take a Hall voltage measurement and records the position information to its memory; otherwise, it skips the position recording of that point. Once the DMM finishes the measurement, it will flag its state to the FPGA and wait for the next measurement trigger by the FPGA. All the above tasks are carried out on the fly. This approach guarantees that every Hall voltage measurement pairs with its position information no matter how fast the stage moves.

The FPGA onboard memory has been configured into three blocks for data block FIFO. Once the first block is full, a flag is set and the control card starts to read the data in the first block while the second block is recording,

^{*}This work was supported by U.S. Department of Energy Office of Science, under Contract No. DE-AC02-06CH11357.

and so on so forth. The FPGA hardware executes all the tasks in parallel at the speed of 40 MHz. This method minimizes the data transfer overhead between the control card and the FPGA. It enables the system to carry out position scanning measurements over a speed of 50 cm/sec with a special resolution of one micron without range limit. The system is capable of scanning measurement in all X, Y, and Z directions. In order to achieve a field measurement precision of 0.05 Gauss, the DMM card has been configured with a measurement precision of 5-1/2 digits. Therefore, with the cross-backplane communications between the FPGA card and the DMM card, each Hall voltage measurement takes less than 1 msec to complete.

The Hall voltage is converted to a magnetic field with the Hall probe calibration data in real time. Once the measurement scan finishes along with the paired positions information, the data is recorded onto the non-volatile data storage of the control card. The field and the position are also displayed on the screen in real time. For ID measurements, a magnetic analysis module written in IDL carries out the data analysis to produce the first and second integrals as well as the phase errors and radiation brightness information of the ID.

MEASUREMENT RESULTS AND DISCUSSION

A typical APS ID magnetic field measurement taken with the Hall probe measurement system in the Zdirection is shown in Figure 3. It is displayed in the Hall probe magnetic field measurement analysis and plot module.



Figure 3: APS ID magnetic field measurements.

The precision of the DMM was set at 5-1/2 digits. The measurement scanning speed was set at 20 cm/sec. With a spatial resolution of 0.2 mm, no missing data points have been observed. It was a 2.4-meter-long ID, the measurement range was set for 3 meters. The parameters shown on the left side of Figure 3 are the field

background calibrations. Total time for the scan is less than 30 seconds.

Much faster scanning speeds with much high spatial resolutions have been tested. The measurements yield missing data point pairs. Since the Hall voltage measurement pairs with its position information, no information was lost.

CONCLUSION

The Hall probe magnetic field measurement system has been upgraded, tested, and commissioned at the Advanced Photon Source. With the latest state-of-the-art FPGA technology and data block FIFO implementation, the new system is capable of continuous scanning measurements of the ID magnetic field at the speed of 20 cm/sec with a special resolution of 0.2 mm. The system measurement reproducibility is 0.05 Gauss.

REFERENCES

[1] L. Burkel, R. Dejus, J. Maines, J. O'Brien, J. Pflueger, I. Vasserman, "The Insertion Device Magnetic Measurement Facility: Prototype and Operational Procedures," ANL/APS/TB-12, March 1993.

LOW-COST EPICS CONTROL USING SERIAL-LAN MODULE XPort

N. Kamikubota[#], N. Yamamoto, J-PARC / KEK, Tokai-mura, Ibaraki, Japan S.Yoshida, KIS, Tsuchiura, Ibaraki, Japan

Abstract

In J-PARC Main Ring, we are interested in a commercial product, XPort, a low-cost serial-LAN converter module. We have introduced it in two different devices: RF-amplifiers with GPIB, and a NIM module with switches. Control messages are transported over network, and converted into EPICS-style records.

Implementation details and operational experiences are given.

INTRODUCTION

The J-PARC Main Ring (hereafter MR) is a high-power proton synchrotron with beam-energy 30-GeV. Since 2008, various machine upgrades toward higher beampower have been carried out [1-3]. The control system for MR has been developed based on EPICS (Experimental Physics and Industrial Control System) [4], where EPICS is a toolkit for large accelerator controls developed and supported by an international community [5].

It is often the case that small devices are newly introduced in order to improve accelerator performance. Typical devices are, for example, a power-supply, an electric circuit board, a stand-alone measuring system, and so on. Sometimes they have a poor interface to link to the original control system.

We, J-PARC MR control group, are interested in a commercial product, XPort [6,7]. It is a low-cost serial-LAN converter module. We implemented it into two newly introduced devices: RF-amplifies and a NIM module with front-panel switches. In both cases, control messages are transported to an EPICS IOC (Input Output Controller) over the network. They are converted into EPICS-style records using EPICS AsyncDriver.



Figure 1: XPort module mounted on a board.

IMPLEMENT XPORT TO DEVICES

What is XPort

XPort is a commercial product provided by Lantronix [4]. It is an embedded device server with various network protocols, such as TCP, UDP, SNMP, TFTP, http, and so on. It has programmable 3 I/O pins, which can be used for serial communication or bi-directional digital I/O signals. The size of XPort is as small as a RJ-45 connector. Thus, it is easy to mount an XPort module on an electric circuit board (Figure 1).

Implement XPort to RF-Amplifier

In order to improve beam quality in MR slowextraction operation, RF-amplifiers are used to add transverse noises to beam bunches [8,9]. Amplifiers are commercial products with GPIB as a default interface. We asked a company to add an XPort module for remote control. Serial messages of GPIB are simply transported over our control network, using TCP protocol.

Table 1 shows part of supported GPIB commands for the RF-amplifier. We developed EPICS configurations, device-support structures and databases as in Figure 3. Network-serial communication feature of EPICS AsyncDriver is used. Control panel was developed using an EPICS tool, MEDM, as in Figure 2.

Table 1: GPIB commands of RF-amplifier.

Function	Command	Replay message
RF SW ON	"P1"	None
RF SW OFF	"P0"	None
RF SW status	"Р?"	"P1" or "P0"
Forward Power	"FMON"	Ex) "2500W"



Figure 2: RF-Amplifier control panel.

EPICS Device Support struct

static struct gpibCmd gpibCmds[] = {

/*CMMAND 0 RF_SW_W *

(&DSET_BO, GPIBEFASTO, IB_Q_LOW, NULL, NULL, 0, 32, NULL, 0, 0, coffOnW, & offOn, "##n"},

/*CMMAND 1RF_SW_RB */

(&DSET_BI, GPIBEFASTI, IB_Q_LOW, "P?##h", NULL, 0, 32, NULL, 0, 0, coffOnRB, & offOn, "##n"},

/*CMMAND 2 FMON */

```
(&DSET AL GPIBREAD, IB Q LOW, "FMONHA", "%H", 0,
32, NULL, 0, 0, NULL, NULL, "$ $ "},
```

EPICS database

record(bi, MRSLW:TVRF_D3_1:STAT:RF_SW) {
field(DTYP, "Al01K")
field(INP, "#L0 A0 @1")
field(SCAN, "\$(scan)") }
record(bo, MRSLW:TVRF_D3_1:0PE:RF_SW_ON) {
field(DTYP, "Al01K")
field(OUT, "#LO AO @O") }
record(longin, MRSLW:TVRF_D3_1:STAT:ALARM) {
field(DTYP, "Al01K")
field(INP, "#LO AO @68")
field(SCAN, "\$(scan)") }

Figure 3: Part of EPICS device support and database.

We have used two 3 kW RF-amplifiers during MR slow-extraction operation in June, 2012, as shown in Figure 4. Only LAN cables were necessary to start remote control. However, we suffered severe network failures. Electro-magnetic noises caused by amplifiers themselves disturb network communication. In Figure 2, a graph in the red circle indicates communication failures.



Figure 4: Two 3 kW RF-amplifiers.

Implement XPort to NIM module

We have developed several customised NIM modules in J-PARC MR control. Some of NIM modules have front-panel switches. Status of such switches is not online.

Some NIM modules are already implemented XPort. In Figure 5, an on-board FPGA chip has connections both to front-panel switches and to XPort I/O pins. We are asking a company to develop FPGA logic, in which it scans switch status periodically, and communicates to the Xport module. As a result, status of switches can be read remotely as TCP messages through XPort.

Basic communication study between an EPICS IOC and a NIM module is underway. Followed by EPICS configuration, we expect some NIM modules will be online in MR operation very soon.



Figure 5: NIM module with switch, FPGA, and XPort.

DISCUSSION

In the former example, use of XPort enables us much simpler implementation than before. To avoid noise interference, we will pay more attentions to network cables in the next run in December 2012. For example, more pieces of ferrite cores will be introduced.

In the latter example, use of XPort is essential for lower-cost. So far, our standard style has been to prepare I/O modules of PLC, connected to a device using several signal cables. With XPort, except for an IOC, we need only one LAN-cable (Figure 6). Drastic cost-down is possible.



CONCLUSION

We have introduced a commercial serial-LAN converter module, XPort, to our J-PARC MR control system. Two different implementations of XPort are demonstrated. XPort enables us to use newly introduced devices, with simpler implementation and lower cost than before.

ACKNOWLEDGEMENT

We would like to thank J-PARC collaborators, especially Dr. Masahito Tomizawa, Mr. Shigeru Murasugi, and Mr. Akinobu Nagura, for their helps during tests with the RF-amplifiers. We also thank Mr. Naomichi Yamazaki, REPIC Co. Ltd., for collaborative developments of FPGA implementation on the target NIM module.

- [1] T. Koseki et al, "Challenges and Solutions for J-PARC Commissioning and Early Operation". IPAC'10, Kyoto, Japan, p.1304-1308.
- [2] T. Koseki et al, J. Particle Accelerator Society of Japan, Vol. 9, No. 1, 2012, p30.
- [3] Y. Sato et al, "Recent Commissioning of High-Intensity Proton Beams in J-PARC Main Ring", HB2012, Beijing, China, 17-21 Sept. 2012, in press.
- [4] N. Kamikubota et al, "J-PARC Control toward Future Reliable Operation", ICALEPCS 2011, Grenoble, France, p.378-381.
- [5] http://www.aps.anl.gov/epics/
- [6] http://www.lantronix.jp/products/xport.shtml
- [7] http://www.lantronix.com/devicenetworking/embedded-device-servers/xport.html
- [8] M. Tomizawa et al, "Approach for High Intensity Slow Extraction from J-PARC Main Ring", PASJ 2012 meeting, Osaka, Japan, 8-11 August 2012, in press.
- [9] A. Schnase et al, "J-PARC MR Horizontal Exciter Test for Transversal Noise Application", Proc. PASJ 2011, p.338.

FACILITY-WIDE SYNCHRONIZATION OF STANDARD FAIR EQUIPMENT CONTROLLERS

S. Rauch, W. Terpstra, W. Panschow, M. Thieme, C. Prados, M. Zweig, M. Kreider, D. Beck, R. Bär GSI Helmholtzzentrum für Schwerionenforschung, Darmstadt, Germany

Abstract

The standard equipment controller under development for the new FAIR accelerator facility is the Scalable Control Unit (SCU). It is designed to synchronize and control the actions of up to 12 purpose-built slave cards, connected in a proprietary crate by a parallel backplane. Inter-crate coordination and facility-wide synchronization are a core FAIR requirement and thus precise timing of SCU slave actions is of vital importance.

The SCU consists primarily of two components, an x86 COM Express daughter board and a carrier board with an Altera Arria II GX FPGA, interconnected by PCI Express. The x86 receives configuration and set values with which it programs the real-time event-condition-action (ECA) unit in the FPGA. The ECA unit receives event messages via the timing network, which also synchronizes the clocks of all SCUs in the facility using White Rabbit. Matching events trigger actions on the SCU slave cards such as: ramping magnets, triggering kickers, etc.

Timing requirements differ depending on the action taken. For softer real-time actions, an interrupt can be generated for complex processing on the x86. Alternatively, the FPGA can directly fire a pulse out a LEMO output or an immediate SCU bus operation. The delay and synchronization achievable in each case differs and this paper examines the timing performance of each to determine which approach is appropriate for the required actions.

INTRODUCTION

In the FAIR control system, a data master issues highlevel commands to control accelerator devices. The frontend controllers in the system react to relevant commands, issuing appropriate actions to their hardware components. Depending on the action to be taken, there are different timing requirements to be met.

Unlike the control system currently deployed at GSI, commands issued by the data master carry an absolute execution timestamp. The front-end controllers must receive commands early enough that they can schedule their actions to achieve the desired result at the correct time. Unfortunately, executing actions takes a variable amount of time. If the action takes 90-110 μ s to execute, then this places two constraints on the system. Firstly, the data master must issue commands at least 110 μ s ahead of time. Secondly, the system must be able to tolerate that the action could be as much as 10 μ s too early or too late.

Issuing commands too far in advance reduces the responsiveness of the system. Once the data master has issued a command, it cannot be aborted. If the situation changes, perhaps due to interlock or contention from another beam user, the system cannot react faster than the slowest action already executing. This neglects, of course, other sources of latency in the system, such as network propagation delay, which only exacerbate the problem. It is thus generally desirable to have fast action execution.

Non-deterministic execution time is a potentially much more serious problem. For example, if a kicker executes an action a few nanoseconds too late, the beam might be lost. However, not all actions require the same precision, and it may make sense to trade accuracy for flexibility in some situations.

Fortunately, the most common equipment controller in FAIR, the Scalable Control Unit (SCU), has several possibilities for executing actions. This paper outlines the timing requirements of various accelerator components in FAIR and explorers the alternatives which could meet them.

USE CASES

The SCU will be the main frontend controller for the FAIR project. It provides a uniform platform connected both to the timing- and the data network of the facility. In turn, the SCU controls Adaptive Control Units (ACU) [1] slaves of various form factors which provide additional features and the necessary hard- and software interfaces to control the actual accelerator components. This means a wide range of magnet power supplies, Radio Frequency (RF) components and beam diagnosis equipment. The set of executable actions of course varies depending on the connected equipment. A magnet power supply for example will be provided with parameters and timed instructions to source a current ramp to its magnet, an RF generator gets different sets of frequency and phase parameters and the time when it needs to switch between them. The basic concept of the SCU envisions a complete separation of data supply and timing/commands. This way it can make use of higher abstraction levels, i.e. complex software, which brings flexibility, is more comfortable and maintainable, and also use low level hardware implementations to provide fast, deterministic behavior for the precisely scheduled execution of commands. The device controlled in the RF use case is called FPGA Interface Board (FIB) [2]. The kicker modules will be controlled by interface devices called IFK via a MIL-STD-1553 based field bus system.

SCALABLE CONTROL UNIT (SCU)

The SCU is mechanically a stack of up to three separated boards. There is the FPGA base board with an Arria II FPGA, two Small Form-factor Pluggable (SFP) slots, DDR3 RAM, parallel flash and a parallel bus (SCU bus)



Figure 1: Block diagram of SCU.

for controlling up to 12 slave devices. In addition the base boards is equipped with White Rabbit [3] circuitry. A Com-Express module with an Intel Atom CPU is mounted to the base board. It has Ethernet, USB and PCIe interfaces. An optional extension board can be connected to the base board for backwards compatibility, that runs a MIL-STD-1553 based field bus interface.

The SCU works as a front-end controller. On one side it is connected to the control system via Ethernet, on the other side it controls slave devices over the SCU bus. It receives 1ns accurate timing information over a White Rabbit link, connected to an SFP. The White Rabbit receiver in the FPGA runs Precision Time Protocol (PTP) in software on a LatticeMico32 (LM32) soft-core CPU. The control system speaks to the Front End Software Architecture (FESA [4]) class running on the Intel Atom which is connected to the FPGA via a PCIe bridge.

EXECUTION ALTERNATIVES

When the SCU has an action to perform at a particular time, it has many alternative execution paths. Each option carries a trade-off between timing fidelity and the expressiveness of the program which performs the action.

FPGA

The SCU's FPGA can be programmed to generate the required output on a phase-aligned 8ns clock edge. When augmented by a fine delay card [5], this can be further improved to a general 1ns accuracy. The only source of non-determinism is the jitter of the FPGA's PLL (ps) and the inherent inaccuracy of White Rabbit (ns). Thus, both the delay and variability are in the sub-nanosecond range for this approach. Unfortunately, this execution path requires custom gateware and/or a simple output action.

LM32

Alternatively, the FPGA can issue an interrupt to an embedded soft-CPU (LM32). This on-chip CPU (with no operating system), can then run software to generate the appropriate action. The delay stems from the time to switch to interrupt context, run the software routine, and output the action. While broadly deterministic, cache behaviour and on-chip bus accesses contribute to runtime variability.

Table 1: Execution Timing Performance

μs	min	mean	max	stddev
FPGA	0	0.001	0.001	0.001
LM32	2.863	2.924	3.217	0.058
Kernel	7.120	13.29	37.73	3.49
Userspace	49.36	62.49	93.33	5.62
FESA	138.9	170.1	246.1	10.8

Atom-Kernel

Venturing further afield, the FPGA can issue an interrupt over PCIe to the Atom processor. The interrupt handler in the kernel driver then takes immediate action. This delays are the same as for the LM32, except that the interrupt is delivered off-chip via the PCIe bus. Furthermore, the Linux kernel may have interrupts masked in some critical sections, increasing the runtime variability.

Atom-Userspace

Rather than executing the action's software in kernelspace, the SCU could also deliver the interrupt to userspace. This adds additional context switch overhead, but provides for a more comfortable programming environment.

FESA

Finally, the userspace program which executes the action could use the FESA architecture [4]. Under this more general framework, the interrupt is translated to an action using multiple threads. This again increases the number of context switches and adds inter-process synchronization delay. However, it arguably provides the most flexible action execution framework.

ANALYSIS

To measure the delay and variability of the alternative execution paths, we connected two outputs from the SCU to an oscilloscope. The first output is the action aligned to the FPGA's 8ns clock. The second output is toggled by the execution path being measured. This approach excludes the variability of direct FPGA execution. However, this subnanosecond delay is dwarfed by the alternatives measured.

To capture worst-case variability, all test systems were subjected to a background work-load and include at least 10000 samples. For the LM32, we ran the white rabbit PTP core in the background, and performed a save/restore of all 32 registers on interrupt context switch. The Atom had a constant background task streaming text over ssh.

We tested the Atom with a real-time patched 2.6.33.6 Linux kernel. The PCIe bridge interrupt handler and kernel tasklet processes were set to real-time priority 99. For the userspace test, the test program had real-time priority 98. FESA set its own real-time priority to 60.

We also measured the LM32 without an instruction cache. This reduced the variability slightly from 354ns to 272ns, but greatly increased the average delay from



Figure 2: Comparison of delay distributions.

2.924 μ s to 3.810 μ s. Most of the variability appears to stem from pending Wishbone bus operations which can take multiple cycles to complete. Our LM32 was clocked at 62.5 MHz and when zoomed into the plot around 3 μ s, it becomes obvious that the distribution has 22 spikes with 16ns intervals. When the background task was removed, this degenerates to 2 spikes, completely removing the variability. The 3 μ s delay stems mostly from saving and restoring the register set. In principle, one could eliminate this cost using disjoint registers in interrupt and non-interrupt contexts.

From our measurements, both the time from interrupt to interrupt handler in Linux and the time from interrupt handler to userspace completion vary significantly. Unfortunately, with only 10000 samples, we could not trigger the worst case behaviour for both delays simultaneously. Adding the two worst-case delays measured separately, we predict that it should be possible for the Userspace delay to hit 120 μ s and FESA 280 μ s.

CONCLUSION

The measured times as presented in Table 1 must be reviewed in the context of different use-cases. As an example, ramping of magnets must be done synchronously. Here, a guaranteed synchronicity of $10-20 \,\mu s$ must be achieved for ring machines like the SIS18 and the SIS100. Another example is the control of kicker magnets, which requires at least 3ns precision and can only be done with FPGA Hardware Description Language (HDL). Software on the COM Express module may only be used for cases, where hard real-time is not required. As can be seen from the differences of minimum and maximum values, none of the solutions involving the CPU on the COM Express module fulfill those requirements, as long as the use of real-time Linux as operating systems is a stringent requirement for software tools like FESA.

For hard real-time the options are FPGA HDL or LM32 software. Here, FPGA HDL provides nanoseconds timing while LM32 software provides a better flexibility. To avoid stringent limitations for future developments of the FAIR

accelerator complex, standard FAIR equipment controllers like the SCU should be designed supporting hard real-time on the nanoseconds scale. If flexibility during runtime is required, the ideal solution could be a combination of both options, where LM32 software creates the action patterns that are phase aligned with high precision by FPGA HDL.

- [1] H. Ramakers et al., "Adaptive Control Unit for Digital Control of Power Converters for Magnets in GSI and FAIR Accelerators", GSI Scientific Report 2008, p. 117, http://www-alt.gsi.de/informationen/ wti/library/scientificreport2008/PAPERS/ GSI-ACCELERATORS-14.pdf
- [2] M. Kumm et al., "Realtime Communication Based on Optical Fibers for the Control of Digital RF Components", GSI Scientific Report 2007, p. 100, http://www-alt.gsi.de/informationen/ wti/library/scientificreport2007/PAPERS/ GSI-ACCELERATORS-14.pdf
- [3] P. Moreira, J. Serrano, T. Wlostowski, P. Loschmidt, G, Gaderer, "White rabbit: Sub-nanosecond timing distribution over ethernet", Precision Clock Synchronization for Measurement, Control and Communication, ISPCS 2009, D0I:10.1109/ISPCS.2009.5340196.
- [4] M. Arruat et al., "Front-end Software Architecture", Proceedings of ICALEPCS07, Knoxville, Tennessee, USA, p. 310
- [5] http://www.ohwr.org/projects/fmc-delay-1ns-8cha

DIAMOND LIGHT SOURCE CONTROL SYSTEMS RELATIONAL DATABASE

K. Vijayan, S. J. Singleton, M. T. Heron, Diamond Light Source Ltd, Oxfordshire, UK

Abstract

The functionality of the Diamond Light Source Control Systems Relational Database (RDB) is described in this article. An Oracle-based RDB and web-based GUIs allow recording of system configuration and configuration change management. Information about the hardware components that make up each beamline crate is stored in the RDB; for each item of control equipment, the status, location and name of the person responsible for the item are held. The Diamond Control System is based on EPICS [1] and has of the order of 500,000 process variables. The RDB maintains a record of the names of these PVs and validates new names against the Diamond Naming Convention, allowing consistency of naming style to be maintained and avoiding name duplication.

Machine operational details such as alarm logs are stored in the RDB and viewed using a web browser. All process data recorded by the control software are archived using the EPICS Channel Archiver; the archiver configuration for each technical area is maintained in the RDB. A further application using the RDB is the electronic logbook (eLog) which is used to record activities by Diamond Operations and Beamline groups.

INTRODUCTION

Diamond Light Source (DLS) is a 3 GeV thirdgeneration light source with a 561m storage ring (SR), a full-energy booster (BR) and a 100 MeV pre-injector Linac [2]. The photon output is optimised for high brightness from undulators and high flux from multi-pole wigglers. The current operational state includes 19 photon beamlines, with a further three beamlines in an advanced stages of design and construction. A further phase of photon beamlines has now been confirmed; detailed design and construction of these 10 beamlines started earlier this year.

The Diamond Control Systems Relational Database is based on the following software:

- Database: Oracle 11g Standard Edition;
- Entity Relationship Diagram: ER Studio;
- Language: PHP (Hypertext Preprocessor);
- Web server: Apache;
- Operating System: Red Hat Linux Enterprise Edition 5.

Development of the RDB was carried out in parallel with the Diamond Light Source design, so that it was available to be populated during the Diamond construction phase. Since then, it has undergone continuous development to meet new requirements as these arise.

DIAMOND NAMING CONVENTION

The RDB structure for the validation and recording of database names is shown in Figure 1. A device name can consist of up to a maximum of 17 alphanumeric characters. It can be broken down into 4 or 5 sections. The '-'character is used to delimit the sections and is the only non-alphanumeric character allowed.

The format is as follows:DD[SSS]-TT-CCCCC-NNWhere [...] indicates an optional section.DDDomainSSSSub Domain (Optional)TTTechnical AreaCCCCCComponentNNIdentifier



Figure 1: Database design for Naming Convention

The device name DD[SSS]-TT-CCCCC-NN uniquely identifies a device, such as a power supply, a gauge, etc. Properties associated with controlling or monitoring that device are identified by extending the device name with record specifiers. These define the signals associated with a device and the associated EPICS records carrying out processing and/or calculations.

ALARM LOG / ARCHIVER

The EPICS Alarm Handler (ALH) is used to log process variable alarms into text files. The contents of these files are automatically imported to the RDB in order to allow them to be viewed using a web browser.

All process data recorded by the control software are archived using the EPICS Channel Archiver. Redundancy is provided by the use of two identical servers which archive data in parallel, though one server is designated the primary archiver and the other as the standby. The archiver configuration for each technical area is maintained in the RDB. This gives users the flexibility to import their individual technical group files to the RDB using a web page (Figure 2). Following an import, a script then restarts the selected archiver process in both archive servers with the updated group file.

😣 Welcome to Diamond Relational Database ::PF	IODUCTION:::::: K.Yijayan - Mozilla F	irefox		
Ele Edit Yew History Backmarks Whool Jock Eleb				
🕐 Minde of Stand Siland Siland			and so it is a second	
🗧 🗧 🎴 🔒 https://ekip.pri.damond.ac.uk/ph	olloginics, framepage php		🕱 🗇 🕾 C 🛃 - Google	P 6 23
	2	Imp	ort Archive File	
boomsib 🥂	Import to	BURT Archiver		
Cioninono	Archive Group File Name *			Browse.
K.Vijavan	Restart selected archiver			
	Replace existing group file	-		
B-GEPICS	(Applicable only for Beamline Archivers)			
Device Entry				
Device-DB Information	Import			
Device Location	Note: File format should be:			
B-GArchive	For Monitor: <pv name=""> <period> <monito< td=""><td>P</td><td></td><td></td></monito<></period></pv>	P		
Import Archive File	For Scan:			
Export Archive File	CPV Halles CPellous			
Archive Record				
Beamline Entry				
Beports				
🖶 🕒 Admin				
🕘 🛅 Equipment				
🕮 🛅 Cable				
😑 🧰 Elog				
Detector Organized & Allgement				
Concrations				
Logout				
The wildcard character '%' may be used in search options				

Figure 2: Import Archive File

EQUIPMENT TRACKING DATABASE

The equipment tracking database stores all information about each item of hardware equipment, such as the manufacturer name, the model, the type of the equipment (crate, IP carrier, etc.). Each equipment item is uniquely identified by its equipment ID in the database. For each equipment item, the status, location and name of the person responsible for the item are also held in the database.

This tracking system identifies the present physical location of each item of equipment together with its history of transfer.

If an equipment item becomes faulty, the fault can be registered using the fault registration web page in the database. Once the fault has been corrected, either by the manufacturer or internally within Diamond, the details of the repair can be stored in the database. Thus a full history record for the equipment item is maintained in the database.

The hardware components that make up each control crate are stored in the RDB. Users can easily modify the crate information using a web browser (e.g. Figure 3). The equipment tracking system has the facility to move all the equipment within one crate to another, for example when a crate is replaced.

Welcome to Diamond Relational Database :::::::#	800UCTION:K.Vijayan - Mezilla Firefox
Ejle Ejdit Vjew Hiştory Bookmarks Vjahool Tools Help	
😌 visitame to Diamana Relational Datations : 🕴	
🗧 🕘 🗧 🖻 🔒 https://do.pri.demond.ec.uk/pt	nlaphics franceses ato 😨 🏠 🕫 🖉 👌 🖉 🗸
	View Beamline Crate Information
👥 diamond	Reamine Crate DCC0000824
, in the second s	Date details
K.Wijayan	Equipment ID Device ID Manufacturer Name Model No Equipment Type Current Locabin Owner
🕞 📋 EPICS	DCSD1008364 BL161-EA-CRATE-01 ELMA EUK10914 CRATE 11U21SLT DLS R79
e- 🗋 Admin	Front Bank
E- Contraction Contraction	DCS0004802
B- BRuipment Entry	1 WECARDICOCIMUMESSOO PMCCARDI <u>IIMCEVR</u> 1
<u>Equipment</u>	2 2
- <u>Warranty Expired</u>	3 3 Dr500003122 Dr500033413
<u>Lego Service/Lailoration Expred</u>	4 TRANSCARD[DT0]8001 4 TRANSCARD[DT[8991
- <u>The vincepp</u> - <u>View Beamine Crate</u> - <u>Beamine Crate-Tree View</u> - <u>B</u> Equipment Transfer - <u>Fault Begistration</u> - <u>Beamine Crate-Tree View</u>	* CONTROLLES * CONTROLLES * PROCEETINGS * CONTROLLES *
Help document Cable Cable Cabl	A CONVERSE 6 VIECCANDIFICAREERING C CONVERSE 9 VIECCANDIFICAREERING C VIENCULAUCIDATION 1 VIECCANDIFICAREERING C VIENUULAUCIDATION 1 VIECANDIFICAREERING C VIENUULAUCIDATION 1 VIECANDIFICAREERI
🕞 📋 Operations	7 00501014750 7 WECARDININISISS20-64M-NIN
The wildcard character W may be used in	8 DESIDENTIALS 8 DESIDENTIALS 8 DESIDENTIALS 8 DESIDENTIALS 8 PRIMARAGOLE (SEGULT
search options	9
	10 10
	12 12
	13 13
	14 14
	15 15
	16 16
	17 VNECATO [1976
	18 18
	19 19
	20 20
	21 21

Figure 3: Beamline Crate Information

CABLE DATABASE

The Cable Database maintains information about all cables used in the storage ring and other areas of Diamond. Each cable is uniquely identified in the database by its cable ID. The naming convention for cable IDs is as follows:

V12-3456P

V	Technical Area (Electrical Discipline)
12	CIA Identification (EPICS name)
3456	Cable Number
Р	Cable Category

The information stored includes cable type, start location, end location, run length, etc. A history of changes to cable information can be viewed from the database. Also there is a facility to roll back the database to get information on a deleted cable.

ELOG

Electronic logbooks are used by the members of the Diamond Operations team and several Beamline groups to record their activities in the storage ring or beamlines. Each group has its own logbook, the structure of which is shown in Figure 4. eLog messages have a title, a message body and a category. The content of the message body is based on HTML, and so can include normal text, images and URL links. An eLog entry can be made in many ways: using the eLog web client, via Microsoft Outlook Email, by means of Blog editors or by setting up automated entries.

The eLog web client allows users to save their logs as draft and preview their draft entries before publishing. Using AJAX (Asynchronous JavaScript and XML) the content is saved automatically every minute without user intervention.

Once published, an eLog entry cannot be modified or deleted. A facility is provided to allow a user to replace an incorrect entry; the replacement entry is then marked as such and provides the means to view the original entry.

Web log (Blog) editors such as Windows Live Writer (WLW) or Classic ScribeFire can also be used to make entries to a logbook.



Figure 4: Logical diagram of eLog interface

By using the ink Blog add-on to Windows Live Writer (Figure 5) users can import handwritten pages and incorporate them in entries to the logbook.

Possible future developments of eLog include:

- A facility to drag and drop images from file explorer to the eLog entry compose page.
- An eLog app to view a logbook and to create new entries for Android devices.



Figure 5: Handwritten entry to eLog using Windows Live Writer

CONCLUSION

The Diamond Control System RDB played a significant role in the successful construction of Diamond, and continues to provide valuable support to Operations and maintenance staff, and those responsible for the development of new beamlines.

By using this Control System RDB, system engineers can easily archive their EPICS PVs, create device names that comply with the Diamond Naming Convention and uniquely identify and track system hardware equipment. Using eLog all users are able to record their work and to view and search the logbook for entries containing information relevant to them.

- [1] Experimental Physics and Industrial Control System; http://www.aps.anl.gov/epics
- [2] R. P. Walker, "Commissioning and Status of The Diamond Storage Ring", APAC 2007, Indore, India.

CURRENT STATUS AND UPGRADE PLAN OF THE DATA-ACQUISITION SYSTEM AT SACLA

T. Sugimoto, A. Amselem, Y. Furukawa, T. Hirono, Y. Joti, T. Kameshima, A. Kiyomichi, T. Ohata, M. Yamaga, R. Tanaka, JASRI/SPring-8, Sayo, Hyogo 679-5198, Japan
T. Abe, A. Tokuhisa, T. Hatsui, RIKEN SPring-8 Center, Sayo, Hyogo 679-5148, Japan

Abstract

User experiments at the SPring-8 angstrom compact free electron laser (SACLA) facility were recently commenced, in March 2012. We have developed a dedicated dataacquisition system that can be used for user experiments. This system is currently capable of acquiring data from 10 multiport charge-coupled device sensors at a data rate of up to 5 gigabits per second (Gb/s). In this paper, we present an overview of this data-acquisition system.

OVERVIEW OF SACLA

The SPring-8 angstrom compact free electron laser (SACLA) facility is an X-ray free electron laser (XFEL) facility located at the SPring-8 site in Japan. This facility is characterized by its compact design; it is about 700-m long and includes the accelerator, undulator, and experimental buildings. To achieve the compact XFEL design, we performed the relevant technical development activities such as of the C-band accelerating cavity, in-vacuum undulator, timing, optics, detectors, and data-acquisition system. A significant advantage of the SACLA-SPring-8 experimental facility is that we can use two X-rays, one from SACLA and one from SPring-8, simultaneously. The first self-amplified spontaneous emission (SASE) lasing of SACLA was achieved in June 2011, and X-ray laser beams have been delivered to users since March 2012 [1]. XFEL wavelengths in the subangstrom region (0.6 Å) have been realized. During the first experimental period from March to July 2012, 25 experiments covering atomic, molecular, and optical physics, ultrafast science, material science, and structural biology were carried out.

OVERVIEW OF THE DATA-ACQUISITION SYSTEM

We have developed a dedicated data-acquisition (DAQ) system that can be used for user experiments at SACLA. The DAQ system consists of a number of components: detectors, a front-end system, data-handling servers, a cache storage system, an event-synchronized database (DB), a high-performance PC cluster (HPC), and a large-bandwidth network. We chose standard interfaces to connect components with each other. To match the needs of future experiments, the DAQ performance will be upgraded by replacing the DAQ system components. Figure 1 shows an overview of the SACLA DAQ system. In this section, we describe the current status of each component.



Figure 1: Overview of the DAQ system. The current DAQ system supports up to 10 multiport charge-coupled device sensors.

Multi-Port Charge-Coupled Device Sensors

The multi-port charge-coupled device (MPCCD) sensor is a two-dimensional X-ray detector developed for SACLA experiments [2]. Each pixel of the MPCCD is $50 \times 50 \ \mu m^2$ with a 16-bit data depth. A single MPCCD sensor module has 512×1024 net pixels. Inclusive of calibration data, 512×1032 gross pixels per shot are acquired by the sensor at a repetition rate of 60 Hz. MPCCD sensors can be used in one of the following typical configurations: 1 (single), 2 (dual), and 8 (octal) sensors. We decided to use either the single configuration or the octal + dual combined configuration. For the single and combined configurations (1 and 8 + 2 sensors), data rates are about 500 megabits per second (Mb/s) and 5 gigabits per second (Gb/s), respectively. We have developed downstream components for the DAQ system to satisfy such high data rate requirements.

We chose Camera Link [3] as the digital interface between the MPCCD sensors and the front-end system. By standardizing the front-end interface as the Camera Link, not only MPCCD sensors but other commercial cameras such as IMPERX [4] and OPAL [5] can also be used. The Camera Link interface provides a bandwidth > 2 Gb/s (base configuration), which satisfies the data-rate requirements of the MPCCD sensors.

Front-End System

Because the physical range of Camera Link is short (several meters), the front-end system is aimed at receiving and transferring data from Camera Link to Ethernet. We developed two different front-end systems [6]: a PC-based system and a VME-based system. The PC-based system is a one-rack-unit Linux PC equipped with a commercial Camera Link grabber board. The advantage of using the PC-based system is its flexibility in supporting more than 100 commercial cameras, including Camera Link full configuration cameras. The VME-based system has a dedicated field programmable gate array (FPGA) processor to forward image data from Camera Link to Gigabit Ethernet (GbE). The advantage of the VME-based system is its efficient hardware processing. We implemented on-thefly lossless image compression using the FPGA processor. The I/O interfaces are implemented with a processor PCI mezzanine card and will be upgraded in the future. For practical application, we chose a better system for each detector; PC-based and VME-based systems are used for the OPAL-2000 camera and MPCCD detector arrays, respectively.

Data-Handling Server

Data-handling servers are inserted between the front-end system and the storage system. Each data-handling server corresponds to an individual sensor. The data-handling servers aim at in-line buffering and processing. We implemented a pipeline fast-in-fast-out buffer system on the data-handling server. The pipeline system keeps the DAQ system stable against network packet loss.

We also implemented an on-the-fly low-level filtering system on the data-handling server. The first implemented low-level filtering is a grid-based region-of-interest (ROI) statistical analysis. Figure 2 shows a graphical user interface (GUI) image of the filtering system. The region to the left shows MPCCD image data, and a histogram of the ROI average photon number is shown in the lower right corner. In the current filtering system, each grid is defined as a 4×8 region. The average photon number of each grid is calculated by the data-handling server, shot by shot. The average photon numbers of the grids are recorded on an event-synchronized DB. By choosing a region of interest with the GUI, primitive image-pattern selection can be performed without the need to read the entire image from the storage system.

Storage System

Experimental data are accumulated in three types of tiered storage systems, whose specifications are detailed in Table 1. Tier-1 systems aim at first-level data caching. We suppose that the capacity requirement for the Tier-1 system is about 200–300 terabytes (TB) per week, which is estimated from the data rate of the MPCCD sensors (10 sensors), each with a 50% duty factor. Storage capacities of Tier-1A and Tier-1B systems are 200 and 250 TB, respectively. We intend to use the two systems in rotation; for example, while one system is used for data readout to the Tier-2 system, the other system is used for data acquisition without performance degradation. We chose single-namespace file system (StorNext and GPFS) for the Tier-1 systems, because the numbers of sensors can vary in ev-



Figure 2: GUI image of the low-level filtering system. The region to the left shows MPCCD image data. A histogram of the ROI average photon number is shown in the lower right corner.

ery experiment. The minimum guaranteed throughput of the Tier-1 system is 70 megabytes per second (MB/s) with 10 simultaneous streams, which satisfies the required data rate of the MPCCD sensors (60 MB/s, 8 + 2 sensors). It should also be noted that the average throughput of the Tier-1 system is 500 MB/s, which satisfies the estimated data rate of the silicon-on-insulator photon imaging array sensor (SOPHIAS), which is under development.

Tier-2 and Tier-3 storage systems will be installed in March 2013. Tiers 2 and 3 aim at long-term data archiving (over more than one year). The Tier-2 system consists of a 1 petabyte (PB) disk array having more than 2 gigabytes per second (GB/s) throughput. The Tier-3 system is an automated tape-library system equipped with 12 tape drives (IBM TS1140) and 7500 tape-cartridge slots (IBM 3592 specifications). The initial capacity of the tape library is 6 PB with 1700 cartridges. The average throughput of a single drive is 200 MB/s. By using GPFS and hierarchical storage management software (TSM), Tier-2 and Tier-3 are united into a single-namespace file system.

Event-Synchronized Database

The XFEL is a discrete beam operating at a 60 Hz repetition rate. To distinguish event data corresponding to each X-ray shot, we developed an event tag system and an event-synchronized DB system [7]. The timing signal is distributed from the XFEL master trigger at 60 Hz. The front-end system utilizes the timing signal both for acquiring the trigger and for shot-number counting. Acquired data from beam-line monitors are tagged by the shot number and are recorded on the event-synchronized DB. An experimental user can acquire beam monitor data from the event-synchronized DB by specifying a certain tag number or timestamp.

Tier	Capacity	Throughput	Hardware	Media	File System
1A	200 TB	70 MB/s (min.) ×10 streams	DDN S2A9900	HDD (SAS)	Quantum StorNext
		500 MB/s (avg.) $\times 10$ streams			
1B	250 TB	$70 \text{ MB/s} \text{ (min.)} \times 10 \text{ streams}$	DDN SFA10000	HDD (SAS)	IBM GPFS
		500 MB/s (avg.) $\times 10$ streams			
2	1 PB	>2 GB/s	DDN SFA10000	HDD (SATA)	IBM GPFS
3	6 PB	200 MB/s (avg.) \times 12 drives	IBM TS3500	Tape (3592)	IBM GPFS and TSM

Table 1: Specifications of the storage systems. Tier-1 systems are used for first-level data caching. Tiers 2 and 3 are used for long-term data archiving.

High-Performance PC Cluster

Because the obtained images from X-ray diffraction experiments are reciprocal lattice spaces, a sophisticated calculation is necessary to extract real-space images. For these analyses, an HPC system has been installed in the SACLA computer room. Performance of the HPC system is 13 TFLOPS with 170 TB of shared storage system. The HPC system is used not only for off-line analysis but also for online instant visualization. Real-space images from on-line reduced analyses help to determine the feasibility of carrying out the experiment. By examining these images, we can determine whether the experimental conditions need to be changed.

Large-Bandwidth Network

The SACLA experimental network consists of three local-area networks (LANs) [8]: DAQ-LAN, DAQ-USER-LAN, and HPC-LAN. In this paper, overviews of DAQ-LAN and HPC-LAN are presented.

The DAQ-LAN is a dedicated network for the dataacquisition system. The DAQ-LAN has two physical network layers: a 10-GbE layer for data transfer and a 1-GbE layer for instrumental control from the user terminal. We segregate the two physical backbones to ensure a high bandwidth of data transfer. We apply link aggregation for the purpose of redundancy. The DAQ-LANs (10 GbE and 1 GbE) are available at every experimental hutch and every experimental station.

We also have another dedicated LAN for data analysis: the HPC-LAN. The HPC-LAN is a backbone network for data exchange among Tier-1, -2, and -3 storage systems and the HPC system. The HPC-LAN is also connected to a 10-Gb wide-area network for the purpose of cooperative analysis using external supercomputers.

UPGRADE PLAN FOR THE DATA-ACQUISITION SYSTEM

We are developing a next-generation X-ray sensor, referred to as SOPHIAS [9]. This sensor is being developed to achieve a higher dynamic range than the present \ge MPCCD sensors. A single SOPHIAS module has $1024 \times$ 2048 net pixels with a 32-bit data depth. The SOPHIAS dea tector will have maximum sensor number of 40. The maximum total data rate is about 20 GB/s at a 60-Hz XFEL repetition rate. Operation of the first SOPHIAS detector with a two-sensor configuration will begin in 2014. To satisfy the data rate of SOPHIAS (480 MB/s for a single sensor), we started developing a new front-end system [10]. According to our plan, the new front-end system supports data rates of up to 2.5 GB/s for the future 300-Hz repetition rate. We are also examining the requirements of the data-handling servers and DAQ-LAN.

We also plan to perform sophisticated on-line analysis using external supercomputers. For the coherent X-ray diffraction imaging experiment [11], our HPC system is insufficient for performing on-line instant visualization. By utilizing external peta-FLOPS class supercomputers, such as the "K computer" [12], we will be able to refine experimental conditions quickly during an ongoing experiment.

SUMMARY

User experiments with SACLA were recently commenced, in March 2012. A dedicated DAQ system has been developed for the user experiments, and this system is now fully operational. We started developing new DAQ components to support the next-generation X-ray sensor SOPHIAS. In 2014, the new sensor and the upgraded DAQ system will come into operation. We have also started developing a cooperative analysis procedure using external supercomputers.

REFERENCES

- [1] T. Ishikawa et al., "A compact X-ray free-electron laser emitting in the sub-angstrom region," Nature Photonics 6, 540 (2012).
- [2] T. Kameshima et al., "Development status of X-ray 2D detectors for SPring-8 XFEL," IEEE Nuclear Science Symposium, Knoxville, TN, USA, October 2010.
- [3] Camera Link(R), Automated Imaging Association, http:// www.visiononline.org/.
- [4] IMPERX Camera Link Cameras, http://www.imperx. com/.
- [5] Adimec OPAL-2000 Camera, http://www.adimec.com/.
- [6] A. Kiyomichi et al., "Development of image data acquisition system for 2D detector at SACLA," Proceedings of ICALEPCS2011, Grenoble, France, 947 (2011).
- [7] M. Yamaga et al., "Event-synchronized data acquisition system of 5 giga-bps data rate for user experiment at the XFEL facility, SACLA," Proceedings of ICALEPCS2011, Grenoble, France, 581 (2011).

respective authors

 $\mathbf{\Theta}$

- [8] T. Sugimoto et al., "Large-bandwidth data acquisition network for XFEL facility, SACLA," Proceedings of ICALEPCS2011, Grenoble, France, 626 (2011).
- [9] T. Hatsui et al., "Development and deployment status of Xray 2D detector for SACLA," International Workshop on Semiconductor Pixel Detectors for Particles and Imaging (PIXEL2012).
- [10] C. Saji et al., "Development of data-acquisition front ends enabling high-bandwidth data handling for X-ray 2D detectors: a feasibility study," International Workshop on Semiconductor Pixel Detectors for Particles and Imaging (PIXEL2012).
- [11] K. J. Gaffney and H. N. Chapman, "Imaging atomic structure and dynamics with ultrafast X-ray scattering," Science 316, 1444 (2007).
- [12] K computer, RIKEN Advanced Institute for Computational Science, http://www.aics.riken.jp/; the K computer is the third fastest supercomputer in the world as of November 2012, http://www.top500.org/lists/2012/11/.

THE IUAC TANDEM-LINAC CONTROL SYSTEM

Ajith Kumar, E.T. Subrahmaniam, K. Singh, B. Sahu, D. Munda Inter-University Accelerator Centre, Aruna Asaf Ali Road, New Delhi, India

Abstract

The 16MV Tandem Van de Graaff accelerator at IUAC is one of the earliest machines to go for a PC based control system. The PDP11, supplied with it, was replaced by an IBM PC running DOS before the accelerator was commissioned in 1989. The present system, commissioned in 1997 and supports LINAC, runs on a network of PCs under the GNU/Linux operating system. Every parameter to be controlled/monitored is assigned a unique identifier consisting of a Label, Function and Unit. The signals are grouped according to location and they are connected to server PCs using interfaces like CAMAC, VME or custom hardware. The features like a user interface, alarm systems, data logging and partial automation are handled by several client programs, communicating to multiple servers to access the hardware over a TCP/IP connection.

INTRODUCTION

Inter-University Accelerator Centre has a 16 MV Tandem Van de Graaff accelerator and a super-conducting heavy ion LINAC. The Tandem was originally supplied with CAMAC Serial Highway Interface and a DEC PDP11 based control system. The PDP11 was replaced by a PC based system running DOS before the accelerator was commissioned [1]. Later on, the addition of the LINAC demanded features like multiple operator consoles and the ability to run special purpose programs to condition the resonator cavity, automatically setting the amplitude and phase of RF etc. To meet the new requirements, a new control system was developed with the following basic guidelines; low implementation and maintenance costs, reliability, flexibility and scalability. PCs connected over Ethernet, running GNU/Linux operating system, forms the basic computer network required for the system. CAMAC, VME and custom hardware are supported.

DESIGN

The main task of the accelerator control system is to monitor/control signals from various devices like power supplies, vacuum systems, beam monitors an so on. Each device may have one or more analog/digital signals to be controlled or monitored, resulting in a large pool of signals. We have assigned a unique identifier for each signal. Three character strings, Name, Function and Unit, uniquely identify each parameter in the whole system. For example "CPS031", "VC", "KV" identifies the Charging Chain power supply voltage control signal. This nomenclature is followed mainly because our accelerator people were already familiar with it. The signals are grouped according to location and each group is connected to server PC, closer



Figure 1: Schematic of Control System Hardware.

to the devices, using interfaces like CAMAC, VME or custom hardware.

The computers connected to the hardware runs a server program supporting the control/monitor of the signals over a TCP/IP connection. Only the unique signal identifier is known outside the server, the hardware details and the complexities of all the signals are hidden from the outside world. A simple message passing protocol, over a TCP connection, supporting functions like getState, set-State, getValue and setValue are used for accessing the signals from remote computers running client programs.

HARDWARE

In the accelerator we have devices requiring analog and digital type signals for controlling/monitoring. Signals from devices located nearby are grouped and connected to one computer, using CAMAC, VME or some custom hardware interface. Several such groups are required to take care of all the signals. Each signal is characterized by a unique name and interfacing details like hardware address and datasize. Most of them are handled using standard modules like ADC, DACs, Output Registers and Input Gates. Special modules were made for interfacing the Beam Profile Monitor and the LINAC resonator controllers. The CAMAC Crate Controller used in the system are equipped with a built-in single board computer having PC104 interface and Ethernet port. On power-up the CAMAC controllers boot from a central server, using the Linux Terminal Server software. The VME Crate controllers also have embedded PCs inside them with Ethernet interface. All of them are connected to the same Ethernet network.

Computers providing the user interface are equipped with four shaft encoder knobs, acting as incremental input devices, for setting the value of analog signals. They are



Figure 2: Screen-shot of operator console program.

connected to the USB port of the PC using custom designed hardware. A schematic of the control system hardware is shown in Figure 1.

SOFTWARE

We have followed a distributed approach and each server has a database storing the details of signals connected to it, there is no centralized database. The databases contain static and dynamic fields of information about the connected signals. The static fields are loaded from a file during startup and the dynamic fields are periodically updated from the hardware. All servers listen over the network, using TCP protocol, for client requests and establishes client connections, after authentication. The only way to access any signal is through a TCP/IP socket, even if it is from the same computer. A server can establish connections to several clients at the same time.

The Client programs provide the user interface, alarm services, logging etc. Every client program works on a set of signals, usually located at different servers. During startup, the required server addresses and the signal identifiers are loaded from a file and the server address of every signal is identified by searching for it, using the identifier. After that, the client program accesses the signals by sending command packets to the appropriate server. The packet will have the signal identification and the operation to be carried out and the reply will contain the requested information. The message packet implements functions like get-State, setState, getValue and setValue, for managing analog and logical type signals.

A client program could be a small one just recording a single parameter or one that can be used for tuning the beam through the entire accelerator. A screen-shot of the most commonly used client program is shown in Figure 2. This GUI is implemented using X-Windows and Motif library. The total number of parameters is grouped into "Pages" for the convenience of displaying. The user defines the page layout by editing text files, which will have the parameters identified by the signal identifier mentioned earlier. All the available pages are loaded during startup and the server for each parameter is identified. The user requests the client to access the various parameters by clicking the mouse and by turning the Shaft Encoder knobs. Client sends requests to the servers for getting and setting various parameters controlling the operation of the accelerator and displays the results. Dependencies between parameters can be defined in a text file that is loaded during startup and the relationships examined periodically to take necessary actions. The interlocks, alarm and a diagrammatic display of the machine are currently implemented using this facility.

The Python Library

Python is an interpreted language with a simple syntax and it is relatively easy to learn. Providing a Python library to access the accelerator parameters enables people working on the accelerator development to write simple scripts for various applications [3]. The unique identifier for each signal makes this task very easy. For example, the Python program given below prints the filament current of the SNICS ion source.

import pelcon
p = pelcon.pserv()
print p.get_value('FILSN1', 'CR', 'A')

The program can run from any computer connected to the control system network. The only information required is access the signal is the identifier consisting of the three character strings. The controlling/monitoring are done mostly using the four functions, get_value(), set_value(), get_state() and set_state(). Multiple clients may try to access the same parameter but the server serializes the requests to avoid any conflict. The Python library is being used for developing programs for data logging and partial automation of the accelerator operation [4].

It is very easy to write GUI programs in Python using one of the toolkits like Tkinter, pyGTK ot PyQt. The introduction of Python language to the control system was due to our experience with it in another project, described in the next section.

RELATED DEVELOPMENTS

The function of an accelerator control system is very similar to that of the computer interfaced laboratory equipment employed in teaching science, the technologies used are more or less the same. We have developed a very low cost hardware device that provides several analog/digital channels where sensor/control elements can be connected. It consists of a micro-controller with a USB interface and some signal processing circuits. The design schematic is shown in Figure 3. The micro-controller runs a server program that listens for the commands coming from the computer over a USB link, performs the requested actions and sends the response back to the PC. The measurements, requiring real-time capability, are done by the



Figure 3: Schematic of expEYES design.



Figure 4: expEYES working as an oscilloscope in the audio frequency range.

micro-controller running a C program. The data analysis and the user interface, written in Python or C, runs on the PC. A hardware communication library and several GUI applications are written in Python to access the sensor/control elements connected it.

The device is called expEYES and it supports more than fifty experiments starting from high school level. The design is open and the royalty-free hardware is currently available for around US\$ 30/-, from several vendors. The hardware is shown in Figure 5 and a screen-shot of it working as a low frequency oscilloscope is in Figure 4. The details are given on the website, http://expeyes.in [5].

ExpEYES hardware has been used for measuring the frequency error signals from the LINAC resonators. It has been integrated into the system using the same scheme used for CAMAC and VME. This system can be used by engineering students to practice the basics of computer interfacing and some of the techniques used in writing control systems.



Figure 5: Photograph of expEYES hardware.

CONCLUSION

The distributed control system, using a simple message passing protocol, was operational from 1997. In the beginning only CAMAC interface was used but it was easy to add support for other hardware interfaces due to the distributed design. The addition of Python library has made program development much more easier and faster. The control system has demonstrated very high reliability,scalability and ease of use. The development work also resulted in some spin-off products that are now being used for teaching science.

- Ajith Kumar et al., "The NSC 16 MV tandem accelerator control system", NIMA 343(1994) 327 – 330.
- [2] Ajith Kumar et.al., "A distributed Control System for NSC Tandem-LINAC", PCaPAC2005, Hayama, Japan, 22-25 Mar 2005.
- [3] Ajith Kumar, "A Python based interface for the Tandem-LINAC control system", INPAC 2011, IUAC, Delhi, Feb., 15-18, 2011
- [4] B.K. Sahu et.al., "Automation of the Control Scheme for IUAC LINAC", INPAC 2011, IUAC, Delhi, Feb., 15-18, 2011
- [5] http://expeyes.in

AN UPDATE ON CONSYS INCLUDING A NEW LABVIEW FPGA BASED LLRF SYSTEM

T. Worm, J.S. Nielsen, ISA, Aarhus University, DK-8000 Aarhus, Denmark

Abstract

ConSys, the Windows based control system for ASTRID and ASTRID2, is now a mature system, having been in operation for more than 15 years. All the standard programs (Console, plots, data logging, control setting store/restore etc.) are fully general and are configured through a database or file. ConSys is a standard publisher/subscriber system, where all nodes can act both as client and server. One very strong feature is the easy ability to make virtual devices (devices which do not depend on hardware directly, but combine hardware parameters.)

For ASTRID2 a new LabVIEW based Low-Level RF system has been made. This system uses a National Instruments NI-PCIe 7852R DAQ card, which includes an on-board FPGA and is hosted in a standard PC. The fast (50 kHz) amplitude loop has been implemented on the FPGA, whereas the slower tuning and phase loops are implemented in the real-time system. An operator interface, including live plots from the regulation loops, are implemented in a host program on Windows. All three levels have been implemented with LabVIEW. The LLRF system is interfaced to ConSys through LabVIEW shared variables.

STATUS OF CONSYS

ConSys was originally made as a new control system for the storage ring ASTRID in 1998 [1-3]. It is now used at several machines at the University of Aarhus, which is including the new storage ring ASTRID2 [4] presently being commissioned. ConSys is also used at Stockholm University to control DESIREE and previously CRYRING as well.

The control system follows the standard model for a distributed control system using the publisher/subscriber method. It is running on Windows and written in Microsoft Visual C++ using MFC base classes. The design of the system is based on an object oriented programming strategy and the core system has been stable for many years. In ConSys the same core software is running on all computers - operator console PC's as well as frontend computers/device servers. Almost all configurable data are stored centrally in a SQL database: This includes all address information needed to locate devices and parameters, scaling data for conversion between binary and physical information, and display information used by a general purpose operator console application to generate operator pages for control and monitoring.

ConSys Devices

Data are collected and controlled on frontend computers through ConSys devices [1]. A large number of different hardware can be controlled and monitored by ConSys. Some hardware is controlled by dedicated devices where the device interfaces needed are coded directly into ConSys. Other equipment is controlled through analogue and digital I/O through general hardware. These kinds of hardware are serviced by standard ConSvs devices where all configurable information is stored in the SOL database. At ASTRID we use home built equipment, as well as industrial systems. In the original control system all general control was done through home built so-called G64 systems - small CPU crates with general purpose analogue and digital control including table based ramping of the machine. Most of these systems are still running today. Newer systems with need for general I/O are now based on Beckhoff PLC (Programmable Logical Controller) systems. PLC systems are relatively cheap, reliable and flexible systems for slow control and data acquisition. Simple automation codes are in some cases made directly in the PLC controller. Examples of programs implemented in PLC vacuum baking control. systems are: vacuum systems, personal interlock/safety safety system, undulator safety system and general averages of ADCs. All PLC systems are connected via Ethernet and serviced by a general ConSys PLC device for monitoring and control. As with our existing general devices all address and conversion information needed by ConSys to service the PLC's is stored in the central SQL database.

Another kind of general purpose I/O device is an interface device to LabVIEW shared/network variables. ConSys supports read and write of simple parameter types (Boolean, Word, Floating point) to shared variables. The shared variable names, location and data types are stored centrally in the SQL database. This device is used with NI CompactDAQ and CompactRIO systems as well as the LabVIEW based RF system presented later in this article.

A similar device is used to interface EPICS channels. The ConSys to EPICS channel interface is used for control and monitoring of the ASTRID2 beam position system, which is based on Libera Electron Beam Position Processors running an EPICS device.

Some hardware have embedded processors with control and monitoring through various interfaces. For this kind of hardware, specialized ConSys devices are implemented. The implementation is made so several instances of the same kind of hardware can be controlled by the same device code. This implies that all instance



Figure 1: Overview of the ASTRID/ASTRID2 RF systems.

Automations

related information, like addressing, is stored in the SQL database. Examples of this kind of devices are power supplies, vacuum gauge controllers, pump controllers and motor controllers. Preferably all devices are connected directly to Ethernet. However, a lot of hardware only has serial port communication. To bring this kind of equipment to Ethernet, Moxa serial port TCP server boxes are used.

ConSys Parameters

In ConSys a parameter is defined as a single simple I/O point of a basic value type like a Boolean, a Word or a Floating point value. An I/O point can be a physical input or output from a piece of hardware or a virtual value stored in the memory of a ConSys device. Parameters are grouped into clusters with all parameters in a cluster having a common cluster name. Each parameter in the cluster has a specific name, in ConSys called a surname. The name of a parameter in ConSys is specified by <cluster name>.<sur name>, like 'BMH100IPSast2.Iw'. A cluster typically relates to a specific piece of hardware, like a power supply, and hence parameters in the cluster are all connected to the same ConSys device. There is, however, no binding on the source of parameters in a cluster. Parameters in a cluster can come from several different devices. This is useful in situations where calculated parameters from automation devices can be mixed with physical values from the device servicing the hardware. A simple example of this is a power supply for a magnet: physical values like current and voltage from a hardware device can be grouped in the same cluster as calculated values for power and magnet field from a calculation device.

Any ConSys parameter can be accessed from any machine - even other frontend computers. This feature is used in automation processes. ConSys strongly supports implementation of automations and many automatic reactions/functions/calculations are implemented at ISA. An example is synchrotron radiation operation: the steps involved in operation are heavily automated. This includes copying different settings for store and accumulation, automatic RF power control, etc.

ConSys Applications

ConSys has a set of generic applications for display, control and automation. The 'Console' program is the main program to display and adjust parameters. It is machine independent - all console pages are fully defined in the SQL database. A console page is built from a selection of clusters, or subset of clusters. Graphic pages based on any graphical background with overlaid parameters are also possible. An important feature of the console is the two control bars, where analogue parameters can be dragged to and controlled in several ways – of which the control through a small homebuilt panel with two digital potentiometers for an 'analogue feel' is the most convenient.

The 'Datalogger' program is used to log ConSys parameters to the SQL database. The parameters to be logged are collected into parameter groups. Each group is logged at fixed user defined intervals. Conditions for logging can be set based on ConSys parameters. z

'CSPlot' is the general purpose plot program for ConSys. CSPlot can plot any parameter, and have an unlimited number of graphs in each window. At start up, history buffers are obtained from ConSys devices. All ConSys devices store read and write histories for almost
all parameters in memory. Typical history lengths are 2 weeks of data (with data being compressed/averaged into larger and larger bins the older they get). If a parameter is logged in the database by the datalogger, the history for that parameter can be extended by reading data back from the SQL database.

LOW-LEVEL RF SYSTEM

The storage rings in Aarhus, ASTRID and ASTRID2, are fitted with a 105 MHz RF system. A new RF system was needed for the new synchrotron light source ASTRID2. Since the low level RF (LLRF) system for ASTRID was quite old and difficult to maintain, it was decided to start by updating the ASTRID LLRF and make an almost identical system for ASTRID2. The systems use a digital control of baseband signals, with IQ demodulation for measurement, and direct amplitude and phase control for regulation. The control is done using a DAQ card which samples the baseband signals using ADCs and provides the control voltages using DACs. All software is programmed in National Instruments LabVIEW (version 2012).



Figure 2: LLRF Software layers.

RF Hardware

Figure 1 shows an overview of the ASTRID/ASTRID2 RF systems. Input from a 105 MHz master oscillator is split into several lines. One part is sent through a voltage controlled phase-shifter, a voltage controlled attenuator and a low-power amplifier, before driving the high power amplifier. To be able to detect both amplitude and phase (full 360°) of the cavity pickup and forward power, IQ demodulators are used. The outputs of the IQ demodulators are at baseband (i.e. in principle DC signals). Amplitude detection of other relevant signals (reflected power, drive power, etc.) are done using simple power level detectors.

Tuning of the cavity on ASTRID is done with two plungers moved by DC motors. On ASTRID2 tuning is done by adjusting the cavity end plate using a stepper motor and appropriate mechanics.

Data Acquisition

The digital control is based on a standard industry PC with an Intel Core I7 processor. The system is equipped with a fast DAQ card - a National Instruments PCIe-7852R, with an onboard Virtex-5 LX50 FPGA, 8 AI, $\pm 10V$, 16 bit, 750 kS/s/ch; 8 AO, $\pm 10V$, 16 bit, 1 MS/s/ch. This fast card samples (500 kS/s) all fast varying signals (cavity pickup (IQ), forward power (IQ), and reflected power). In addition a slower multifunction card, a NI PCIe-6323, X series card with 16 AI, 250 kS/s, 16 bit; 4 DAC, 900 kS/s), 16 bit; 48 DIO, is used to detect other, slower varying signals, like master oscillator power.

Software Description

The RF software has three layers - FPGA, real-time and host program, see Figure 2. In the FPGA of the fast DAQ card amplitude and phase signals are calculated from the I and Q signals from the IQ demodulators. The cavity amplitude signal is, via a PID loop on the FPGA used to regulate the drive power through a voltage controlled attenuator. The bandwidth of this loop is ~50 kHz.

All signals (both I and Q and amplitude and phase) are via DMA sent to the real-time program. In the real-time program the phase difference of the cavity signal and the forward power signal is used to control the tuning of the cavity. The regulation here is a simple limit crossing algorithm. The advantage of a limit crossing algorithm is that the tuning motors are actuated less often, reducing wear.

The real-time program has all the controls and indicators, including plots, to operate the system. However since the system will be head-less (no screen) in normal operation, almost all control and indicators are also available through Shared Variables and data streams for fast plot data. A LabVIEW host program access these Shared Variables and data streams allowing full remote operation of the system including plots to monitor RF (baseband) and regulation signals. ConSys is also, through the general Shared Variable device, able to access all simple control and status.

- K.T. Nielsen et.al: "ConSys A CONTROL SYSTEM WITH A HIGH DEGREE OF ABSTRACTION", Proc. of PCaPAC'99, KEK, Tsukuba, Japan, poster P24.
- [2] J.S.Nielsen, K.T.Nielsen and T.Worm: "ConSys A new control system for ASTRID and ELISA", TUP46G, EPAC98.
- [3] www.isa.au.dk/consys
- [4] S.P. Møller, N. Hertel, J.S. Nielsen: "Status and very first commissioning of the ASTRID2 synchrotron light source", TUPPP003, IPAC2012.

PLC-BASED CONTROL SYSTEM FOR LINEAR ACCELERATOR (LCS) AT **EBC KHARGHAR, BARC**

A. S.Chachondia, B.B.Biswas^{*}, G.Ganesh, M.B Patil, R.K.Patil^{*}, M. Kumar, D.P. Chakravarthy^{*}, K.C. Mittal, and L.M. Gantayet, BARC, Mumbai, India

Abstract

Currently the 10MeV Linac is being used for different research and industrial applications. The control system in operation was developed using CAMAC based Data Acquisition System (DAS) and Hard-wired Interlock System. It is proposed to replace the CAMAC system with a state-of-the-art indigenously developed Programmable Logic Controller (PLC) that is verified to the level of a Class IB computer-based system used in nuclear power plants. A PLC node comprises of two VME bus based CPU boards (PowerPC MPC7447, 600MHz) working in redundant mode. The inputs and outputs are common to both CPUs. The I/O boards are hot swappable and intelligent. An intelligent Ethernet board is used for communication with a PC running the SCADA software and industry standard communication protocols drivers. The PLC hardware and software has undergone rigorous verification and validation. A user-friendly 'Application Development Environment' is provided to the process engineer for building the application using pre-defined function blocks. The LCS developed using PLC is to be used for operating the Linac irradiation facility, remotely as well as locally in a fail-safe mode, with sequential start-up and sequential shut-down. Apart from system status monitoring, data archiving, alarm generation and setpoint adjustments, it monitors the important parameters and trips the Gun Modulator High Voltage (GM HV), the Klystron Modulator High Voltage (KM HV) and the Electron Gun Power Supply (EG PS) on fault conditions.

INTRODUCTION TO LINAC^[1]

The 10 MeV Linac Irradiation Facility consists of the RF Linac and conveyor system. It is housed in a twostoried building with 2.6 meters thick concrete shielding walls. The Linac structure with electron gun, RF cavity, gate valves, Sputter Ion Pumps (SIP), and beam transport line is at the first floor as shown in figure 1, whereas scan magnet, scan horn, titanium foil, SIP for scan horn and conveyor system for material handling is at ground floor. The Electron beam at 50 KeV is generated in electron gun

*retired

with LaB₆ cathode and injected into the on-axis coupled cavity Linac which accelerates the electrons to 10 MeV. A 2856 MHz, 6MW Klystron based RF Power source is used to establish the required electric field of 18 MV/m inside the Linac. After acceleration, the magnetic sweep scanner deflects the beam in the scan horn and is taken out to the atmosphere through a 100 cm X 7 cm X 50 µm titanium foil exit window for irradiation jobs. The Linac up to titanium foil is maintained at 10^{-7} torr vacuum with the help of rotary-backed turbo molecular pump and sputter-ion pumps.

Linac can be operated in various modes like RF Conditioning Mode, Vacuum System Maintenance Mode, Beam Trial Mode and Job Irradiation Mode using the LCS.



Figure 1: 10- MeV LINAC.

SYSTEM REQUIREMENTS

The main function of the system is to trip the GM HV, the KM HV and the EG PS against unhealthy conditions of various digital and analog inputs. The system is also used by the operator for operating the subsystems of Linac during various modes of operation. Fail-safe operation, with sequential start-up, sequential shut-down and tripping of Linac is the main requirement of LCS in all modes of operation. The system checks a programmed (but modifiable) set of interlocks at every stage of the startup and shutdown sequence.

LCS is capable of processing approximately 150 analog inputs, 200 digital inputs and generating 108 digital outputs. LCS also receives commands from operator to operate the pumps, cooling compressors, cooling fans, power supplies etc and generates a 200 msec pulse output signal to operate the equipment. Whenever any command is received from the operator, the LCS operates the corresponding sub-system and subsequently checks the status by scanning the corresponding digital input signal and communicates the status back to the operator. Whenever a command to operate the equipment is unsuccessful, an error message is generated and sent to the operator.

SYSTEM DESCRIPTION

The configuration of the system is given below:



Figure 2: System Configuration.

The LCS is being developed using PLC as DAS and COTS SCADA package for data storage and data display in different formats. The in-house developed PLC has PowerPC based CPU cards and real- time operating system (RTOS) for managing the CPU resources.

Personal Computers are used for Operator Interface. The system has one Operator Console named as "LCS – OC" at the control room G-14A at ground floor. From this OC the Linac operations are controlled. To view the Linac sub-system data, additional Display Stations (DS) are provided at 122A Power supply room, Scan Horn area, Linac shell area, and Project Manager Room. No control operations are possible from these additional Display Stations.

The embedded software running on PLC samples all the digital and analog signals every 100 ms and communicates with SCADA software running on OC and DSs on a 100Mbps LAN. PLC sends the current values of all signals to the OC on the Ethernet link periodically. PLC also sends the alarms and trips generated along with time stamp to the OC as and when they are detected. The OC saves the important signals at regular configurable intervals and sends the user commands to PLC.

The LCS works as a stand-alone system. The safety interlocks are built at the software level and backed by an independent stand-alone hard-wired interlock system.

The LCS-OC communicates with Pulse Generation and Measurement System (PGMS) comprising of oscilloscope, function generator, RF driver. Pulse signals of 10 μ s pulse width like klystron current, reflected RF power, forward RF power, Beam Energy, Beam Current etc. are monitored using an oscilloscope and sent to OC for storage and display.

The PLC is programmed using the Engineering console (EC). Both system configuration and application configuration need to be carried out before the PLC can be deployed as a DAS. System configuration along with application configuration are compiled together and downloaded to the PLC. Changes to the configuration can be made by authorized personnel.

On every start up, the system carries out self-diagnostics immediately before it starts scanning the inputs. Further diagnostics is carried out periodically at user-defined interval. If any of the cards of the system is detected faulty, the selfdiagnostics is repeated and only on confirmation of the fault, it is annunciated. The system also checks for the Real Time Clock (RTC) on the CPU boards on start up.

The PLC software executes on the node configuration data. During initialization, PLC software reads system configuration data and configures itself. It reads application configuration data and interprets it for cyclic execution of the configured application.

MODES OF OPERATION

LCS is used in all the modes of Linac operations to check and ensure sequential safety interlocks at start-up, trip interlocks while in operation and sequential shut-down interlocks at the time of shutting down. LCS trips the Linac in case any trip related parameter crosses its limit. The various Linac modes of operation and the role played by LCS are described below briefly.

RF Conditioning Mode

In this mode Beam is not energized (GM and EG PS is not $\stackrel{2}{\leftarrow}$ energized). RF Power is fed to cavity by energizing the Sklystron and Klystron Modulator. Conveyor system, Ozone

Removal System, and Foil Blower Cooling System are not energized. LCS checks the vacuum level, water flow, human safety interlock before feeding the RF power to the cavity. LCS does not allow operating other subsystems in this mode.

Vacuum System Maintenance Mode

Linac has to operate in this mode when the vacuum of the order of 10^{-7} torr has to be generated from atmospheric pressure. In this mode LCS ensures that only vacuum related systems can be turned ON. It ensures that SIPs can not be turned ON till rotary backed Turbo Molecular Pump (TMP) is operated to achieve the vacuum of the order of 10^{-5} torr in Linac. After achieving vacuum of 10⁻⁵ torr, LCS allows operator to turn on SIPs to achieve and sustain the vacuum of the order of 10^{-7} torr for beam generation and operation.

Beam Trial Mode

Linac has to operate in this mode before declaring the facility ready for the user's application. In this mode, Linac is energized for normal beam operation. LCS carries out start-up and shutdown as per the accepted sequence. Parameters like Beam Energy, Beam Current, Beam Power, conveyor speed and radiation dose for the irradiation job are validated using LCS.

Job Irradiation Mode

Linac enters into this mode of operation after successfully completing the start-up interlocks. In this mode of operation Linac is energized as per the job's requirement. After successfully completing the irradiation job, LCS checks shutdown interlocks for smooth shutdown of Linac. For ex. LCS ensures that Mobile Shield door and Room 121 door remain closed till ozone level falls below 100 ppb.

HARDWARE DETAILS [2]

The PLC hardware is designed around in-house developed modular electronic hardware boards that use proven and widely used components. VME32 (A24, D16) and a proprietary I/O bus are used as the two system buses.

The boards in the system are arranged in two 19" bins (CPU bin and I/O bin). The CPU bin contains two 7-slot VME backplanes. Each VME backplane accommodates one CPU board, one Ethernet board with dual media (ENET-DM) and one Protocol translator board (PTB).

The CPU board is based on PowerPC 7447A microprocessor running at 600 MHz. The PTB provides a bridge between VME bus in the CPU bin and I/O bus in the I/O bin. The ENET-DM board provides 100-Mbps Ethernet connectivity that is used to establish communication interface with OC.

I/O bin accommodates all the required sixteen I/O boards and two IO Bus eXtender (IOX) boards. The I/O boards are accessed by both the CPUs in the CPU bin. The I/O board access is facilitated by PTB and IOX board connection through flat cable.

The LCS uses five Analog Input Output Boards, seven Digital Input Boards and four Digital Output Boards.

OPC SERVER FOR PDCP

For data transfer between the LCS-PLC and LCS-OC, Packed Data Communication Protocol (PDCP) is used at the application layer. TCP/IP is the base protocol. An OLE for Process Control (OPC) Server for PDCP is used for data communication. PDCP is a rugged protocol capable of handling data of various sizes like bit, word, float and double. It can also handle alarms, messages and diagnostics data.

GRAPHICAL USER INTERFACE (GUI)

EC provides the GUI required for creating the node configuration. The system configuration details like no of input / outputs, no of cards, type of signals, their names and other details are configured using the 'Application Development Environment' software package on EC. The application is configured using pre-defined function blocks as the building blocks.

The SCADA package on DSs is configured to display the current data and events/alarms in a tabular format. Last event/alarm is added at the bottom of the list. Historical data is also available for viewing. The operator can print selected data and transfer the desired data to CD/DVD on demand.

CONCLUSION

Since the control system is designed using an in-house developed PLC, long term maintenance, support and upgradation will be possible. The control logics of the system are built in the form of function block diagrams and are easily modifiable. The process engineers can carry out changes in the control logics without consulting the PLC developer. State-of-the-art hardware has gone into building the system and so obsolescence of component will set in late in life of the system.

REFERENCES

[1] K.C.Mittal, "Status of Industrial Electron Linac Development Program at BARC, India" InPAC-2011, at IUAC New Delhi, Feb, 15-18 2011.

M.K.Sharma etal.,"Hardware Standardization [2] for Embedded Systems" NSNI 2010, BARC, Feb24-26 2010.

STATUS OF THE ULTRA FAST TOMOGRAPHY EXPERIMENTS **CONTROL AT ANKA**

D. Haas, W. Mexner, T. Spangenberg, A. Cecilia, P. Vagovic KIT, ANKA, Karlsruhe, Germany A. Kopmann, M. Balzer, M. Vogelgesang, H. Pasic, S. Chilingaryan KIT, Institute for Data Processing and Electronic, Karlsruhe, Germany

Abstract

X-ray imaging permits to spatially resolve the 2D and 3D structure in materials and organisms, which is crucial for the understanding of their properties. Additional temporal resolution of structure evolution gives access to dynamics of processes and allows to understand functionality of devices and organisms with the goal to optimize technological processes. Such time-resolved dynamic analysis of micro-sized structures is now possible by aid of ultrafast tomography, as being developed at the TopoTomo beamline of the synchrotron light source ANKA.

At TopoTomo, the whole experimental workflow has been significantly improved in order to decrease the total duration of a tomography experiment down to the range of minutes. To meet these requirements, detectors and the computing infrastructure have been optimized, comprising a Tango-based control system for ultra fast tomography with a data throughput of several 100 MB/s. Multi-GPU based computing allows for high speed data processing by using a special reconstruction scheme. Furthermore the data management infrastructure will allow for a life cycle management of data sets accumulating several TByte/day.

The final concept will also be part of the IMAGE beamline, which is going to be installed in 2013.

INTRODUCTION

Third-generation light sources offer new possibilities for X-Ray imaging experiments. Fast imaging techniques became available, thanks to the high flux and to new fast CMOS detectors. One of the fields benefiting from is micro-tomography, where the speed-up is generating large data sets in a few seconds.

The reconstruction of these large 3D data sets was usually performed offline. Due to the reduced reconstruction time from initially several hours down to only a couple of seconds the reconstruction became part of the online data processing chain. For the TopoTomo beamline at ANKA [1, 2] a GPU-based ultrafast tomography framework (UFO) has been developed in order to merge all components of a tomographic scan into a homogenous workflow.

Remaining challenges concern the need for a high performant, but still reliable control system, embedded experimental workflow, a quick online rendering and visualization. The final evaluation is still done by commercial software. Last, but not least, for handling experimental data of up to several TByte/day, a data-andlife-cycle management has to be introduced.

CONTROL SYSTEM OF TOMOGRAPHY

At ANKA the different layers (hardware, control, user interface) of a beamline communicate via the Tango software bus [3] (see Fig. 1). All hardware components are realized as Tango servers. Also the SCADA System WinCC-OA (PVSS2) [4] for vacuum control, etc., can been regarded as a Tango server. The Tango client SPEC [5] and its easy-to-handle macro language is used as command-line scripting interface to adapt the beamline optics and auxiliary elements to the specific experimental needs.



Figure 1: Communication (blue arrows) and dataflow (red arrows) between the different software layers and devices.

Instead of the slow overall-beamline control system, an optimization to reduce the total duration of a tomography experiment requires a significant improvement of the performance of all layers and of the IT infrastructure. For the software layer, the dead time (~ms) of a pure Tangobased software communication has to be replaced by hardware based triggering and the streaming of the data has to be designed carefully.

For the infrastructure of the TopoTomo beamline four servers (see Table 1) are involved in the workflow of a tomography scan. The different servers are connected to

each other by a 10 Gbit fiber optics network, which reaches a data throughput of 700 MByte/s.

Currently, SPEC, running on the control machine, represents the front end to the user and manages the experimental workflow. Depending on the driver implementation, the Tango device servers for the detector are handled via a windows server or are connected directly to the Linux processing machine for better performance. Both paths are transparent to the UFO-framework via the libuca-library [6].

Table	1:	Servers	at	Topo	Tome
-------	----	---------	----	------	------

Server	CPU	GPU	RAM	Hard disk
Control machine (user interface)	Xeon E5520	Nvidia Quadro NVS 450	6 GB	500 GB
Detector server (windows)	Xeon X5650	none	72 GB	20 TB
Detector/Proces- sing server (linux)	2xXeon E5620	2xGeforce GTX 580	144 GB	20 TB
Visualisation machine	4xXeon W5590	Nvidia Quadro FX 5800	148 GB	3,7 Tb
		Nvidia Tesla C1060		

The high-speed tomography scan is initiated by SPEC via a software trigger to the detector, starting a continuous frame acquisition. Each frame of the detector generates a hardware trigger to the free-running rotation stage (*Aerotech*), which is precisely recording the angular position for every pulse. By avoiding a frame trigger for the command level, the total duration time of the scan is significantly decreased.

After the image acquisition during the scan, the image data of the detector, having a size of easily up to 160 GByte, together with the position data of the rotation stage are buffered on a local Raid 6 storage (16 drives). The further processing is done by the UFO framework (Ultra-fast X-ray Computer Tomography), which is described in detail in the next section. The data set can be reloaded and preprocessed via the framework. The volume reconstruction is done in minutes by a GPU based back-projection algorithm on the reconstruction machine.

The visualization of the reconstructed volume is done by VG Studio Max [7] on a dedicated visualization machine. Finally the data is archived at the Large Scale Data Facility [8], a storage supported by KIT.

UFO-FRAMEWORK IN DETAIL

The highly optimized image processing algorithms are managed in a novel parallel computing framework. The framework ensures an optimal performance of the imageprocessing pipelines, while it provides a modular and extensible platform for shared development of these algorithms. The framework is intended for continuous processing of streamed data and covers the full signal chain with interface to the detectors, pre-processing, reconstruction algorithms and fast storage of the data.

The framework is implemented in C and uses the GLib standard library for basic data structures and algorithms. The core components follow the GObject object model to provide language bindings for arbitrary languages (such as Python or Javascript) [9]. The OpenCL API is used to exploit the parallel processing capabilities of recent GPUs for data processing, using a variety of hardware [10]. Although OpenCL is portable, users can provide optimized kernels for their algorithms depending on which hardware the kernel is executed on. Based on this core, developers implement filter nodes that either provide, process or store data. The processing may be executed using the GPU's, CPU's or both types of computational resources simultaneously. Data transfer from and to different devices are handled transparently, therefore users do not have to care explicitly about memory management. The nodes are then connected in a graph structure to describe the actual flow of computation as a composition of smaller sub-operations. At run-time, each filter node is then mapped to a computation thread and executed.

The implemented modules include input/output nodes, several pre-processing routines and the reconstruction code. Access to a variety of cameras is provided via the Unified Camera Abstraction library (libuca). It allows developers to use different cameras with a single software interface. Furthermore, a Python extension module is provided to allow developers to use Numpy/Scipy packages for processing the data.

FIRST TOMOGRAPHY WORKFLOW TESTS

The efficiency of the different hardware and software layers within the fast tomography workflow was tested by a white-beam experiment at the TopoTomo beamline at ANKA.

Experimental Setup

For the tomography scan a CMOS high speed detector (*Photron SA-1*) was used, which has a maximum frequency of 5400 frames/s at full-frame mode. The optics consists of a 74-µm-thick LSO:Tb scintillator and a ED eyepiece (f=180mm, numerical aperture (NA) 2.8 *Nikon Nikkor*) combined with an objective (f=50mm, max. NA 0.43, *Rodenstock TV- Heliflex*). Hence it results in a total magnification of 3.6x, which yields an effective pixel size of 5.5µm.

The image acquisition of the CMOS detector in combination with the rotation stage was synchronized during the tomography scan by hardware triggered pulses. The speed of the rotation stage was 450°/s by taking 2000 frames/s with the CMOS detector. Consequently, the total duration of the tomography scan with 800 projections takes 0.4s [11].

Results

The tomography scan, described in the chapter before, was reconstructed once with PyHST [12] and secondly with the UFO framework. The overall reconstruction time of both algorithms is shown in Table 2.

Algorithm	Overall reconstruction time
PyHST	31.3 s
UFO- Framework	12.5 s

The reconstructed volume (see Fig. 2) was rendered in VG Studio Max on the visualization machine.



Figure 2: Volume rendering of an egg of the stick insect *Peruphasma schultei*.

OUTLOOK

The current challenge is to organize and automatize the full experimental workflow, including the data management. Caused by the HDRI project [13], the main goals for the near future are to store the data, including metadata, in the Nexus [14] format and to implement a life-cycle management. The life cycle starts with the proposal of the user and ends with the archiving of the acquired data for a predefined period.

For a full automatization of the experiment partial manual operations, like sample exchange via a robot, succeeding rotation axes alignment, and autofocussing still needs to be fully automatized, so that the whole experiment can be operated by an inexperienced user via a graphical user interface.

All these tasks result in a considerable decrease of the overall time of a tomographic scan and make it more comfortable. The final concept will be implemented and further developed at the upcoming IMAGE beamline at ANKA.

ACKNOWLEDGEMENTS

This work has been supported by the Helmholtz Portfolio Extension "Large Scale Data Management and Analysis" with contributions of the Data Life Cycle Lab "Key Technologies" and the "Data Services Integration Team". Beamtime at ANKA is acknowledged.

- [1] A. Rack, T. Weitkamp, S. Bauer Trabelsi, P. Modregger, A. Cecilia, T. dos Santos Rolo, T. Rack, D. Haas, R. Simon, R. Heldele, et al., "The microimaging station of the topotomo beamline at the ANKA synchrotron light source", *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 267 (2011): p. 1978 - 2009.
- [2] P. Vagovic, T. Farago, T. dos Santos Rolo, S. Chilingaryan, T. Müller, A. Cecilia, T. van de Kamp, E. Hamann, A. Riedel, D. Haas, W. Mexner, M. Fiederle, T. Baumbach, "Recent upgrades at the micro-tomography station at TopoTomo Beamline at ANKA light source", *will be published in Journal of Synchrotron Radiation*.
- [3] J.-M. Chaize, A. Goetz, W.-D. Klotz, J. Meyer, M. Perez, E. Taurel, "TANGO – an object oriented control system based on corba", *International Conference on Accelerator and Large Experimental Physics Control Systems*, (1999): p. 475 – 479.
- [4] PVSS2, http://www.pvss.com/.
- [5] SPEC, http://www.certif.com/.
- [6] M. Vogelgesang, S. Chinlingaryan, T. dos Santos Rolo, A. Kopmann, "UFO: A Scalable GPU-based Image Processing Framework for On-line Monitoring", *The Third International Workshop on Frontier of GPU Computing, Liverpool, UK, 25 - 27 June, 2012*, FGC 2012.
- [7] VG Studio Max, http://www.volumegraphics.com/.
- [8] R. Stotzka, W. Mexner, T. Dos Santos Rolo, H. Pasic, J. van Wezel, V. Hartmann, T. Jejkel, A. Garcia, D. Haas, A. Streit, "Large Scale Data Facility for Data Intensive Beamlines", *Proceedings of the 13th International Conference on Accelerator and Large Experimental Physics Control Systems*, (2011): p. 1216 1219.
- [9] GNOME library, http://developer.gnome.org/glib/
- [10] OpenCL The open standard for parallel programming of heterogeneous systems, http://www.khronos.org/opencl/.
- [11] A. Cecilia, A. Rack, P.-A. Doussard, T. Martin, T. dos Santos Rolo, P. Vagovic, E. Hamann, T. van de Kamp, A. Riedel, M. Fiederle, T. Baumbach, "LPE grown LSO:Tb scintillator films for high-resolution X-ray imaging applications at synchrotron light sources", *Nuclear Instruments and Methods in Physics Research A*, 648(2011): p. 321 - 323.
- [12] S. Chilingaryan, A. Mirone, A. Hammersley, C. Ferrero, L. Helfen, A. Kopmann, T. Dos Santos Rolo, P. Vagovic, "A GPU-Based Architecture for Real-Time Data Assessment at Synchrotron Experiments", *Nuclear Science, IEEE Transactions on*, 58 (2011): p. 1447 1455.
- [13] High Data Rate Initiative, http://www.pni-hdri.de/.
- [14] Nexus, http://www.nexusformat.org/.

HYPERARCHIVER: AN EVOLUTION OF EPICS CHANNEL ARCHIVER

M. del Campo*, I. Arredondo, ESS-Bilbao, Zamudio, Spain J. Jugo, University of the Basque Country, Leioa, Spain M. Giacchini, L. Giovannini, INFN/LNL, Legnaro, Italy

Abstract

Data storage is a primary issue in any research facility. In the EPICS middleware based accelerator community, Channel Archiver has been always considered the main reference. It works with Oracle and MySQL, probably the best well known relational databases. However, demanding requirements at minimum costs have fostered the development of a wide range of alternatives, like MDSPlus (Consorzio RFX), SciDB (BNL) or Hypertable (IFNF). This document launches a tool called HyperArchiver, which was firstly developed at IFNF (Italy) and eventually customised by ESS Bilbao (Spain). Based on a NoSQL database named Hypertable, it focuses on large data sets management with maximum scalability, reliability and performance. Besides the update and further customization made at ESS Bilbao, HyperArchiver is presented with a set of GUIs, in order to provide an easy use and integration with any general control system. A LabVIEW VI and two cross-platform PyQt GUIs for both Hypertable data retrieval and HyperArchiver control have been developed and successfully tested at ESS Bilbao (see Figures 1-3).

INTRODUCTION

Particle accelerators are very complex and expensive devices. Therefore a reliable distributed control system allowing easy maintenance and upgrading turns basic in this kind of facilities. EPICS (Experimental Physics and Industrial Control System) is a set of Open Source software tools and applications, which provides the infrastructure for distributed control systems. Many particle accelerator, large experiments and major telescopes facilities use it to build complex control systems made of tens or even hundred of computers, networked together to allow communication between them and provide control and feedback from different locations. In this context, data storage is a very important issue, not only as part of the control system itself, but also for the proper functioning and use of the accelerator and its experimental lines.

EPICS ARCHIVING TOOLS

The most standard EPICS standalone client for data archiving is called Channel Archiver [1], which is based on the relational databases MySQL and Oracle (therefore it is also called RDB Archiver). Unfortunately, they both have some well known drawbacks, which explain the efforts made towards the improvement of its performance and usability [2]. However, this has also fostered the development of a wide range of alternatives focused on high performance and large-scale dataset management, like MDSPlus,

SciDB (BNL) [3] or Hypertable (IFNF) [4]. In this work only the last one will be discussed, but it should be kept in mind that they are all possible solutions for the RDB Archiver limitations.

Channel Archive

The Channel Archiver is the standard archiving tool for EPICS. Using the EPICS Channel Access (CA) network protocol, it can collect real-time data from any CA server on the network. It stores the full data set offered by CA (values, timestamps, status information, engineering unit names and display, control and alarm limits), allowing scanning at a fixed period or on change.

Since its original design, the EPICS Channel Archiver has undertaken several significant transformations. The version publicly known as an EPICS extension [1] was developed at the Spallation Neutron Source (SNS), Los Alamos, USA. Its main feature is the usage of a relational database instead of the indexed files used by the original Channel Archiver [5]. This version of Channel Archiver supports both MySQL and Oracle RDB and is entirely written in Java as part of the Control System Studio (CSS). This tool significantly improves data access and retrieval, in comparison to the original indexed file. Nevertheless, both Oracle and MySQL have some disadvantages like the expensive costs of Oracle or the effective maximum table size in MySQL, which can become a great limitation considering that RDB Channel Archiver stores all samples in one single sample table.

HyperArchiver

HyperArchiver [6] is a modification of the Java RDB ArchiveEngine which uses Hypertable instead of MySQL or Oracle. It was developed at INFN/LNL in Legnaro (Italy) and customised at ESS Bilbao (ESS-B) afterwards. Hypertable is a non relational, high performance and distributed database released under GNU license, which focuses on management of large data sets with maximum scalability and reliability. HyperArchiver is actually a hybrid version, as samples (bulk data) are stored in Hypertable database, but the basic static data of EPICS PVs is still recorded in RDB Archiver's MySQL database. First tests at Brookhaven National Laboratory (BNL), USA, have already shown very good performance on data insertion and retrieval [7].

HYPERARCHIVER AT ESS BILBAO

HyperArchiver Customization

HyperArchiver version released at Legnaro was meant to be used with Hypertable 0.9.3.3, and had some proved

^{*} mcampo@essbibao.org

limitations on arrays management. For these reasons, at ESS-B some further modifications where required before it could be properly used. The features added at ESS-B to the original HyperArchiver are the following:

- Upgrade from 0.9.3.3 to 0.9.4.3 and further Hypertable versions (a new architecture based on namespaces was introduced from version 0.9.4.3). Last setups at ESS-B are currently working with version 0.9.5.6.
- Improve the management of large arrays and the way they are stored inside Hypertable. This was something that had not been tested during HyperArchiver development at INFN/LNL, mainly focused on single PVs.
- Provide the user the possibility to automatically create the EngineConfigImport XML configuration file directly from an EPICS database file. This goes along with a general effort at ESS-B to standarize all configuration files, providing conversion tools.

Integration into the General Control System

The standard use of HyperArchiver remains the same as for the classical RDB Channel Archiver. They both can be easily controlled from any command-line interface terminal. Nevertheless in order to simplify its use, especially for operators who might not be used to the UNIX terminal interface, an additional effort was made to develop some graphical user interfaces (GUI).

PyQt: PyQt4 is a set of python bindings for the Qt 4 cross-platform GUI/XML/SQL C++ framework. A standalone PyQt4 GUI was written in python at ESS-B and and has been established as the internal standard interface for HyperArchiver clients. It allows not only the initialization and termination of the HyperArchiver services but also the customisation of all its configuration parameters with no need to use the command line terminal or any external text editor tool. This avoids the user a lot of tedious work.



Figure 1: PyQt GUI for HyperArchiver Control.

LabVIEW: HyperArchiver is just one of many tools used in a control system. Therefore its integration into the general control system should be as easy and quick as possible. This was the purpose of a set of LabVIEW virtual instruments (VI) designed to control the Archive Engine with the same functionalities as the PyQt interface. This makes extremely easy the integration of a data storage engine in any of the LabVIEW control programs which are used at ESS-B at the moment, by simply including the mentioned VIs (programs/subroutines) inside the general block diagram.



Figure 2: LabVIEW VI for HyperArchiver control.

Data Visualization

Regarding data visualization, the INFN/LNL version of Hypertable included a further modification of CSS code for the retrieval of historical data from Hypertable instead of MySQL or Oracle. It pictures data by means of the CSS data browser. CSS is a very powerful tool which provides a lot of functionalities. However, at this moment at ESS-B, less complex and more specific tools were requested. In this way, another python GUI was developed in order to have at disposal a simple data retrieval tool, lighter than CSS. It uses the Hypertable Thrift Client Interface to access the database and after thorough tests at ESS-B [8] it proved to be even faster than CSS Data Browser. Moreover, the use of python made it possible to integrate the data retrieval tool into the general HyperArchiver PyQt GUI, making available all the functionalities required for the control and use of HyperArchiver from a single multiplatform user friendly interface.

RESULTS

This work presents the EPICS experience at ESS-B regarding data storage and database management. To date, two different versions of EPICS Archive Engine have been used and tested to study their performance and suitability to ESS-B requirements: RDB Archiver and HyperArchiver.

RDB Archiver: RDB Archiver was set up with a MySQL server. Both the Archive Engine and the relational database worked without any issue during laboratory tests. However, first real tests at ITUR (the ESS-Bilbao's frontend test stand for ion sources) revealed a potencial problem regarding size limitations of the tables. ITUR's control system is characterized by the use of several array PVs, sam-

		HyperTable GUI		
	NAME	DESCRIPTION		
1 🗆 10	C:ISHN:LR	Local Remote mode switch		
IOC:ISHN:Presost		Status of general presostate		
3 🗆 10	C:ISHN:Run	RT Program On/Off		
4 🗆 10	C:ISHN:BD:ACCT:BeamCurr	ACCT Beam Current. Array format.		
5 🗹 10	C:ISHN:BD:DCCT:BeamCurr	DCCT Beam Current. Array Format.		
5 🗆 10	C:ISHN:BD:FAR:BeamCurr	Faraday Cup Beam Current. Array format.		
7 🗹 10	C:ISHN:DB:OnOff			
8 🗆 10	C:ISHN:HCPS:DC:Vwarm			
9 🗆 10	C:ISHN:HCPS:PULS:Duty	Plasma PS Duty		
10 🗆 10	C:ISHN:HCPS:PULS:MB:Curr	Plasma PS MB Curr		
11 🗆 10	C:ISHN:HCPS:PULS:Ref	Plasma PS Current		
	iii		>	
itart Date	Time: 18-09-2012 14:17:17	 ✓ End DateTime: 18-09-2012 15:17:17 ✓ ✓ Graph Plot 		
Eacl	h variable in one file	Each plot in one window All plots in one window		
 All variables in one file 		All variables in one plot		
	Show Data	All variables in one plot		

Figure 3: PyQt GUI for Hypertable data retrieval.

pled at a high sampling rate, for waveform storage. Considering that RDB Archiver manages arrays creating an entry into a single MySQL for every item of each array, size limitation turned out to be a problem. The maximum table size for MySQL databases is usually determined by operating system (OS) constraints on file sizes, not MySQL, but can be so restrictive as 4GB on a 32-bit architecture MySQL. Table 1 lists some rough aproximations on different OS [9]. The use of a proprietary distributed cluster of Oracle's databases was dismissed at first at ESS-B, which leans towards the promotion of the Open Source model. The better suitability of a distributed large-scale dataset oriented database thence emerged.

Table 1	:	File	Size	Limits	on Different	OS
			~			~~

Operating System	File size Limit
Win32 w/ FAT/FAT32	2 GB / 4 GB)
Win32 w/ NTFS	2 TB
Linux 2.2-Intel 32-bit	2 GB (LFS: 4 GB)
Linux 2.4+	4 TB (using ext3)
Solaris 9/10	16 TB
MacOS X w/ HFS+	2 TB

HyperArchiver: HyperArchiver is actually a modification of RDB Archiver which stores data into the NoSQL database Hypertable instead of the RDB MySQL or Oracle. Hypertable is designed to manage the storage and processing of data on a large cluster of commodity hardware, providing resilience to machine and component failures. At ESS-B it has proved to be a reliable and scalable alternative to MySQL. Furthermore, other accelerator facilities have shown their interest on HyperArchiver as an alternative to the traditional RDB Archiver, like Diamond (UK).

Briefly, HyperArchiver emerged as an evolution of the standard RDB Archiver, modified to work with Hypertable as the main underlying database. The current trend towards NoSQL databases seems natural in the field of particle accelerators. Nowadays this kind of facilities generate huge amounts of data, which have to be immediately processed and properly recorded. NoSQL databases have been especifically developed to manage large volumes of data that do not necessarily follow a fixed schema. Moreover, they employ distributed architectures, which allows scalability and tolerance to hardware failure, both issues of great importance in any research facility.

NEXT STEPS

This aim of this work is to emphasize the feasibility of an evolution in the data archiving field of EPICS control systems. This is made through the example of HyperArchiver, a customization of the classical EPICS RDB Archiver which uses Hypertable instead of MySQL or Oracle. However, the huge amount of NoSQL databases focused on large datasets like Hypertable, creates a wide range of similar possibilities based on different databases, or even on different storage methods. The most important ones have already been mentioned, but with no further references, because they have not been tested yet at ESS-B. Therefore, even if HyperArchiver has proved to be a good and reliable archiving client, the logical path to follow in the future would be to test and compare other methods available, in order to determine the advantages and disadvantages of each one, trying to figure out which one is more suitable for each scenario. It is worth mentioning that this job has already been started at BNL, where HyperArchiver was actually conceived. They have created a common test bench to evaluate the various archiver developments [10].

- [1] http://ics-web.sns.ornl.gov/kasemir/archiver/
- [2] J. Rowland, M.T. Heron, S.J. Singleton, K. Vijayan, M. Leech, "Algorithms and Data Structures for the EPICS", ICALEPCS 2011, Grenoble, France.
- [3] N. Malitsky, D. Dohan, "A Prototype of the Next EPICS Archiver Based on the SciDB Approach", ICALEPCS 2011, Grenoble, France.
- [4] http://www.lnl.infn.it/~epics/joomla/ archiver.html
- [5] http://arxiv.org/abs/cs/0110066v1
- [6] M. Giacchini, L. Giovannini, M. Montis, G. Bassato, J.A. Vasquez, G. Prete, A. Andrighetto, R. Petkus, R. Lange, K. Kasemir, M. del Campo, J. Jugo, "HyperArchiver: An EPICS Archiver Prototype Based on Hypertable", ICALEPCS 2011, Grenoble, France.
- [7] http://www.lnl.infn.it/~epics/ HyperArchiverAix2010.pdf
- [8] M. del Campo, J. Jugo, M. Giacchini, L. Giovannini, "EPICS HyperArchiver: Initial Tests at ESS Bilbao", IPAC2011, San Sebastiàn, Spain.
- [9] http://dev.mysql.com/doc/refman/5.1/en/ table-size-limit.html
- [10] http://sourceforge.net/projects/ epics-archbench

EPICS MySOL ARCHIVER INTEGRATION BETWEEN EPICS AND MySOL

A. Roy*, R.B. Bhole, S. Pal, D. Sarkar, VECC, Kolkata, India

Abstract

The performance evaluation and analysis of intersystem dependency of the various subsystems of the Superconducting Cyclotron (SCC) demand a well configured data logging, archiving and historic analysis facility for large number of control parameters along with on-line failure analysis facility of every system. Experimental Physics & Industrial Control System (EPICS) is used as development architecture of the control system of these systems with MySQL as database for large amount of relational data management. This combination requires integration between EPICS and MySQL server. For this purpose, MySQLArchiver as an EPICS Extension is developed for data logging and archiving of control parameters into MySQL database. This extension also provides a web based tool for online monitoring of control parameters and historic analysis of archived data. This paper describes the software architecture, implementation, as well as method of configuration for any other EPICS based control system as a utility. This facility is also elaborated with examples, web page views and experiences of deploying it in SCC.

INTRODUCTION

The performance of a particle accelerator is measured by its beam stability. The beam stability is achieved through fine tuning of the system parameters. The requirement of archiving the system parameters originates here. The Experimental Physics & Industrial Control System (EPICS), a standard open-source dual layer software tool for designing distributed control system, is widely used in large accelerators around the world for implementing supervisory control software. It has its own Channel Access (CA) archiver for storing system parameters into a proprietary database using channel access protocol. Since modern accelerators e.g. Superconducting cyclotron are comprised of various critical systems e.g. cryogenic system which are entirely vendor dependent. These systems have their own proprietary or third party supervisory system mostly without support for EPICS channel access. While fine tuning such accelerators with EPICS at supervisory control layer, EPICS channel archiver fails to store the third party system parameters. Again configuration and maintenance of the archiver demand expertise in EPICS tool set.

In contrast, interface to an open relational database e.g. MySQL, PostgreSQL etc. is more readily supported by most of the industry standard control software. The availability of expertise and support for these databases is plenty in industry. Hence an archiver based on such open relational database engine with channel access support will mitigate the requirement of storing control parameters from heterogeneous systems (EPICS and non-EPICS/third party) in a common platform. An EPICS archiver is developed using MySOL as database engine for archiving and retrieving the control parameters of Superconducting Cyclotron (SCC), Kolkata, MySOL is chosen due to its simplicity for installation and maintenance.

SOFTWARE OVERVIEW

An archiver is a client tool, i.e. CA Client, from the perception of EPICS dual layer architecture. The main purpose of the archiver is to collect the control parameters using CA protocol from various Input Output Controllers (IOC) distributed over network and then it stores the values into a database. The user interface for retrieving the archived data for analysis purpose is also an integrated part of the archiver. The software architecture and other components of EPICS MySQL Archiver are described below.

Architecture

The software architecture of EPICS MySQL Archiver is shown in Fig.1. A multilayered architecture with a common memory resident database is adopted for this application.



Figure1: Software architecture of EPICS MySQL Archiver.

The top layer of the software is consists of a CA Client interface and a Web server interface. The CA Client interface is dedicated for acquiring the values of the EPICS based parameters through CA link. The intermediate layer is a memory resident real time database (RT DB) keeping the latest values. The lowest layer is MySQL client layer dedicated for communicating with MySQL database server using SQL commands. This layer is responsible for archiving data from RT DB layer into database and also for retrieving archived data on demand. A web based user interface is provided through Web server interface. The user interface is equipped with tools for viewing system and subsystem wise online status of

^{*}r_ani@vecc.gov.in

all archived parameters, searching historic data of multiple system / subsystem parameters for analysis and downloading of the historic data with date & time stamp as spread sheet for offline analysis. The third party systems are interfaced directly with MySQL database server through respective standard client tools supported by them.

The Archiver

The archiver is designed to log data along with date and time stamp, available in the database server, either in scan mode or in monitor mode. In scan mode, the process parameter data is read at a predefined interval using CAGet facility. The scan interval can be defined from 1 second to 999 seconds by the user. While in monitor mode, the archiver creates monitoring event to the respective IOC and receives data from IOC, whenever it changes. The archiver is implemented as a multithreaded application. There are dedicated threads for managing MySQL client interface for data insertion and retrieval, CA connection managements, CA callback, command prompt interface for debugging, a timer thread and web server thread. A dedicated thread is implemented to retrieve latest data, for display, from the database for non-EPICS systems logging data independently. There is user authenticated facility for modification, addition or deletion of logged parameters.

The open source libraries e.g. EPICS CA library, MySQL client library and 'gd' library are used for building the archiver. The MySQL client library is used for connection management with database, generating SQL queries and error handling. The gd library is required for generating historic plots as 'png' image. This format of image is chosen for the following reasons. Since the image of historic plot is to be uploaded along with web pages, minimising image footprint will improve the performance and 'png' results into minimum image size. Again historic plots are line plots with sharp contrast; hence png format is the best without any loss of information.

A configuration file, a text file containing the list of parameters to be archived along with system / subsystem names, archiving mode (scan / monitor), parameter type (EPICS / non-EPICS) and archiving interval (in case of scan mode), is required by the archiver for configuring the database. The archiver creates and maintains independent tables, with date column as the index column, at database for individual systems. The indexed tables enhance the searching efficiency. An example configuration file, used for Superconducting Cyclotron archiver, is shown in Fig.2. An example command, used from application 'Top' directory, to start the archiver is shown below.

bin/linux-XX/mysqlArch [host IP] [database name] [config file-1] [config file-2]...

where.

host IP = IP address of the machine running the archiver

database name = name of the database to be used for archiving the data, assumed to be residing in the same machine

config file -1 = configuration files, multipleconfiguration files are supported

```
system Vacuum ca
       sub Main machine
       pv VAC MM BC2 ca monitor 1
       pv VAC_MM_BC3 ca monitor 1
      pv VAC_MM_OVC ca monitor 1
pv VAC_LOLINER ca scan 1
pv VAC_UPLINER ca scan 1
      sub Vault_beam_line
pv VAC_BTS_GL1 ca scan 1
system CRYO_DEL_SYS nc
       sub PRESSURE
       py PI1 nc monitor 20
       pv PI3 nc monitor 20
       pv PI4 nc monitor 20
       sub TEMPERATURE
       pv TI1 nc monitor 20
       pv TI20 nc monitor 20
       pv TI3 nc monitor 20
       pv TI4 nc monitor 20
Figure 2: Archiver configuration file.
```

Web Interface

A multilayer web based user interface is embedded in the archiver for monitoring, analysis and retrieval of archived data. This facilitates users to interact with the archiver from anywhere in the control LAN without installing any application. The various facilities included in the user interfaces are system view, sub-system view, online monitoring of parameters and connection status, historic analysis of upto five parameters of different systems, downloading of archived data and authenticated access to add/delete/modify parameters. The web interfaces of the above facilities are shown in Fig.3 to Fig.5.

MySQL Server

MySQL Server, Version 5.0.22, is deployed for archiving control system data in SCC. Control system history is stored in transaction safe InnoDB format. Database tables are placed in host local storage. An Auto backup mechanism is employed for protecting the archived data from hardware failure. In this mechanism, a snapshot of the whole database is stored in local storage and in tape drive using database dump technique on each day. System backups are taken on weekly and monthly basis also.

MySql Archiver's Web Server (refres	rate = 10 sec.) - Micr	rosoft Internet Ex	plarer				- 2
e Edit View Favorites Tools Help						4	
3 Back • 🕥 - 💌 😰 🏠 ,	🔎 Search	ekes 🚱 🍰•	👌 🖬 • 🗖	111 1	3		
idress 🔕 http://172.24.4.80/isys/Vacuum/	sub/Main_machine						9
	l	EPICS 1	MySql /	Arch	iver		
			Historic plot				
		<u>S</u> ub-sy	ystem: Vacuu stem: Main_n	m nachine			
Nam	e Value	Sub-sy	ystem: Vacun stem: Main_n Date stamp	m nachine State	Log type	Description	
Nam VAC_MM_BC	e Value 2 1.110e-07	Sub-sy Time stamp 15:24:33.304	ystem: Vacum stem: Main_n Date stamp 2012-11-22	machine State OK	Log type scan	Description	
Nac VAC_MM_BC VAC_MM_BC	e Value 2 1.110e-07 3 6.608e-08	Sub-sy Time stamp 15:24:33.304 15:24:33.304	ystem: Vacuu stem: Main_n Date stamp 2012-11-22 2012-11-22	machine State OK OK	Log type scan scan	Description	
Nam VAC_MM_BC VAC_MM_BC VAC_MM_BC	e Value 2 1.110e-07 3 6.608e-08 4 0.000e+00	Sub-sy Sub-sy 15:24:33.304 15:24:33.304 15:24:33.304	ystem: Vacuu stem: Main_n Date stamp 2012-11-22 2012-11-22 2012-11-22	state OK OK	Log type scan scan scan	Description	
Nam VAC_MM_BC VAC_MM_BC VAC_MM_BC VAC_MM_BC	e Value 2 1.110e-07 3 6.608e-08 4 0.000e+00 NA	<u>Sub-sy</u> <u>Tme stamp</u> 15:24:33.304 15:24:33.304 15:24:33.304	ystem: Vacuu stem: Main_n Date stamp 2012-11-22 2012-11-22 2012-11-22	machine State OK OK OK ?	Log type scan scan scan scan	Description	
Nat VAC_MM_BC VAC_MM_BC VAC_MM_BC VAC_MM_BC VAC_MM_BC	e Value 2 1.110e-07 3 6.608e-08 4 0.000e-00 8 NA 2 4.610e-06	Sub-sy Sub-sy 15:24:33.304 15:24:33.304 15:24:33.304 15:24:33.304	ystem: Vacuu stem: Main_n Date stamp 2012-11-22 2012-11-22 2012-11-22 2012-11-22	State OK OK OK OK	Log type scan scan scan scan scan	Description	
VAC MM BC VAC MM BC VAC MM BC VAC MM BC VAC MM BC VAC MM OW VAC LOLINE	Value 2 1.110e-07 3 6.608e-08 4 0.000e+00 8 NA 2 4.610e-06 8 7.225e-02	Sub-sy Sub-sy 15:24:33.304 15:24:33.304 15:24:33.304 15:24:33.304	ystem: Vacuu stem: Main_n Date stamp 2012-11-22 2012-11-22 2012-11-22 2012-11-22 2012-11-22	Machine State OK OK OK OK OK	Log type scan scan scan scan scan scan	Description	

Figure 3: Sub-system page.



Figure 4: Historic analysis page.



Figure 5: An example historic analysis view.

SYSTEM OVERVIEW

The example EPICS MySQL Archiver is deployed on a Dell PowerEdge 2950 server having dual Intel Xeon E5320 (2 x 4MB cache), Intel 5000X chipset, 4 x Hot swappable SAS drive of approximately 300GB with RAID 5 configuration. The system is running on RedHAT Linux with standard Linux Ext3 file system.

CONCLUSION

An EPICS MySQL Archiver is installed for archiving SCC control parameters. At present around 100 control parameters of various systems e.g. Vacuum, RF, Main magnet, Cryogen Delivery System (CDS), LHe plant, are being logged into the database with a frequency of 50 parameters per second. Among the various systems, the CDS is a non-EPICS system and logs parameter data directly to the database using a third party MySQL Client interface. As all parameters are being logged with a common time stamp available at database server, hence it helps to provide a snapshot with respect to time of all parameters irrespective of EPICS/non-EPICS system. This system is running satisfactorily for a considerable period of time resulting into a data volume around 200GB. It also helps to resolve various inter-system dependent issues in the cyclotron.

Although the example system is deployed on Linux operating system, the same application can be ported on Windows operating system.

USING MEMCAHED AS REAL-TIME DATABASE IN THE SPARC **CONTROL SYSTEM**

G. Di Pirro, E.Pace, INFN LNF, Italy

Abstract

The first implementation of the SPARC control system was based on a distributed TCP/IP data server: each frontend CPU had its own server to distribute data to the console. We decided to move the system to a NoSOL key value database. We decided to use an open source database Memcached. This is a database that is high performance key-value cache optimized for speed only. For this reason we could use memcached not for storing data, but as a channel of communication between frontend processors and consoles. The first object that we have installed is the camera system. We chose this class of elements because the amount of data is high; cameras are at least 640x480 with 8 bit. In this first installation we made some speed test: we increased the speed transfer and the data transfer is now independent from the number of high level CPUs that are using the same image. The success of this installation convinced us to bring the entire data transfer of SPARC control system to use Memcahed as data server.

SPARC

The SPARC*[1] (Sorgente Pulsata e Amplificata di Radiazione Coerente, Self-Amplified Pulsed Coherent Radiation Source) (Fig.1) project is to promote an R&D activity oriented to the development of a high brightness photo injector to drive SASE-FEL experiments at 500 nm and higher harmonics generation. Proposed by the research institutions ENEA, INFN, CNR with collaboration of Universita' di Roma Tor Vergata and INFM-ST, it has been funded in 2003 by the Italian Government. The machine is installed at Laboratori Nazionali di Frascati (LNF-INFN). It is composed by an RF gun driven by a Ti:Sa laser to produce 10-ps flat top pulses on the photocathode, injecting into three SLAC accelerating and 6 undulator sections.

PHI Figure 1: SPARC. The SPARC project is financially supported by the

EU Commission in the 6th FP, contract no. 011935 EUROFEL and contract no. RII3-CT-2003-506395 CARE

CONTROL SYSTEM DESCRIPTION

The control system should guarantee and simplify machine operation. In general the main operations in an accelerator control system are: data taking, display of information, analysis, command execution and storage. The simplest and functional control system has distributed processors on a classic three levels architecture (Fig. 2).

- *First level*: At this level we find the console with its human interface to allow the operator to control the machine, a logbook to share information within the collaboration, a database to store all information coming from the machine and a web tools to help the management of the control system and to share some information outside the collaboration:
- Second level: At this level we find the front-end CPU that executes commands and handle all the information about the status of the machine available at the first level. Meanwhile it automatically saves data from its various elements in two ways: on value changes and/or at fixed time intervals;
- Third level: This is the acquisition hardware where we find an appropriate acquisition board or the secondary field bus to acquire data from the real element.

The interconnection bus between the levels is a Gigabit Ethernet LAN.



Figure 2: Control System Structure.

ELEMENTS

The control system allows to control all machine elements (from the Laser until undulator) and their diagnostic instruments.

Table 1 describes the kind of elements under control and their own interface.

database. In a relational database is the stiffness in the definition of the data that have lengths required.

I	Table 1. Element Controlled				
Element	Number	Interface			
RF Modulator	2	TCP/IP			
RF Low Level	1	PXI Digitizer			
Vacuum Pump	30	DAC, ADC, IO, Serial			
Vacuumetr	12	Serial			
MagnetPS	50	Serial, ModBus			
Flag	24	Serial Motor			
Camera	24	IEE1384, GigaEth			
BP M	12	Bergoz ADC			
BCM	2	Bergoz DVM			
Faraday Cup	1	High speed digitizer			
Laser Photodiode	1	High Speed Digitizer			
Filter wheel	3	Serial motor			

TT 1 1 TT

Element Abstraction

To control an element means control its hardware interface. For example to control the magnet mean control its power supply.

In this case we make an abstraction of the power supply to allow include the main characteristic of the element and we define a class.

We have two kind of cluster, (see Fig. 3) one contains all information that is necessary to control the power supply (called static part). It contains the type of interface, the eventual conversion factor and some parameter such as min max value and so on. The second cluster contains dynamic information such as the current, the voltage, the status and the error.

MEMCACHED AS COMMUNICATION INFRASTRUCTURE

In a control system is natural developing a communication channel between the front-end CPUs and the user based on a simple client server mechanism.

This communication system has a weak point in performance. The system can be influenced by the number of customers that require data to the server running on the front-end, and the possibility that the acquisition process is stopped or the computer is switched off. This second case leads to a general slowdown of the console display and in system performance due to timeout.

A method for exchanging information between a system that produces data and one using data is a



Figure 3: Magnet Clusters.

Development, especially in the field of web, nonrelational database type key value NoSQL db, largely simplifies these problems. These systems allow you to associate a value with each key.

A NoSOL database allows you to store information of variable length within the same database as there is a simple association between a key and its associated data.

The non-relational databases have received an impulse in the development of Web applications and cloud where, given the high number of nodes, it is difficult to define a priori the tables. Their simplicity and no need to predefine the tables make the system very flexible in the implementation.

There are several versions of NoSQL databases in particular we want try the system based on RAM for speed reasons We chose an open source implementation of a NoSOL based on ram called Memcahed.

Memcached [2] is a NoSOL key-value database. It is a high performance key-value cache. It is intentionally a dumb cache, optimized for speed only. Applications using memcached should not rely on it for data, a standard database guarantee, but applications can run much faster when cached data is available in memcache for the display.

IMPLEMENTATION

Transferring the information we decide to transfer the whole cluster because the transfer time is similar to transfer a single information (the normal dimension of a dynamic cluster is normally less than 1500 byte). The choice of data is performed by the console software. If we need to transfer a big amount of data, i.e. image or waveform, we developed a specific data tape. All the data are transferred in binary code,

The communication protocol with the database is easy to develop we write some subroutines directly in TCP/IP without using Memcahed libraries. This choice has allowed us to use memcahed also for operating systems where the libraries are not developed.



Figure 4: Packet Structure.

The protocol is simple: in Fig. 4 you can see the description of the packet. The transmission standard TCP/IP the only requested part is the header part all other are option. There are two kind of header one for request information (Fig. 5) and we have as answer header data (Fig. 6). The answer has different length in function of the request.

MAGIC	Opcode	Key Length				
(1 byte)	(1 byte)	(2 bytes)				
Extra Length	Data Type	Reserved				
(1 byte)	(1 byte)	(2 bytes)				
Total Body Leng	th					
(4 bytes)						
Opaque						
(4 bytes)	(4 bytes)					
CAS (Compare a	CAS (Compare and Swap)					
(8 bytes)						

Figure 5: Request Header.

We have developed a library in Labview that allow data transfer to the database so that easily integrate in our system of control.

Each element has been associated with a key and the data were written in binary.

MAGIC	Opcode	Key Length
(1 byte)	(1 byte)	(2 bytes)
Extra Length	Data Type	Status
(1 byte)	(1 byte)	(2 bytes)
Total Body Leng	th	
(4 bytes)		
Opaque		
(4 bytes)		
CAS (Compare a	nd Swap)	
(8 bytes)		
	MAGIC (1 byte) Extra Length (1 byte) Total Body Leng (4 bytes) Opaque (4 bytes) CAS (Compare a (8 bytes)	MAGIC (1 byte)Opcode (1 byte)Extra Length (1 byte)Data Type (1 byte)Total Body Length (4 bytes)Opaque (4 bytes)Opaque (4 bytes)CAS (Compare and Swap) (8 bytes)

Figure 6: Response Header.

The first object that we have tested is the camera system. We chose this class of elements because the amount of data is high (our camera is at least 640x480 with 8 bit). In this first installation we make some speed test: we increased the speed transfer and we have the data transfer independent from the number of consoles taking the image. The success of this installation convinced us to bring the entire data transfer of SPARC control system[3] to use memcache as data server.

The performance of the image transfer is very good we can transfer any kind of real time image with the follow time:

Image 640x480 16 bit 40 ms sample variance 8 ms Image 640x480 8 bit 23 ms sample variance 8 ms

We do not see variation on performance from 1 to 4 consoles in simultaneous acquisition from the camera.

CONCLUSION

The use of database NoSQL there has demonstrated both in performance and in simplicity of realization. So it is possible to use such database basic data for a control system.

The future development will be in the evaluation into using a database like this but hard as a database for the data history.

ACKNOWLEDGEMENTS

We want to We are also grateful to all the SPARC staff for their continuous suggestions and encouragement for making a good and useful job.

- [1] SPARC Project Team, Sparc Injector TDR http://www.lnf.infn.it/acceleratori/sparc
- [2] http://memcached.org/
- [3] G. Di Pirro et al., "The SPARC Control System", THP042, http://jacow.org, Proc. of ICALEPCS2009, Kobe, Japan.

CONTROL SYSTEM INTEROPERABILITY, AN EXTREME CASE: MERGING DOOCS AND TINE

P. Duval, A. Aghababyan, O. Hensler, K. Rehlich, DESY, Hamburg, Germany

Abstract

In controlling large facilities one is rarely able to manage all controllable elements via a common control system framework. When the standard framework must deal with numerous 'foreign' elements it is often worthwhile to adopt a new framework, rather than 'disguising' such components with a wrapper. The DOOCS[1] and TINE[2] control system frameworks fall into this scenario. Both systems have a device server oriented view, which made early mapping attempts (begun in 2000) immediately successful. Transparent communication, however, is but a small (albeit important) part of the control system merger currently taking place. Both systems have well-established central services (e.g. archiving and alarms), and possess a general 'culture' which might dictate to a large extent how something is actually 'done'. The long term goal of the DOOCS/TINE merger is to be able to make use of any tool, from either the DOOCS or TINE toolbox, on any control system element.

We report here on our progress to date, concentrating on the REGAE accelerator, and plans for the XFEL accelerator (to begin commissioning in 2015).

INTRODUCTION

'Interoperability' is a bit of a trendy word these days and it is important to be clear at the outset what we mean by 'control system interoperability'.

Any control system framework will likely provide interfaces to popular scientific and engineering software such as MatLab and LabView as well as popular user utilities such as Python, Java, .Net, and the like. If these interfaces are not native to the software in question then one speaks of 'interoperability' with regard to allowing the control system to interface ('interoperate') with such external software packages. In this paper, however, we refer to 'interoperability' as being that between the different control system frameworks themselves.

Since circa 1990 control system frameworks have been typically recognized by their names rather than, say, 'the control system they use at KEK'. Likewise there has been a strong tendency for institutes to adopt an existing controls framework, rather than 'inventing their own'. The most popular of these is EPICS[3]. There are nonetheless a large number of institutes which base accelerator control on something else, for example TANGO[4], ACS[5], STARS[6] or, our primary focus here, TINE[2] and DOOCS[1].

Consequently when the primary control system is not, for instance, EPICS it often occurs that, over the course of operations, some provision must be made to interface to exotic EPICS elements which invariably creep into the system. This is in fact one of the primary motivations for pursuing interoperability. Experiments and test equipment from other facilities can suddenly introduce timelines, not to mention complexity, which necessitate seamless, rapid, and robust integration of foreign components into a control system. Epics2tine [7] is one of the first attempts to do this systematically. Since then, a number of translation interfaces and gateways such as tango2tine, epics2tango, etc. have been available.

In this vein, a doocs2tine translation layer was embedded directly into the DOOCS libraries in the year 2000. This constituted the primary step in the eventual control system merger now taking place.

Below we will first discuss what the interoperability between control system frameworks might mean in general and then give specific details concerning what it means to merge two relatively distinct control system frameworks. We note here that this goes far beyond the simple ability to 'trade data'.

CONTROL SYSTEM FRAMEWORK INTEROPERABILITY

There are in principal three ways to go concerning the interoperability between two distinct control system frameworks [8]. If System A refers to the primary control system framework, then each of these interoperability methods amounts to translating requests from System A into System B language, obtaining results, which are then translated back to System A language. This can be achieved by a stand-alone gateway process, by incorporating the translation layer directly within the System A client-side API, or by incorporating the translation layer within the System B server-side API. The relative merits of these approaches have been discussed before [8]. Solutions such as the Joint Controls Project (JCOP) [9], Control System Studio (CSS) [10], or java DOOCS Data Display (jddd) [11] focus on the second method listed above. We note here that the third method, server-side translation layers, being the most invasive is also the most demanding, as the introduction of any new software (the translation layer) on the front-end elements places these critical components at new risk. Nevertheless, it is precisely this third method which allows a control system merger to take place in the first place and is the key to the DOOCS/TINE merger we now describe below.

MERGING DOOCS AND TINE

Device Servers versus Databases

Control system frameworks have a general perspective concerning the accelerator control points. Some, such as,

EPICS or VISTA [12], have a database view of the controllable elements, where one thinks of 'getting' or 'setting' (or 'monitoring') some item in a database. Others, such as TANGO, TINE, and DOOCS, have a device server view of the controllable elements, which are regarded as devices at some location. Here one thinks of calling the methods of some device. That both TINE and DOOCS both have a device server perspective makes the task of merging the two considerably less daunting. DOOCS and TINE also have a three-tier naming hierarchy to identify a 'device' along with a property name to identify a 'method'. Unlike DOOCS, however, TINE elements can also take on a 'property server' view, whereby a server does not represent an interface to a device collection so much as a service with properties, each of which in turn might refer to a different collection of keywords. We shall come back to this point below.

Request-Response Translation

The request-response translation between DOOCS and TINE is straightforward as long as both systems agree on the contents of the data being transferred. The early doocs2tine layer in fact concentrated on ensuring that the set of data types used in DOOCS were matched in TINE and vice versa. Besides the standard primitive data types, both systems also provide compound data types for atomic transfer (e.g. a name, a float, and an integer value). Such data types must of course exist in both systems. TINE also allows user-defined structures, which are not directly supported in DOOCS and presents a potential problem. However, the individual fields of a TINE structure are accessible via the normal DOOCS API.

At this point in the merger (~ 2001), all DOOCS servers are now 'visible' and accessible to TINE clients and all TINE servers are visible and accessible to DOOCS clients. That is, we now have the ability to 'trade data', and in a systematic way. In fact, the full gambit of the efficient transport techniques available in TINE (e.g. asynchronous communication, contract coercion [13]) are now available in DOOCS via the TINE protocol.

Culture Shock

In practice, although both systems offer rich client programing, Servers in a DOOCS-centric facility such as FLASH are usually accessed via *ddd* or *jddd* [11] panels, which are 'simple' clients with data acquisition and display widgets. Servers in a TINE-centric facility such as PETRA III are usually accessed via rich clients written in java, using RAD (Rapid Application Development) tools such as ACOP [14]. A successful merger implies that a client developer can remain in his culture of expectations and be unaware of the idiosyncrasies of either framework.

The panel approach tends to place the burden on the Server developer to provide data 'ready to display', which is not a bad thing. It also tends to decouple the panel developer from making data update decisions. In the early days, a *ddd* panel would synchronously poll a TINE server even though a more efficient asynchronous communication was available. In addition, TINE server developers have been known to overload specific method calls, delivering differently encoded data based on the requested data type and input. A panel application accessing such a method will only access the 'default' method call.

Such considerations really only provide caveats to the client application developer and do not impact per se on a merger of the two systems. What does impact more strongly is the inherent control system browsing within the panel builders and other browsing tools. Here naming conventions and cultures along with browsing logic play a strong role in meeting expectations.

As noted above, TINE also supports 'property servers'. Browsing such servers requires querying the keywords of a property as opposed to querying the properties of a device, as is the case with device servers. Although the naming hierarchy remains the same, such browsing logic must be incorporated in the relevant DOOCS utilities in a DOOCS-centric system with TINE property servers.

Infrastructure

Assuming we have addressed request-response mapping and the culture shock aspects of client applications communicating with a mixture of DOOCS and TINE servers, can we claim to have merged the two control systems? We have of course achieved something remarkable, but the answer to this question remains a resounding 'no'. What still needs to be considered is the infrastructure aspects behind the frameworks.

Archiving

An accelerator control system will have an archive system, an alarm system, naming services, and security to go along with the general culture and behavioral aspects and expectations of a user within either a DOOC-centric or TINE-centric facility.

Both DOOCS and TINE provide a local history subsystem, where the history of specific properties can be acquired directly from the servers, and there are utilities in both DOOCS and TINE which can access and display this information. However, each utility is expecting functionality which may or may not be present depending on the pedigree of the server. At the time of this writing, the expectations of either culture are approximately only 50 per cent met, with archive reading utilities often making 'if that didn't work, then try this' decisions. We will not discuss the TINE Central or Event archive systems nor the DOOCS DAQ system at this juncture, except to note that these additional add-on services do not reflect on the merger status.

Alarms

Alarm mapping was introduced in 2009 and is by and large successful. We note that DOOCS servers 'push' alarm information to a central server, whereas TINE servers set alarms which are then 'pulled' by a central server. The alarm mapping consists then of DOOCS servers setting alarms for access via the TINE central

alarm server and for the TINE central alarm server to push selected alarms to the DOOCS central alarm server. The alarm utilities of either system can then be used to view alarms.

Naming Services

Naming servers for both DOOCS and TINE are similar in that the address of a specific device server, based on its context and server name are resolved centrally with the results being returned to the caller. Device and property information is then obtained directly from a specific server, meaning that the server must be on-line to receive that latter information. The principal complication to this scenario occurs when the device server in question is not a device server residing on a single host but is instead a device group. In DOOCS such configurations are handled administratively, whereas in TINE they are usually handled via plug-and-play. The group server mapping is done seamlessly as long as the proper information is provided within a DOOCS server's configuration file.

Security

Security can be a real show-stopper. DOOCS security is based on a *unix*-style gid and uid (group ID and user ID) access mask of the caller, whereas TINE security is based on the caller's user name and/or the network address. Where gid and uid information is unavailable, DOOCS servers attempt to match the caller's user name with available NIS or LDAP information in order to ascertain it. This approach works fine except in the case where a TINE middle layer server is attempting to issue a command to a DOOCS server. In such cases the user name of the caller is then the TINE Middle-Layer FEC (Front End Controller) name, which is definitely not a user name to be found in any NIS or LDAP table. Thus commands from such a Middle Layer are rejected. To overcome this difficulty, TINE servers now note whether a specific call is directed at a DOOCS server and if so supply the original user name of process in the command request.

Turing Tests

On could speak of undergoing Turing tests at various levels in order to determine the state of a merged system. Would a client programmer using his favorite development tool be able to distinguish between a DOOCS server and a TINE server? Do utility applications such as alarm or archive viewers behave differently depending on the flavor of the framework being used? Do remote process control applications, such as front end watchdogs, depend in any way on which kind of server process is being monitored?

The tacit goal is of course to be able to answer 'no' to all of the above questions. In reality an expert will always be able to detect differences. However the degree to which these Turing tests are being passed is sometimes remarkable, particularly as concerns the lay user. To be sure, a browsing tool suddenly indicating a *property server* is a dead giveaway that the target must be a TINE server, as would be a target property indicating a structure data type. Alarm viewing applications on the other hand do not readily distinguish between DOOCS and TINE alarms. And although archive functionality mapping is not yet complete archive viewing applications likewise do a remarkably good job displaying data. One can now, for instance, drag and drop from a *jddd* panel into the TINE archive viewer. Framework independent remote process control is currently being addressed.

Status

FLASH is a DOOCS-centric facility but has long had native TINE servers in control, notably for the magnets. PETRA-III is a TINE-centric facility but likewise makes use of native DOOCS servers, notably in the vacuum subsystem. The 'exotic' cases here have over the years had their (mostly minor) issues, but could always be dealt with on a special basis.

The Relativistic Electron Gun for Atomic Exploration (REGAE) facility at DESY provides an excellent test bed for determining our progress in the DOOCS/TINE merger as it consists of a good mixture of TINE and DOOCS servers, as well as a good mixture of TINE rich client applications, *jddd* panels, and MatLab applications in the control room. In REGAE, virtually all DOOCS servers are communicating only via the TINE protocol, even when contacted by a *jddd* panel.

After an initial period of 'growing pains', operations in the REGAE control room have been smooth for well over a year, demonstrating the current success of the merger. This bodes well for the X-ray Free Electron Laser (XFEL) project currently underway at DESY.

- [1] DOOCS; http://doocs.desy.de
- [2] TINE; http://tine.desy.de
- [3] EPICS; http://www.aps.anl.gov/epics
- [4] TANGO; http://www.tango-controls.org
- [5] ACS; http://www.cosylab.com/solutions/ICT/ACS/
- [6] STARS; http://pfwww.kek.jp/stars
- [7] "An EPICS to TINE Translator", Z. Kakucs, et al., PCaPAC 2000 Proceedings.
- [8] "The Babylonization of Control Systems", P. Duval et al., ICALEPCS 2003 Proceedings.
- [9] JCOP; http://en-dep.web.cern.ch/en-dep/internal/JCOP/
- [10] "The CSS Story", M. Clausen et al., These proceedings.
- [11] jddd; http://jddd.desy.de
- [12] VISTA; http://www.vista-control.com
 [13] "The TINE Control System Protocol: How to Achieve High Scalability and Performance", P. Duval and S. Herb, PCaPAC 2010 proceedings.
- [14] ACOP; http://public.cosylab.com/acop/site/

TANGO FOR EXPERIMENT CONTROL

J.Meyer, L.Claustre, S.Petitdemange, O.Svensson, A.Götz, ESRF, Grenoble, France T.Coutinho, J.Klora, ALBA, Barcelona, Spain F.Picca, M.Ounsy, A.Buteau, SOLEIL, Paris, France

Abstract

The Tango control system framework allows you to control an accelerator complex as well as single equipment. The framework contains the communication bus with the standard communication modes (synchronous, asynchronous, event driven) as well as the basic hardware access modules, GUI tools and development kits, bindings to commercial products (LabView, Matlab, IgorPro) and services (administration, archiving, access control) to set up a control system.

Tango was mainly developed by several synchrotron light sources that have to support not only the accelerator complex but also a lot of experimental end stations. For synchrotron experiments we have to control the whole process from basic hardware access over data taking to data analysis.

This paper describes in the first part the special features of Tango allowing flexible experiment control. The dynamic configuration, the rapid hardware interface development and the sequencing and scanning framework are some examples.

The second part gives an overview of some packages developed in the Tango community for experiment control: A HKL library for diffraction computation and diffractometer control, a library to control 2D detectors and a data analysis workbench with workflow engine for on-line and off-line data analysis. These packages are not part of Tango and can be used with other control systems.

TANGO BASICS

What is Tango?

Tango [1] is a control system tool kit developed by a community of institutes. It is object oriented with the concept of devices (objects) for each piece of hardware or software to be controlled. Tango classes are merged within operating system processes called Device Servers. Three types of communication between clients and servers are supported (synchronous, asynchronous and event driven).

But Tango is not only the software bus which handles the communication between device servers and clients. The Tango tool chain offers software from the hardware interface to the graphical user interface for several programming languages as shown in Table 1.

Tango utilities are available, with the basic installation, for code generation, device configuration and testing and for administration and survey of a whole Tango control system.

An archiving and a configuration snapshot system usable with Oracle or MySQL are also available.

Module	Description
Core Libraries	Client/Server communication libraries for C++, Python and Java
Device Classes	About 300 hardware interface classes are available to download [2]
GUI Frameworks	Available for C++ and Python using QT, for Java using Swing and a web interface written in PHP
Client Bindings	LabView, Matlab and IgorPro
Tools	Pogo – Code generator for device classes in C++, Python and Java
	Jive – Configuration and testing tool
	Astor – Administration and survey of the Control system
Archiving	Archiving and snapshot system with GUIs and web interface. Usable with Oracle and MySQL
Alarm System	Event driven alarm service
Sardana	Framework for experiment control : Interface standardization, configuration, sequencing, command line interface

Rapid Interface Development

Important for flexible experiment control is a short development cycle to interface and integrate new hardware into the control system. Tango with its code generator (Pogo) offers a very comfortable way of implementing new hardware interfaces. The graphical interface builder allows the user to design the network interface and the main functionality (state machine, refreshing rate, memorization, etc.) of a device server.

The device server code can be generated in three programming languages: C++, Python and Java. The user has only to fill-in the code necessary to access the underlying hardware. For any change of the interface or for example the state machine, Pogo can be used to modify the existing code (Fig. 1).

Dynamic System Configuration

For experiment control systems distributed over several hosts it is very handy to have a centralised way to survey and to reconfigure running software. Tango offers such an administration system which is part of the core package. Two parts are necessary, the Starter server which is running on the distributed hosts and the graphical administration interface Astor (Fig. 1).



Figure 1: Astor and Pogo.

With Astor it is possible to survey all running servers on the distributed hosts, start or stop servers, run generic test tools, start the configuration interface (Jive) or set-up the logging service on devices. Astor is designed as the central entry point to the distributed control system.

Scanning and Sequencing

The Tango framework offers two ways to prepare scans and sequences for an experiment.

1. The SARDANA project [3], developed at ALBA, offers a generic user environment for scanning and sequencing in Python.

Today the user environment consists of a highly configurable standard graphical user interface (Fig. 2), a command line interface understanding SPEC [4] commands, and a way to compose new applications either by programming or with a graphical interface builder. It further consists of a Python macro executer, a standard set of macros, a range of common hardware types (like motors, counters, cameras, etc.) and a configuration editor to set all this up.



Figure 2: Sardana Configuration GUI.

2. At SOLEIL a Tango scan device server [5] was developed which offers sophisticated scanning and data acquisition functionality. To implement experimental procedures the graphical work flow editor Passerelle [6], was customized (Fig. 3). This allows even non programmers to prepare sequences to run an experiment.



Figure 3: Passerelle Work Flow Editor.

PACKAGES FOR EXPERIMENTS

On top of the basic Tango control system several packages are available, inside the collaboration, which can be helpful for experiment control. When using Tango as control system the package integration is immediate. But they are Tango independent and can be adapted to other underlying systems.

Diffractometers

The Soleil synchrotron in Paris, France developed a Clibrary for reciprocal space transformations [7]. The purpose of the library is to factorise single crystal diffraction angles computation for different kinds of diffractometer geometries (Fig. 4).



Figure 4: Reciprocal space transformation.

The main features are:

- Mode computation
- UB matrix computation
- Crystal lattice refinement
- Pseudo axes (psi, eulerians, q, ...)

Today the HKL library can handle 5 different geometries:

- 2 circles.
- Eulerian 4 circles
- Eulerian 6 circles
- Kappa 4 circles
- Kappa 6 circles

and supports operation modes as:

- Bisector
- Constant omega, chi, phi

For a usage within the Tango control framework a configurable device server based on the HKL library and the corresponding graphical user interface (Fig. 5) are available for download [8].



Figure 5: Diffractometer GUI.

2D Detectors

A significant number of 2D detectors are used in large scale facilities' control systems for quantitative data analysis. Common control parameters and features can be identified in these devices, but most of the manufacturers provide specific software control interfaces. A generic image acquisition library, called LImA [9], has been developed at the ESRF for a better compatibility and easier integration of 2D detectors to existing control systems.

Detector	Interface
Basler	GigE
Prosilica	GigE
Ueye (IDS)	GigE
Pilatus (Dectris)	300K, 1M, 2M, 6M, 6M-F
PCO Edge	Camera-Link
Pco Dimax	GigE + Camera-Link
PhotonicScience 4022	USB
RoperScientific	PVCAMSDK
Andor I-Kon	USB
PerkinElmer Flat-Panel	Proprietary board
ADSC 315r	Proprietary board
MarCcd 165	Proprietary board
Mythen (strip detector)	Proprietary board
XPAD	Proprietary board
Maxipix (ESRF)	Espia
Frelon 2K (ESRF)	Espia

Figure 6: LImA Detector Plug-ins.

The main design goals are:

- Control system independence
- A rich set of common functionalities, providing alternative software solutions when features are not implemented by the hardware.
- Intensive use of multi-threaded algorithms, synchronised by events for ensuring the performance of high throughput detectors.

Today we have already a big number of available detector plug-ins for LImA developed by the Tango community (Figs. 6 and 7).



Figure 7: LImA Internal Structure.

Data Analysis Work Bench

The Data Analysis WorkbeNch (DAWN) [10] is an eclipse [11] based workbench for doing scientific data analysis (Fig. 8). Developed in collaboration by Diamond and the ESRF, it implements sophisticated support for:

- Visualization of data in 1D, 2D and 3D
- Python script development, debugging and execution
- Workflows for analyzing scientific data calling Python and binary codes (implemented with the Passerelle work flow editor [6] : see Figure 3)

DAWN is not restricted to one scientific domain. It is developed by and for the synchrotron community foremost but has strong overlap with other communities like neutron scattering, photon science and any scientific communities with the above or similar needs.

DAWN is being used at the ESRF and Diamond for automating online and offline data analysis.



Figure 8: Data Analysis Workbench.

- [1] http://www.tango-controls.org
- [2] http://www.tango-controls.org/device-servers
- [3] http://www.tango-controls.org/static/sardana
- [4] http://www.certif.com
- [5] http://tango-ds.cvs.sourceforge.net/cvsroot/tango-ds/ InputOutput/Scan
- [6] http://code.google.com/a/eclipselabs.org/p/passerelle
- [7] http://people.debian.org/picca/hkl
- [8] http://tango-ds.cvs.sourceforge.net/cvsroot/tango-ds/ Instrumentation/Diffractometer
- [9] http://lima.blissgarden.org/
- [10] http://www.dawnsci.org
- [11] http://www.eclipse.org

CONTROLS ARCHITECTURE FOR THE DIAGNOSTIC DEVICES AT THE EUROPEAN XFEL

O.Hensler, A.Aghababyan, L.Petrosyan, V.Petrosyan, K.Rehlich, DESY, Hamburg, Germany

Abstract

The European XFEL X-ray laser is a 3.4-km-long facility, which runs essentially underground and comprises three sites above ground. It will begin on the DESY site in Hamburg, Germany and runs mostly below surface to the XFEL research site, which is to be erected south of the town of Schenefeld. First electron beam is expected in 2014 in the injector and in 2015 through out the whole facility. The XFEL will be an electron accelerator and its electron beam will be used to create intense photon beam in long undulator sections to be used in various research fields.

For controlling all diagnostic devices like a toroid, BPM or BLM, it is planned to use the new MTCA.4 crate standard instead of VME. ATCA is an emerging standard from the Telecom Industry and adapted with the PICMG MTCA.4 branch for physics usage. The communication on the backplane utilizes the high-speed serial PCIe communication plus precise clock lines and SATA interface. The MTCA.4 hardware supports hot-plug mechanism and remote monitoring and control via IPMI over Ethernet. Some of the diagnostics will be connected to 16Bit ADCs with up to 125Mhz sampling rate from Struck Company or to an internal DESY development call DAMC2. The software architecture is based on the DOOCS control system known from the FLASH accelerator. The raw data from the ADCs will be read via DMA transfer by one server process. Then this raw data will be distributed locally on the CPU using a message passing system based on the ØMQ project. The receiving server processes in order to calculate these data into engineering units. Everything works in an event driven way.

The current hardware and software concept will be presented.

INTRODUCTION

At FLASH over the last 15 years, VME systems have been used for all kinds of readout and control of fast diagnostics like BPM, toroid or wirescanner. The more then 30 years old VME standard has several drawbacks e.g. the parallel bus, no standardized crate management and it is relative expensive. So, it was considered to move to a more modern crate standard for the European XFEL [1] project. Since FLASH [2] is the "test bench" for XFEL, it was decided to install the same hardware and software for the FLASH2 extension and migrate slowly the old VME systems to the μ TCA one.

Since 2002, the new Advanced Telecom Computing Architecture (ATCA) emerged on the market, which is based on PC technology and serial communication links.

In addition to ATCA, a smaller version of it, based on the ATCA mezzanine cards, called μ TCA was defined. This standard was extended even further to MTCA.4 [3], formerly known as μ TCA for Physics.

The reasons, why μ TCA was chosen for the XFEL are the following:

- ∞ Modern standard based on common PC technology
- ∞ Fast serial communication via PCIe
- ∞ Standardized crate management via IPMI
- ∞ Hot Swap capability
- ∞ Capable of redundant crate layout
- ∞ More cost efficient then VME

DOOCS

The Distributed Object Oriented Control System DOOCS [4] is the leading system for the FLASH accelerator and chosen for the XFEL. DOOCS is a common client/server control system and based on an object-oriented approach at the front-end/server and client/display side. It is mainly implemented in C++, but there is now a Java client-side implementation available called jDOOCS, which is the communication interface for the new display tool jDDD [5]. An interface for MATLAB clients is provided. The communication protocol is based on ONC Remote Procedure Calls (RPC), but efforts are on the way to replace it by the TINE [6] protocol. The client interfaces of EPICS and TANGO are integrated.

µTCA HARDWARE

The μ TCA crate is a 19" form factor frame to accommodate the AMC cards. There are already several versions of these crates on the market, which allow sliding in the AMCs in horizontal or vertical orientation. One may purchase simple desktop test crates or full-featured 19" versions with 12 slots with the possibility of redundant fans, power supplies, CPUs or MCHs.

MTCA.4

For the XFEL project, the standard is enhanced to host vertical orientated AMC boards with double height and the possibility to attach Rear Transition Modules (RTM). These RTMs are used to connect IO signals and to allow signal-conditioning circuitry. Compared to 6U VME boards, it offers about 30% more PCB space. The idea of this standard extension is to add features, which are required especially for industrial and physics applications, like precise clock distribution over the backplane. This standard is called MTCA.4 and defined by the PICMG organization in 2011.

Used commercial AMC modules at DESY

- ∞ Several CPU modules from different vendors with an Intel or PowerPC CPU
- ∞ Hard disk carrier for 2.5" SATA drives
- ∞ IndustryPack module carrier for one or three modules
- ∞ SIS8300: 10-channel 125MSample 16bit ADC from Struck Company plus RTM analog interface
- ∞ MCH (MTCA Carrier Hub)
- ∞ Slow Analog and Digital IO
- ∞ CANbus. 2 lines
- ∞ Ethernet/Telco interfaces
- ∞ Power Supplies

Available in-house AMC modules at DESY

- ∞ XFEL timing system interface
- ∞ XFEL MPS (Machine Protection System) interface
- ∞ DAMC2, Virtex5 board with SFP connections
- ∞ LLRF controller
- ∞ LLRF RTM 1.3GHz down converter

Several more RTMs are in design phase at DESY.

IPMI CRATE MANAGEMENT

One of the key features of µTCA is the ability to monitor and control the whole system by a well-defined management, independent from the CPU and the operating system, called Intelligent Platform Management Interface (IPMI). This crate management is done locally by the MircoTCA Controller Hub (MCH), which controls the power supplies, manages the 12V supply of every module including the CPU, enables the hot-plug functionality, takes care of the PCIe switching between AMC modules, has Ethernet switching capabilities and provides an IPMI interface to the external Ethernet. Every AMC module has to provide a well-defined set of information, like an ID, a serial number or its power consumption, otherwise it is not accepted by the MCH and no 12V is supplied. The IPMI protocol runs over Ethernet allowing external tools for monitoring. An IPMI channel to connect to the console output and to the BIOS of the CPU is under development.

A DOOCS server, running on a separate middle-layer computer, was developed to readout and control all the properties of the µTCA crate. This server is able to detect the insertion or removal of AMCs, creates new server locations and starts monitoring this new card. The jDDD panels are then able to reflect these changes dynamically, so that these panels will always show the actually installed modules.

SERVER ARCHITECTURE

The software concept of front-end DOOCS servers running on a µTCA crate will be explained by the example of a toroid readout in the following. Very similar implementations are planned for BPM, BLM and other fast diagnostic devices, which require bunch resolved readout. To provide a transparent transition to the new server, a new facility was created in the DOOCS naming system, which allows keeping the same device, location and property names.

As operating system, Ubuntu Linux 12.04 Long Time Support (LTS) has been chosen and an installation service is provided to allow network booting and crash recovery.

In order to read the toroid signal, three DOOCS server will be involved and described here in the order of operation.

- ∞ The x1timer server is in charge of configuring the XFEL timer hardware through a Linux device driver interface. Via this driver, the server receives a SIGUSR1 interrupt just after a macro-pulse, which synchronizes the server with the accelerator repetition rate. After reading several information from the hardware, like the macro-pulse number or the timestamp, the server sends this data package to the SIS8300DMA server, using a message passing system.
- ∞ In order to configure the SIS8300 hardware, this server creates one location per 10-channel ADC module, which keeps all necessary hardware information. A 108MHz external clock clocks the ADCs, leading to a buffer size of 90000 16Bit words, which allows covering an 830µs long pulse. The SIS8300DMA server receives the data package from the x1timer via a callback function. Inside this callback function, the internal SIGUSR1 method is called to loop over all locations, where one location corresponds to one ADC board. Inside of this location, the DMA transfer is setup and the raw ADC data is stored in local 16bit arrays and tagged with the information from the x1timer server. This buffer is then passed to the messaging system to be pushed to the toroid server.
- ∞ At startup, the toroid server registers with one of the raw data channels of the SIS8300DMA server. After the raw data is read, this server sends the data to the toroid server in order to do the conversion to a charge reading. The data is reduced here from the 108MHz sampling to one data value per bunch. As FLASH runs with up to 9MHz bunch repetition rate, a float buffer of 7200 values will be created. In addition the toroid server is computing the number of bunches and archives the charge reading over time. This server does the Ethernet communication to the Data Acquisition system DAQ [7] and to the displays as well.

ØМО

In the former VME based systems, a similar architecture as above described is implemented, but it is using shared memory inter process communication (IPC) to pass the raw data from the DMA- to the toroid server. The system is synchronized by the usage of semaphores, which can lead in problematic dead locks.

Using a message passing system like ØMQ [8], this problem is solved by design, as no locking mechanism is required.

ØMQ is an open source socket library with several language bindings for C, C++, Java, .NET, Python and others. The library may carry the messages locally via Inter Process Communication (IPC), via TCP or multicast.

The IPC functionality of ØMQ is integrated into the core server library of DOOCS now and implemented for several single and array like DOOCS data types. The sender process has to provide a data buffer and enable this sender functionality at start-up for the desired data types. The receiver process may connect to this property by using a standard DOOCS address string. A special class is provided to keep this string and attach the callback function. This string may be changed online to reattach to another channel.

It was easy to integrate ØMQ into the DOOCS system; first experience shows, that it runs very stable and after restart of a process in the chain, the reconnects are working fast and reliable.

STATUS

At the moment, several μ TCA crates have been already installed at the FLASH accelerator and are used in standard operation, mainly serving as interface to the old 9MHz FLASH timing system or for slow analog or digital IO read-out. For all these crates, IPMI middle-layer servers are setup for monitoring and control.

One μ TCA crate is permanently attached to the 9MHz timing plus the 108MHz external clock to allow parasitic operation and development of the toroid system with two toroid signals connected. On this computer, the above described software architecture has been installed. Some more timing information need to be passed through the messaging chain to adapt the readout to changes of the accelerator settings automatically.



Figure 1: Toroid data display with 2 bunch, 0.35nC, $1\mu s$ repetition rate.

The software runs stable and the performance seems sufficient, even though a full equipped crate has not been

tested yet. The latency of the data flow from the driver through all three servers to the toroid server was measured and is around 3ms as shown in Figure 1.

Installation of the hardware and software will start in spring 2013 for FLASH2 and for the XFEL one year later.

SUMMARY

The XFEL fast diagnostics and controls will be based on μ TCA front-end systems. The required hardware has been designed and tested; the mass production and the purchasing of boards will start soon. The software architecture has been proven to work with toroid readout and will be adapted now for other systems.

- [1] http://xfel.eu
- [2] http://flash.desy.de
- [3] http://www.picmg.org
- [4] http://doocs.desy.de
- [5] JDDD": A Java DOOCS Data Display for the XFEL E.Sombrowski, K.Rehlich, ICAEPCS 2007
- [6] http://tine.desy.de
- [7] Evolution of the FLASH DAQ System, V.Rybnikov et al, ICAEPCS 2009.
- [8] http://www.zeromq.org

PC BASED REAL TIME DATA EXCHANGE ON 10GbE OPTICAL **NETWORK USING RTOS ***

R.P. Gupta, Green Systems, Ahmedabad, India H. Dave, IPR, Gandhinagar, India

Abstract

The traditional embedded systems are expensive to adapt to the new requirements. The Personal Computer based systems offer alternatives for industrial controls. It reduces the capital cost and provides a solution for multiple applications. However, limitations of PC based controls should be resolved. PC operates on a non-real time OS with non-deterministic response to real time events and data. The real-time pre-emptive kernel for Linux uses Xenomai for better solution. A real-time 10GbE data exchange optical network using Xenomai extension for Linux is demonstrated. The hardware based on Intel 82599 10GbE Ethernet PCIe network card supports IEEE1588 standard for synchronization. deterministic response to real-time interrupts and events.

The benchmark testing comprises nodes and data sources, for data exchange among nodes, which would improve the performance of PC control systems. Data sources and consumers include time synchronization, hardware and software events broadcasting. A single network cable is used for exchange of status and control data among nodes. Moreover, the open source Ubuntu Linux RTOS will help the future development.

INTRODUCTION

Industrial control systems are evolving from dedicated systems to generalized and flexible systems. Traditional controls were designed around dedicated embedded system, wherein the design was applicable for a given set of control parameter and process. These systems would need major changes for new set of control parameters, computations and processes. The personal computer based controls offer better computational power, flexible and faster systems, via Ethernet [1]. The complex scientific experiments, [2] require more computational power and better controls [3]. Multiple applications and wider communication support on PC could offer costeffective solution for real-time network.

Implementation of PC based industrial control system encounters many limitations [1, 2]. PCs were basically designed for applications with non-real time responses. Operating Systems (OS) for PC do not exhibit deterministic response to real time events and data [4]. Most of the common OS are not pre-emptive. However, as per [4], Xenomai kernel for Linux OS is a better preemptive kernel, which provides deterministic real time response. Xenomai is an alternative to the proprietary real time operating system, because it extends GNU/Linux \gtrsim with real time performance [5].

* Work is supported by research grant No. BRFST/NFP/2012/Feb/N/05; under the National Fusion Programme of the BRFST at IPR, Ahmedabad

The pre-emptive kernel needs a better support from the system hardware, so that hard-real time interrupts are serviced within a given time-frame. PC based controllers require networked nodes for data acquisition, event monitoring and control.

With the advent of 10GbE Ethernet controller cards, it is now feasible to setup a multi-node network with reasonable throughput and latency performances [6]. Ethernet controllers such as 82599 and X540-T2 are now available from Intel, with Linux drivers. Hard-real time network are possible with Ethernet controllers [6], due to faster PCI express interface. The fast PCIe 2.0 interface on a PC ensures faster response from host CPU. The 10GBASE-T converged network adapter from Intel such as X540-T2, offers advantages of low latency, load balancing on multiple CPUs, MSI-X support and flexible I/O virtualization.

The real-time data exchange needs reliable time synchronization. In particular, a hard real-time network needs synchronized time stamping on data packets. Intel's controller support IEEE1588 Precision Time Protocol (PTP). Time synchronization is achieved with accurate time reference from GPS connectivity.

According to the above facts, network adapters for optical cable (X-520-SR2), as well as for Cat6A copper cable (X540-T2) were used to create 10GbE network, running on Linux with Xenomai real time framework.

An experimental real-time network is demonstrated for benchmark testing of 10 GbE network using Linux RTOS. Standard network performance test utilities such as Netperf, Iperf and Jperf were used for the benchmarking. Tests were performed both for TCP and UDP data exchange.

REAL TIME DATA EXCHANGE

The real time data exchange require pre-emptive kernel with deterministic response time. There are two open source real time extensions for Linux namely RTAI and Xenomai. However, Xenomai is being updated by the open source community. The comparative study [2] of VxWorks, RTAI and Xenomai indicates that performance of Xenomai is acceptable for most applications. It is necessary to apply ADEOS patch for Xenomai. We have used standard open source patches available for Xenomai kernel. For the present testing, the interrupts priority was ensured with minimum external devices connected to the CPU. The main objective is the basic setup of real time network using 10GbE network card, wherein the PC is used exclusively for the benchmark testing.

Network Setup

The 10GbE network was setup around a PC with Intel Ethernet network adapters. For hard-real time network a pre-emptive kernel is required with reasonably fast Ethernet controller adapters. Intel provides new range of 10GbE Ethernet controllers with virtualization features as well. Linux drivers are available for X520 and X540 series of network controllers; hence we selected these Ethernet cards with dual adapters.

The PC based real time data network comprises of the following setup:

- A test node PC (named as NewI5) with Intel Core i5-CPU @3.10GHzx4, Memory 3.6GB, with PCIe 2.0 x8 host interface. Installed with Ethernet controller card: Intel X540-T2 10GbE dual adapter with RJ45x2 for Cat6A sealed cable. The Cat6A cable provides better performance for 10Gbps network.
- A test node PC with Intel® Core i5- CPU @ 3.10 GHz x 4; Memory: 3.8 GiB with PCIe 2.0 x8 host interface. Installed with Ethernet controller card: Intel X-520-SR2; Intel 82599 based 10GbE dual port optical network adapter; Optical cable is LC/LC multimode fibre.

The 10GbE network was setup around standard PC running on Ubuntu Linux OS. Dual adapters were loop backed with switchless network, for testing the maximum possible performance. Following OS and kernels were used to create Linux kernel with Xenomai extension.

• OS: Ubuntu 12.04, Vanilla Linux kernel version 3.2.21;

• Adeos patch 3.2.21 -x86-1.patch was used with Xenomai 2.6.1.

It created an Ubuntu bootable version with Xenomai extension. The stable version 2.6.1 of Xenomai has been released recently on July 10, 2012, with many bug fixes. The Xenomai related tests such as Xeno-test etc., were successfully confirmed on both the nodes.

Network Performance Benchmark Tools

Netowrk performance is measured by utility software that measures TCP/IP and UDP performance at the application or protocol level. The popular and proven benchmark tools are mostly used for reliable comparison of performance results. The benchmarking tool should test throughput and RTT (Round trip time) latency of the network.

Standard open source tools such as Netspec, Netperf, Iperf, Jperf, Sockperf etc., were evaluated. We finally decided on Netperf, Iperf and Jperf, having popular support, for all measurements. Netperf measures throughput and also RTT latency. Iperf with Jperf provides graphical results with jitter information for UDP protocol. All the selected tools work under Server/client communication system with command line interface. The Netperf tools also provide CPU utilization for useful indicator of the network performance. The results are presented in graphical format by Jperf. We have selected TCP and UDP protocols for comprehensive evaluation of the network.

BENCHMARK TEST RESULTS

For performance measurement, the proposed network was tested with Ethernet as the only active port. USB port and all other ports were inactive, so that CPU is mainly utilized for Ethernet activity. The aim of the test was to determine the upper limit performance of the network under ideal test conditions...

Netperf Test

The measurement command and result for Netperf tool is shown in Table 1. The server / client setup provides result for UDP protocol on client node. The Trans rate measures the latency time, without lost datagram for UDP.

Table 1: Netperf results for UDP protocol

NEWI5:~# netserver -L 192.168.10.1 -p 5679			
Starting netserver with host '192.168.10.1' port '5679' and family AF_UNSPEC			
NEWI5:~\$ netperf -t U 192.168.10.1 -p 5679 MIGRATED UDP REQU 192.168.10.2 () port 0 AF_ AF_INET : first burst 0	UDP_RR -L 192.168.10.2 -H EST/RESPONSE TEST from INET to 192.168.10.1 () port 0		
Netperf results for UDP protocol	163840		
Netperf results for UDP protocol	163840		
Request Size bytes	1		
Resp. Size bytes	1		
Elapsed Time(s)	10.01		
Trans. Rate/ s	102245.9		

The measurement result for TCP protocol is shown in Table 2, wherein it indicates that transfer rate is 82986 per sec, about 19% lower than UDP. The ping result is shown

Table 2: Command for TCP protocol

NEWI5:~# netserver -L 192 Starting netserver with host and family AF_UNSPEC NEWI5:~\$ netperf -t TC: 192.168.10.1 -p 5679 MIGRATED TCP REQU from 192.168.10.2 () port 192.168.10.1 () port 0 AF	2.168.10.1 -p 5679 '192.168.10.1' port '5679' P_RR -L 192.168.10.2 -H JEST/RESPONSE TEST : 0 AF_INET to F_INET : first burst 0	uthors
Netperf results for UDP protocol	16384	ctive a
Netperf results for UDP protocol	87380	e respe
Request Size bytes	1	y the
Resp. Size bytes	1	12 b.
Elapsed Time(s)	10.01	201
Trans. Rate/ s	82986.52	nt ©
		Copyrigh

in Table 3, which shows average RTT of 10 us and mdev is 6 us. This fast response is useful for real time applications.

Table 3: RTT, as given by ping command
EWI5:~# ping -I 192.168.10.1 192.168.10.2
4 bytes 192,168,10,2; icmp reg=4 ttl=64 time=0.006 ms

6 packets Tx, 6 received, 0% packet loss, time 5000ms	
rtt min/avg/max/mdev = 0.006/0.010/0.022/0.006 ms	

Jperf/Iperf Test

Measurement results with Iperf and Jperf are shown in the Fig. 1 and Fig. 2. For basic UDP test, the BW is restricted to 100 Mbps, hence data loss is 0%.



ODP Server with juter of 20sec, BW = 100Mops			
			Lost/Total
Transfer	Bandwidth	Jitter	Datagrams
122649			
KBytes	100454 Kbps	0.002 ms	0/85436 (0%)

Figure 1: UDP Server output.

The Table 4 for UDP output with bandwidth of 1Gpbs indicates data loss of 0.78%. With UDP, data loss increases with increase in the required bandwidth.



The benchmark testing of Ethernet network using Linux PCs, with Xenomai framework extension has been performed with 10GbE Ethernet controller cards. The results, suggest the following:

Table 4.	UDP	server	output	with	BW =	1Gh	ns
14016 4.	0D1	201 101	output	with	DW =	100	72

iperf -s -u -P 0 -i 1 -p 5001 -C -f g			
Interval	Bandwidth	Jitter	Lost/Total
			Datagrams
(s)	(Gbps)	(ms)	(%)
0.0-10.0	0.68	0.002	0.78
			(4566/582135)
0.0-10.0	1 datagrams received out-of-order		

The RTT of the order of 6 to 10 uSec is possible. Real time pre-emptive Linux kernel with Xenomai extension using Intel's 10GbE Ethernet controllers can be used for complex applications. TCP protocol offers reasonable performance with higher reliability. UDP protocol may be used for lower bandwidth application, however at higher bandwidth the data loss increases. For multiple node networks, TCP protocol is recommended. A real time network with the order of 10usec round trip response time is possible with pre-emptive Linux system.

Table 5: Netperf command for UDP request response with CPU utilization

NEWI5:~# netserver -L 192.168.10.1 -p 9999			
NEWI5:~\$ netperf -t UDP RR -L 192.168.10.2 -H			
192.168.10.1 -р 9999 -с -С	_		
Socket Send bytes	163840		
Size Recv bytes	163840		
Request Size bytes	1		
Resp. Size bytes	1		
Elapsed Time(s)	10.01		
Trans. Rate/ s	98200.4		
CPU local (% S)	22.72		
CPU remote (% S)	22.72		
S.dem local (us/Tr)	9.255		
S.dem remote(us/Tr)	9.255		

The Table 5 below shows the UDP response request with CPU utilization, wherein CPU utilization is 23% and transaction rate is higher than TCP protocol.

- Kun Ji and Won-jong Kim, "Real-Time Control of Networked Control Systems via Ethernet" International Journal of Control, Automation and Systems, vol. 3, no. 4, pp. 591-600, Dec.2005
- [2] F. Leroux et al., "New Developments on Tore Supra Data Acquisition Units", Proceedings of ICALEPCS, Grenoble, France, pp. 922-925, 2011.
- [3] Ondrej Cevan, "Installation and Implementation of Xenomai on an Industrial ... IO-Card to the Xenomai Infrastructure", Vienna University of Technology TRAINEESHIP PROJECT, pp.1-87, July-August 2006

- [4] A. Barbalace, et al., "Performance Comparison of VxWorks, Linux, RTAI and Xenomai in a Hard Real-Time Application", IEEE TRANS.ON NUCLEAR SCIENCE, VOL.55, NO. 1, FEB.2008
- [5] Byoung Wook Choi, et al., "Real-time control architecture using Xenomai for intelligent service robots in USN environments", Intel Serv Robotics(2009) 2:139-151 DOI 10.1007/s 11370-009-0040-0
- [6] Amarpreet Ugal "Hard Real Time Linux *using Xenomai*" on Intel Processors", Intel Xenomai on i7 core,Oct.2009.

MASTER SLAVE TOPOLOGY BASED, REMOTELY OPERATED, PRECISION X-RAY BEAM PROFILER AND PLACEMENT SYSTEM FOR HIGH PRESSURE PHYSICS EXPERIMENT AT INDUS-2 BEAM LINE

H.S. Vora*, P. Saxena, V. Dubey, I. Singh, C.P. Navathe, A.K. Sinha, A. Upadhyay, M.N. Singh, T. Ganguli, S.K. Deb, RRCAT, Indore, India

C. Narayana, JNCASR, Bangalore, India

Abstract

RRCAT has commissioned a beam-line on Indus-2 synchrotron facility for carrying out Angle Dispersive Xray Diffraction Measurement. A typical high pressure measurement is carried out by placing the sample in the Diamond Anvil Cell (DAC) with the sample located in a region of beam diameter within 50-100 µm. The X-Ray beam has to pass through the DAC to ensure maximum illumination of the sample with the X-Rays. An X-Y beam scanner/locater cum placement system is developed, which scans an area of 10 x 10 mm² with resolution of 10 to 100 µm in rough scan mode and fine scans selected area with programmable resolution of 2.5 to 25 µm. The scanner acts as slave to the PC in which master GUI grabs the data on serial port and plots the image of X-ray beam. It also analyses and detects the coordinate with maximum intensity. Thus the DAC can be placed at the desired location with an accuracy of 2.5µm anywhere within 10 x 10 mm², for performing experiment. Developed system takes only ~5 minutes to search the beam and a few seconds to place DAC at any the desired location within the scanned area.

INTRODUCTION

RRCAT has commissioned beam-line (BL-12) on Indus-2 synchrotron facility to carry out Angle Dispersive X-ray Diffraction (ADXRD) measurements. Since its commissioning, this beam-line has been used by various researchers, however, for its wider utilization; it is upgraded to perform measurements at high pressures using Angle Dispersive geometry. High pressure XRD measurements are extremely important to determine the structural properties of materials under extreme condition and provide information about the bulk modulus, stable phases etc. It is carried out by placing the sample in the Diamond Anvil Cell (DAC) that has a very small sample volume with a cross-sectional circular opening area of ~150 micros. Alignment of the incoming X-Ray beam from the beam-line with sample in DAC to ensure maximum illumination of the sample is thus a challenging task. Further, as the X-Rays are ionizing radiations and The exposure to the user needs to be essentially avoided, the alignment needs to be carried out in a shielded environment using a remote controlled system. To accomplish the above tasks, a master slave topology based remotely operated precision beam locator cum placement system and interactive control software to measure beam size was developed.

The DAC is mounted on a two axes translation stage, with motion in a plane perpendicular to the X-Ray, and an X-ray sensitive detector is placed behind the cell. The software guides the DAC through a matrix of user selected points and the detector, gives the X-Ray intensity through the DAC as a function of its X & Y position. These data are presented in image format, which can be displayed in various shades of color. The optimum position of the DAC is determined as the co-ordinates where the transmission of X-Rays through the DAC is the maximum.

SYSTEM DESIGN PHILOSOPHY

• ADXRD beam line of Indus-2 has been used by various researchers may or may not be software conversant. Keeping this point in mind, GUI based robust software have been designed and developed, which is self-explanatory and has capability to guide even a beginner. To fulfill this, in software design, elaborate error handing has been incorporated.

• The instrument uses indigenous components like XY motorized translation stages manufactured in India and its fast driver and controller have been developed inhouse at RRCAT.

• Keeping current and future software trend in mind it was decided that software must operate on future Windows platform (64 bit) instead of currently used Win XP.

SOFTWARE

The GUI based software is named **Lakshya**, functions as master and is developed using Vb.Net. XY translation stage control and data acquisition unit based on micro controller and its software is developed in assembly language, which functions as a slave unit of **Lakshya**. All the settings of XY translation stage and data acquisition module are done in master's GUI and final user approved settings are only transmitted to slave for execution. Current software uses 10 x 10 mm² area to locate x-ray beam. Keeping future need and different applications in

^{*}vora@rrcat.gov.in

mind software is developed in such a way that higher scan size and data resolution (8 to 16 bits) can be adopted.

Figure 1 shows main window of Main window of Lakshya. The scanned image size depends on selected resolution i.e. Image Width & Height = 10000/Selected resolution.

Software salient features

- (1) Rough scan: In this mode, an area of $10 \times 10 \text{ mm}^2$ with selectable resolution of 10 to $100 \text{ }\mu\text{m}$ is carried out. This provides the tentative X-ray beam location in 2-D format. It takes ~ only 5 to 6 minutes for a rough scan at a scanning resolution of 50 μm .
- (2) Fine scan: The area is selected from rough scanned image with programmable resolution from 2.5 to 10 μ m in steps of 2.5 μ m. This provides accurate 2-D map of X-ray beam. Beam size can be measured manually selecting coordinates or automatically. It also provides accurate beam profile.
- (3) The DAC / detector can be placed on desired position within accuracy of $\geq 2.5 \ \mu m$. The position point is picked by a mouse click from scanned x-ray beam image.
- (4) After placement at the desired location the Si photodiode detector provides the beam intensity at set position which provides long term stability of the beam.
- (5) Programmable S/N enhancement facility has been incorporated to seek weak signal.

EMBEDDED TRILOCHAAN IN LAKSHYA

Lakshya has been integrated with existing Trilochan software for image processing and measurement facilities. It directly provides 2-D Intensity profile and 3-D view in different viewing angles also directly measures X-ray beam size. To reduce noise, facility of programmable low pass filter and median filtering can be used which are shown in the Figure 2.

SLAVE UNIT HARDWARE

The scanner is controlled by an indigenously developed 80C552 microcontroller based CPU card having internal

10-bit ADC, ports for motor control, front panel display and keypad connections. Scanning is performed at 2.5 μ / step in both X and Y direction. To reduce the overall scan time, high speed stepper motors are used, which are controlled by an indigenously developed universal control cum driver card interfaced to CPU. Provision of optical limit switches is made for safety of the system in both directions.

Further, to improve reliability of data transfer on serial link to avoid EMI, an optical fiber based serial interface have been incorporated which provides reliable data transfer 57.6 kbps. Pre-amplifier for detector head (Si photodiode AXUV-100) uses the current to voltage converter having trans-impedance of 1 G Ω and dynamic range of 120 dB. Input pins are guarded to avoid the leakage. Precision zeroing circuit is also added to compensate for background signal from detector. The unit has provision for digitally programmable gain selection of 1, 10, 100 and 1000 from master PC and an averaging based selective signal to noise ratio enhancement facility has been incorporated to capture weak signal.

PERFORMANCE AND RESULTS

The data was taken, at 17 keV photon energy, on the image plate with an acquisition time of about 20 minutes. The samples were placed inside a DAC with a gasket diameter of about 150 microns. No significant gasket peaks are observed because the precision x-y scanner stage ensured that the X-Ray beam passes exactly through the centre of the DAC. Figure 3 shows the diffraction patterns of LaB₆ as a function of pressure. The absence of the gasket peaks and the shift of the Au peaks with pressure are clearly seen. The cubic phase peaks of the LaB₆ are highlighted in the data.

CONCLUSION

We have designed and deployed the Lakshya system for Indus-2, BL-12, RRCAT to precisely align the incoming X-ray beam on the DAC for perform high pressure measurements. Developed system takes only \sim 5 minutes to search the beam and takes only few seconds to place DAC at any the desired location within the scanned area.



Figure 1: Main window of LAKSHYA shows marked rough scan image and fine scan in progress.



Figure 2: Embedded Trilochan in Lakshya for processing and measurements of acquired scanned beam.



Figure 3: X-ray beam 2-D map and data for sample LaB6 at different pressure.

A FLEXIBLE AND TESTABLE SOFTWARE ARCHITECTURE: APPLYING PRESENTER FIRST TO A DEVICE SERVER FOR THE DOOCS ACCELERATOR CONTROL SYSTEM OF THE EUROPEAN XFEL

A. Beckmann, S. Karabekyan, J. Pflüger, European XFEL, Hamburg, Germany

Abstract

Presenter First (PF) uses a variant of Model View Presenter design pattern to add implementation flexibility and to improve testability of complex event-driven applications. It has been introduced in the context of GUI applications, but can easily be adapted to server applications. This paper describes how Presenter First methodology is used to develop a device server for the Programmable Logic Controls (PLC) of the European XFEL undulator systems, which are Windows PCs running PLC software from Beckhoff. The server implements a ZeroMQ message interface to the PLC allowing the DOOCS accelerator control system of the European XFEL to exchange data with the PLC by sending messages over the network. Our challenge is to develop a well-tested device server with a flexible architecture that allows integrating the server into other accelerator control systems like EPICS.

TECHNICAL BACKGROUND

The European X-Ray Free-Electron Laser (XFEL) Facility will use three ~200m long undulator systems to produce laser light by the Self Amplified Spontaneous Emission (SASE) process [1]. The photon energy can be varied by a change of the undulator gap. The corresponding motion control is implemented with Beckhoff TwinCAT PLC software on Windows PCs.



Control Network

Figure 1: Integration of undulator control into DOOCS.

Figure 1 shows how undulator control is integrated into the Distributed Object-Oriented Control System (DOOCS), which is the accelerator control system of the European XFEL facility [2]. The DOOCS Server, running on a Linux host, defines for the undulator system a set of properties, such as the undulator gap. These properties can be read or modified by DOOCS client applications (not shown in the figure) in order to control the undulator system. The DOOCS Server exchanges the property values with the Device Server using the ZeroMQ message transport library [3]. The Device Server runs on a Windows host together with the PLC of the undulator system. Both exchange the property values using the Beckhoff Automation Device Specification (ADS) protocol.

The reason for having two servers is that ADS is supported only for Windows platforms and the DOOCS Server is supported only for Linux platforms. The message interface between both servers crosses this platform border. An additional benefit of the message interface is that it adds flexibility on the machine control side. Anything may connect to the Device Server, regardless of the platform it runs on, as long as it is able to generate appropriate messages. It could for example be an EPICS Channel Access Server (CAS) with only little effort to implement the message interface.

DESCRIPTION OF PRESENTER FIRST

So, how does Presenter First help in developing the Device Server application? Presenter First proposes a pattern for structuring the code in a specific way, and a process for developing the application in a specific sequence [4]. The pattern improves testability of the code and offers some flexibility regarding the interaction with the environment. The process saves effort by developing on the basis of the intended behaviour of the application.

The Pattern

Applications are implemented using a variant of the Model View Presenter (MVP) pattern, as shown in Fig. 2.



Figure 2: MVP variant used in PF.

The presenter represents the behaviour of the application that is defined by the functional requirements, the model manages the application data and logic, and the view interacts with the environment. Model and view do not communicate directly with each other because this would limit flexibility; instead both communicate with the presenter over clearly defined interfaces. They send events to the presenter to trigger some behaviour, which

usually results in method calls back to either model or view.

The use of interfaces is essential for the testability, because it decouples the presenter from the model and view. The presenter depends only on the interfaces and not on the objects that implement the interfaces. Within the test environment the model and view can then be replaced with a mock model and mock view that provide the same interface but behave as required by the test.

Flexibility comes with the separation of code, which interacts with the environment, into the view. This part can be changed without the need to modify either model or presenter. It just has to implement the view interface. Currently, the device server uses ZeroMQ to interact with the network. If this needs to be changed, only the view requires modifications reducing the effort to do so.

The Process

The development of an application starts with the presenter. The functional requirements, which define the application behaviour, are analysed for their impact on the model and view. The result is the definition of events and methods for the model and the view interfaces. Once all requirements have been analysed, the interfaces are completely defined and implementation of model and view may now start.

The advantage of this approach is that the application exactly fulfils the functional requirements: it behaves exactly as intended. Over-engineering by implementing functionality that is not required is avoided, which saves development effort and reduces the complexity of the code.

DEVICE SERVER DEVELOPMENT

The use of Presenter First is described by taking the Device Server as an example. The Device Server is a C# application that runs as a Windows service on the host with the PLC. Unlike regular applications, a Windows service is started by the Windows Service Control Manager (SCM) during boot and stopped during shutdown.

Software Architecture

Figure 3 shows the basic software architecture of the Device Server. It is divided into two large modules: the Windows Service module and the Control module.

Windows Service contains code that registers the Device Server with the SCM. Its only purpose is to start and stop the Control module whenever the Device Server itself is started or stopped by SCM.

Control contains the instances of model, view and presenter, which together implement the applications functionality. Once the Control module is started by the Windows Service module, the presenter processes events from either model or view according to the implemented behaviour. Events from the model are triggered by the PLC, for example when PLC variable values have been

changed. Events from the view are triggered by the ZeroMQ interface for example when a message arrives.



Figure 3: Software architecture of Device Server.

Applying Presenter First

As proposed by Presenter First the development starts with the presenter by analysing the functional requirements of the Device Server. These are:

- 1. The service shall activate or deactivate the message interface, when it is started or stopped by SCM respectively.
- 2. The service shall respond to request messages received at the message interface.
- 3. The service shall send an update message via message interface, whenever values of PLC variables change.

In the following, the analysis is described taking the second requirement as an example. It starts with a request message arriving at the ZeroMQ message interface. A message is in principle a byte array with a certain length. ZeroMO does not care about the content; it just tells the view that a number of bytes have been received. The parsing of the message content is considered as being part of the application logic, so that only the model knows how to do it. Therefore the incoming message needs to be passed to the model. To do so, the view sends an event to the presenter that a request message has been received. The event handler inside the presenter then reads the message bytes from the view, translates them into a more convenient to use message object, and passes it to the model. The model processes the message object and returns another message object as a response. The presenter translates the response message object back into a byte array and passes it to the view. Finally, the view sends a response message on the ZeroMQ message interface.

The analysis results in the following interfaces. The view interface requires:

- an event indicating that a message has been received,
- methods returning the length and bytes of the received message,
- a method to set the length and bytes of the response message

The model interface requires:

• a method to process the received message, returning a response message

The remaining task is now to implement the events and methods from these interfaces inside the model and view.

Testing

The Device Server is tested on unit and on system level. Unit level tests verify the functionality of individual software units, such as the presenter. System level tests verify the integration of the complete server into the environment, which consists of the Windows SCM, the DOOCS Server and the PLC.

Some parts of the Device Server cannot be tested on unit level, since they interact with the environment that cannot be generated during an automated unit test run. These are:

- the Windows Service module
- the view
- the part of the model that communicates with the PLC via TwinCAT ADS Communication Library

To reduce the probability of errors in these parts, their code is as simple as possible, which means that there is almost no processing inside these parts. This also reduces the effort to test these parts manually on system level. The Windows Service module is very simple since it just passes down start and stop from the SCM to the Control module. The view is a little bit more complex since it contains some inevitable processing related to the use of ZeroMQ. Finally, the ADS library, which is used by the model to communicate with the PLC, is wrapped with an ADS interface adapter. Instead of calling methods directly from the ADS library, the model calls methods from the interface adapter. Much like the interface between presenter and model, this decouples the model from the ADS library and increases its testability. The adapter itself is very simple, since the interface methods simply call the corresponding ADS library methods.

The presenter is tested on unit level by replacing model and view with mocks that implement the same model and view interfaces. Mocks are specialized test case objects, which in principle track calls to the interface methods. At the end of a test, the mocks verify that the methods have been called the correct number of times. They may also verify that the correct arguments have been passed, for example that a method has been called once with an integer argument equal to 1. For more complex test cases, mocks can be set up to return values, or even to raise program exceptions. Tracking method calls with mocks is known as behavioural verification, in contrast to the classical state verification, where data values (or state of data) are checked after some part of the code has been executed [5]. It is therefore ideally suited to complement Presenter First with its focus on the behaviour of an application.

A typical presenter test case creates instances of the presenter, the mock model and the mock view. The mocks themselves are taken from the mock object library Moq [6]. The test case then configures the mocks and starts the

test by triggering one of the events from either the model or view interface. At the end of the test case the mocks perform their verification by checking whether methods have been called the correct number of times and with the correct arguments.

The model is tested on unit level much like the presenter by replacing the ADS interface adapter with a mock adapter. However, test cases start with a call of a method from the model interface instead of triggering events. There is a test case for each model interface method, so that the model is almost completely tested except for the ADS interface adapter at the boundary that communicates with the PLC, as mentioned earlier.

Finally, the Device Server is tested on system level. It is installed as a Windows service on the host with the PLC. Then, the DOOCS Server and PLC are used to stimulate the Device Server. Afterwards the state of the DOOCS Server and PLC are verified manually.

Experience

The use of Presenter First allows testing most of the functionality already on unit level. This notably reduces the probability of errors detected on system level. The code is cleanly structured, which simplifies debugging and refactoring. The developped Device Server runs stably over a longer period in a test setup with two undulators controlled by one DOOCS server.

SUMMARY

Presenter First has been used to develop a well-tested Device Server for the undulator control system of the European XFEL facility. The software architecture provides flexibility regarding integration into the machine control system.

The Presenter First software architecture increases the testability of applications with event-based behaviour, which is important for mission critical software such as the undulator motion control Device Server. Development is guided by the behavioural requirements and results in compact code that does exactly what is required and nothing else.

Flexibility comes with the separation of interface and application logic. The interface can be changed without the need to adapt the application logic.

- M. Altarelli et al. (ed), "The European X-ray free-electron laser Technical Design Report", DESY Report 2006-097, 2006.
- [2] K. Rehlich et al., "The Accelerator Control Systems at DESY", in ICFA Beam Dynamics Newsletter No. 47, pp. 139-166, 2008.
- [3] http://www.zeromq.org/
- [4] M. Alles, D. Crosby, C. Erickson, B. Harleton, M. Marsiglia, G. Pattison, C. Stienstra, "Presenter First: Organizing Complex GUI Applications for Test-Driven Development," Agile 2006, Minneapolis, July 2006.
- [5] http://martinfowler.com/articles/mocksArentStubs.html
- [6] http://code.google.com/p/moq/

DESIGN, DEVELOPMENT AND ANALYSIS OF A COMPREHENSIVE OPEN SOURCE SYSTEM FOR PROACTIVE MANAGEMENT OF SECURITY ASPECTS OF A CONTROL NETWORK

S.S. Tomar[#], S.N. Chaudhari, H.S. Chouhan, V.K. Maurya, A. Rawat, RRCAT, Indore, India

Abstract

Control networks can only be assumed to be secure, when they work in complete isolation and all communication ports of the constituent control devices are disabled and are closely monitored for security breaches on 24X7 basis. With more and more control systems being developed using Common Out of The Shelf (COTS) computers, using windows OS, the chances of virus attacks on such control networks is extremely large. Handling zero day virus attacks or virus attacks with unknown cure, is a serious challenge for control network administrators. Another important aspect, somehow related to the security of the control network, is the rising temperatures of the control devices, because of 24X7 operation. All this is difficult to handle manually or using disconnected systems and hence there is a requirement of a comprehensive system which can do all this automatically. In this paper we will discuss the various security related parameters of the control networks and then present a simplified design followed by development details of a comprehensive open source system for proactive management of the security aspects of the control network.

INTRODUCTION

Modern day control networks are large and connected to Internet. They have distributed architecture and contain control, information & resource sharing related network components. COTS computers with Windows Operating System (OS) are used widely in large control networks with distributed architectures. Generally Windows OS based systems are prone to malware attacks. Due to this and the proximity of Internet to such networks, the security issues in such networks are numerous. Managing all these issues proactively and cost effectively is a huge challenge [1], but solutions in the form of numerous Free Open Source Software (FOSS) tools exist.

Copyright © 2012 by the respective authors

Out of numerous challenges in securing advanced distributed architecture [2], [3] control networks, proactive management of a) zero day malware attack and b) the overheating of control network components in distributed networks are important. The security concern about malware infected PCs on a network, is very serious, in control networks, since it sometimes causes transfer of control of the systems to some third party on Internet. The not so old "stuxnet" [4] and "duqu" menace are striking examples. The second security cum safety concern is the overheating of switching and server components, due to

#tomar@rrcat.gov.in

long continuous operations, in large distributed control networks. The situation caused by non working network communication channels and servers, due to overheating, in a control network can cause serious operational issues.

In this paper, we carry out risk analysis of a modern day Distributed Architecture Control Network (DACN) and then identify the areas for proactive management of its security aspects. We present a comprehensive approach to implement a secure DACN [5] using the various FOSS tools. This is followed by the design and development details of our in-house solutions, conceived, for addressing the above mentioned security concerns [6]. The detailed security analysis of the proposed comprehensive approach has been carried out and presented in the paper.



Figure 1: Distributed Architecture Control Network.

TYPICAL DACN

Figure 1 depicts the block diagram of a typical DACN which has the following salient features:

- Uses COTS PC, with windows OS, for computation and decision making.
- Uses Ethernet Technology with wide usage of Quality of Service (QoS) feature for real time communication with other systems.
- Uses modern communication servers like name, email and web for publishing information, thus is connected to Internet in some form or the other.
- Has some data acquisition and control hardware.
- Has some alarm generation system in place.
- Has a storage system for storing events.
- Has redundancy for Computation, Decision making, Communication and Data acquisition & Control.
SECURITY ASPECTS OF DACN

The first step in managing the security aspects of any network [7] is the detailed risk analysis [8] of the setup. A detailed study of the DACN highlights the following risks factors:

- COTS PC with Windows OS, has risk of malwares changing the functionality of the system and in some cases transferring control of the system to third party.
- Ethernet technology based networks, spread over large campuses, have associated risks to Confidentiality, Integrity and Availability (CIA) of the data flowing through it.
- Mail/web/name servers have numerous risks in the form of malwares changing the functionality and data of these systems, SPAMs flooding email accounts and Domain Name System (DNS) poisoning, thereby causing Denial of Service (DoS) to legitimate users.
- Internet connection on a DACN poses all the risks towards CIA of information in the DACN.
- Distributed locations of the DACN components, have the associated risks of environmental changes getting unnoticed, thus leading to overheating and malfunctioning of systems.

Based on the risk analysis, we identify the areas for proactive management of the security aspects of a DACN connected to public network.

- Authentication Authorization Accounting (AAA)
- Firewall and DeMilitarized Zone (DMZ).
- Network Access Control (NAC).
- Network traffic encryption.
- Network traffic monitoring.
- Virus/SPAM management.
- Network fabric (Servers/Routers/Switches/PC /Software) management.
- Log monitoring and analysis.
- Alarm communication management.

SECURE DACN USING FOSS TOOLS

As with any network, the only design goal of a secure DACN is to make it secure by ensuring the CIA of information across various systems. Figure 2 depicts a very basic design of the secure DACN. The salient features of the design are as follows:

- A multi layered DMZ approach has been followed to restrict the access of servers over public networks, depending on the requirements.
- A proxy server based Internet access approach has been followed for internal users instead of the common direct Internet or Network Address Translation (NAT) approach to govern the web traffic flowing in/out of the DACN. This helps in providing access control and auditing every access to those networks.
- Filtering of web traffic for virus and SPAM has been done at every port of entry and exit of the traffic to/from public network.
- AAA of every access to the network has been incorporated in the design.
- Visibility of the resources, over public network has been reduced by using the perimeter level firewall in the design.
- Network traffic in the DACN has been encrypted at the network layer level.
- Network monitoring systems have been incorporated to monitor the traffic entering in/out of every DMZ, thus taking care of unusual/ unwanted traffic.
- Automated log analysis systems have been incorporated.
- Abnormal network event related real time multimode alarm generation & communication system has been incorporated.

To make the system cost effective, development of the system has been done using the below mentioned FOSS tools and Linux as OS.



Figure 2: Securely designed modern day Distributed Architecture Control Network.

- AAA CentOS Directory Server and FreeRadius.
- Firewall and DMZ IPtables with Bastille and firewall builder is used to build and manage the IPtable rules.
- Network Access Control PacketFence and Authenticated Squid Proxy server.
- Network traffic encryption OpenVPN and Apache with secure hypertext transfer protocol.
- Network traffic monitoring Traffic Capturing using mirroring and Netflow tools like ipcad on OpenBSD OS and Open Source Security Information Management (OSSIM) with nagios/snort, NfSen plugins.
- Virus/SPAM mgmt. Clam AntiVirus (ClamAV) for virus filtering, SpamAssassin for SPAM control on servers. ClamAV and Microsoft Security Essentials on user end PCs.
- Network fabric management OSSIM to monitor host and service availability. Bastille for server hardening and NetDisco for asset discovery.
- Log Analysis Webalyzer for web logs and OSSEC with automated log analysis feature.
- Alarm communication system Asterisk for communicating alarms using the telecommunication network, emails using email network, web server for status information display in the form of web pages.

In the following two subsections we discuss techniques to further enhance the security of this basic secure DACN by adding our in-house conceived, designed and developed systems.

a) Malware Detection System

In networks using squid proxy for access of Internet/Intranet, squid log analysis technique can be used to detect malware infected PCs on the network. The idea is that since the malwares like to proliferate & use Internet/network as the channel for proliferation hence on networks using proxy services, footprint of malwares are present in the proxy logs. We have designed & developed a complete system of automatically detecting suspected malware infected systems, blocking their access to Internet, informing the concerned users about the blocking and providing the user with a mechanism to unblock the PC for future Internet access after removal of the malware. Figure 3 illustrates the concept of the complete solution which consists of following subsystems.

- Malware infected PC detecting and blocking subsystem.
- User PC status checking subsystem.
- User controlled PC unblocking subsystem.

First of all the entire network is configured to use a squid proxy server for access to outside network/Internet.

The logging and the authentication options are enabled in the squid configuration file. Whenever a



Figure 3: Conceptual block diagram of the zero day Malware detection/blocking/unblocking solution.

malware infected PC tries to access Internet, unauthenticated proxy requests are generated, which are marked with TCP_DENIED tags in the logs. The same tag is also generated for genuine unauthenticated requests of the users who may sometimes mistakenly give wrong passwords. But the difference in both the cases is that a genuine user may fail to give wrong password a limited/countable number of times only, while a malware infected PC tries and retries multiple times. This characteristic of the malware infected PC generating excessive TCP_DENIED log lines is exploited in this system to detect them. The routing table modification with a non routable entry for an IP is used to block the access of a particular PC/IP of the squid proxy server.

Malware infected PC detecting and blocking subsystem consists of a script, in the squid proxy server, with the following algorithm:

- Step 1. Define a threshold limit for generated TCP_DENIED requests for a PC/IP.
- Step 2. Read the squid access log file for lines containing TCP_DENIED word. Only consider the lines that have got augmented since the last read.
- Step 3. Read each line containing TCP_DENIED tag and count the number of such lines generated for each PC/IP.
- Step 4. Compare the count of such lines with the threshold rate limit as defined in step 1.
- Step 5. If the count of such lines generated by a PC/IP is more that the set threshold limit then mark the PC/IP as malware infected.
- Step 6. For IPs listed in step 5, make a non routable entry in the routing table of the proxy server for each of them for blocking their access.

The design of the "user status checking subsystem" includes a portal with necessary web pages to publish the status of the user PC/IP. This subsystem is residing on another server, configured with a portal/web server, which is accessible to the PCs even when proxy access is

not present. A script with the following algorithm is used to provide the required functionality:

- Step 1. Get the IP details of the PC from where the status check function is being performed.
- Step 2. Read the routing table of the proxy server which was modified by the detecting/blocking subsystem.
- Step 3. Check for the routing table entry related to the PC/IP from where the status is being checked.
- Step 4. If routing table entry for the IP exists then generate a web page showing Blocked = true as the PC status else generate a page with Blocked=false message

The design of the "User controlled PC unblocking subsystem" includes, extending the capabilities of the portal setup for the second subsystem with a script with the following algorithm:

- Step 1. Get the IP of the PC from where the user requested for unblocking.
- Step 2. Read the routing table of the proxy server.
- Step 3. If the IP related routing entry is present in the proxy server's routing table then delete it.

b) Distributed Component Overheating Management System (DCOMS)

To prevent overheating of DACN components located at distant locations, this system has been designed and developed. It uses the Simple Network Management Protocol (SNMP) feature of the managed network switches. The idea is to continuously poll the switch for the value of the SNMP temperature variable and perform actions like generation of email alert and phone calls to the concerned administrators or shutting down of servers. Figure 4 illustrates the complete concept of the developed system.



Figure 4: Conceptual block diagram of the DCOMS.

The system design incorporates the following features and algorithms:

Step 1. Authorized persons can register network switches and servers to be monitored using the

application/web interfaces. This information is stored in XML format.

- Step 2. A program with the following algorithm executes in the DCOMS server:
 - a. Read the database of registered switches and get IPs of the registered switches.
 - b. For each switch IP.
 - i. Connect to switch using the SNMP client and get SNMP variable value of the temperature variable.
 - ii. Read the email-ids and phone numbers associated with the switches.
 - iii. Read the shutdown threshold value of the switch
 - iv. Read the registered server IPs associated with the switch
 - v. If the current temperature exceeds switch overheating threshold then
 - 1. Send email to the concerned persons using the email server
 - 2. Generate a phone call to the concerned persons using the asterisk server

The design of the system for incorporating functionality of automatic server shutdown is:

Step 1. On the server (For registration, one time)

- a. Execute a program/script (register.sh) with the following algorithm:
 - i. Create a restricted shell user in the server and copy the root public key of the DCOMS server to its authorized users list.
 - ii. Install a script (shutdown_ server.sh) with the following algorithm:
 - 1. Read the message file as sent by DCOMS server.
 - 2. If the value of shutdown flag is true then initiate shutdown.
- b. Make an entry in cron scheduler to schedule the program for execution in every minute interval
- c. Make an entry in rc.local file to reset the shutdown flag value.
- d. Copy the DCOMS server root public key into the .ssh folder of the restricted shell user.
- Step 2. On the DCOMS server (For every switch)
 - a. If the current temperature of the associated switch exceeds server shutdown threshold then:
 - i. Send shutdown message to the servers
 - ii. Send email to the concerned persons using the email server
 - iii. Generate a phone call to the concerned persons using the asterisk server.

Step 3. On the server (For deregistration, one time)

- a. Delete the cron entry related to shutdown server.sh script
- b. Delete the rc.local entry for changing the shutdown flag.
- c. Delete the shutdown_server.sh script.
- d. Delete the restricted shell user.

For the development of malware detection system \approx and DCOMS, Linux (CentOS 5.7) as OS, Apache as web \odot server, JAVA for developing the switch threshold \equiv

configurator, HTML and JavaScript for developing web interfaces, XML for storing information, CRON for scheduling, PHP, BASH scripts, *scp* for message passing to the remote hosts, SNMP for polling switches current temperature, Asterisk as telephony system and Qmail as mail server has been used.

SECURITY ANALYSIS

The analysis of the developed, secure DACN, is carried out with respect to the basic defining factors of security – CIA triad- of information in any network.

Confidentiality:

The AAA setup ensures that any flow of message on the network can be initiated only after proper authentication and authorization check. The Accounting feature of both the Directory Server and the RADIUS server provide the necessary audit trails of the accesses. For meeting the requirement of keeping the messages flowing in the DACN, between the various systems highly confidential, traffic encryption at network level is done by using the OpenVPN infrastructure. The use of authenticated squid proxy, enabled for logging, helps in auditing all the accesses made. The use of firewalls, DMZ, NAC and NAT feature provides access control and helps to meet the authorization objective.

Integrity:

Use of dual authentication scheme for authentication - as provided by OpenVPN and AAA infrastructure - by using digital certificates and passwords ensures integrity of the data flowing in the DACN. The AAA infrastructure provides all controls like password expiry, strong passwords, MD5 encrypted storage of passwords etc., to comply to ISO 27001 certification requirements. The use of antivirus software at every entry and exit point of the network ensures that the information published is unaltered because of malware attacks. The use of OSSEC, ensures continuous monitoring of the integrity of various important system configuration files. The use of Hash Message Authentication Code (HMAC) feature of OpenVPN also provides integrity checks of packets at network layer level.

Availability:

The malware detection system increases the availability of the DACN resources by reducing the chances of DoS. The DCOMS ensures continuous availability of DACN with healthy components. The use of network monitoring tools like OSSIM, NfSen, Nagios, Snort ensures that the resources are monitored continuously for failures and security compromising events. The use of OSSEC enhances availability by performing automated log analysis and generating alarms for unusual events. The use of Asterisk, Email and Web server provides functionality of communicating alarms using telecommunication network, email network and web respectively.

CONCLUSION

In a large distributed architecture network, foolproof security is impossible to achieve, thus its security aspects have to be managed. Modern day DACNs have more security requirements as compared to the resource & information sharing networks. In this paper we identified areas for proactive management of security aspects of the DACN. The basic secure DACN can be built using FOSS tools, which are freely available. The paper here presented a working set of FOSS tools available to develop a basic secure DACN.

In this paper we presented our approach to tackle two challenges existing in distributed architecture networks. The first one, of zero day malware attack, has been tackled by using the squid proxy server and log analysis automation technique. The entire process of log analysis automation, to generate alarms in case of malware detection has been presented. The second problem related to continuous monitoring of the distributed set of components for temperature changes have been handled using the available FOSS tools. The technique presented here provides a complete solution to proactively manage the distributed components for temperature variations at distributed locations.

REFERENCES

- [1] Martin Naedele, "Addressing IT Security for Critical Control Systems," Proceedings of the 40th Annual Hawaii International Conference on System Sciences, 07.
- [2] Falk, R., "Industrial Sensor Network Security Architecture," Fourth International Conference on Emerging Security Information Systems and Technologies (SECURWARE), 2010.
- [3] "Architecture for SCADA Architecture for Secure SCADA and Distributed Control Systems Networks," 2010, http://www.juniper.net/us/en/local/pdf/ whitepapers/2000276-en.pdf
- [4] Karnouskos, S., "Stuxnet worm impact on industrial cyberphysical system security," IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society, 7-10 Nov. 2011.
- [5] Zhihu Wang, "Design and realization of computer network security perception control system," IEEE 3rd International Conference on Communication Software and Networks (ICCSN), 27-29 May, 11, Nanning, China.
- [6] Sopko, M. Winegardner, K., "Process control network security concerns and remedies," 2007 IEEE Cement Industry Technical Conference.
- [7] Richard Kissel, Kevin Stine, Matthew Scholl, Hart Rossman, Jim Fahlsing, Jessica Gulick, NIST SP 800-64, "Security Considerations in the System Development Life Cycle," Revision 2, 2008, http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP800-64-Revision2.pdf
- [8] Henryka Jormakka, Pekka Koponen, heimo Pentikainen, Henna Bartoszewicz Burczy, "Control systems of critical infrastructures, security analysis," 2009, http://elektroenergetyka.pl/upload/file/2009/4/elektroenerg etyka_nr_09_04_3.pdf

WHAT IT TAKES TO MAKE A SYSTEM RELIABLE

M. Clausen, M. Möller, S. Rettig-Labusga, B. Schoeneburg, DESY, Hamburg, Germany

Abstract

What is a reliable system and how is reliability defined? This depends on the actual situation and in which environment the system is operated. If you can rely on a scheduled downtime of the controlled system every week, reliability is defined in hours or weeks. In this case the system must run just longer than the scheduled downtime. If the system has to continuously operate for months and even years, your requirements are rising. In cases where continuous operations must be guaranteed even during software or hardware updates, redundant systems come into play. The hardware selection process is driven by basic requirements like 'no moving parts' or 'redundant power supplies'. This implies the selection of possible (fan-less) CPU boards with passive cooling. It also implies no hard discs and reduces therefore the selection of possible operating systems. Continuous operation during updates requires redundant controllers/ CPUs also in addition to redundant power supplies. The latter has a lot of impact on the software running inside the controllers. We will describe the selection process of the components we have chosen and summarize our experience of several years of operations.

REQUIREMENTS

The requirements for process controllers in cryogenic systems are extremely high. Cryogenic systems have to run 24/7 in periods lasting for a year or more. A downtime of the control system will typically cause a downtime of the cryogenic plant of two to four hours to recover cryogenic operations. This does not take into account the systems being fed with liquid helium or the accelerator operations depending on the helium supply.

For the new European XFEL the situation will be even more extreme. The cold compressors of the cryogenic plant are extremely sensitive to distortions of any kind. Any unstable environment will cause an immediate shutdown of the cold compressors. The recovery time is expected to be between twelve and twenty hours at least. Stable process controls have an even higher priority.

POSSIBLE APPROACHES

How can these necessary uptimes be achieved? At DESY we have gained a lot of experience with commercial process control systems. They have the advantage to be supported by a company and should expectedly be reliable. The disadvantage being that they provide a closed environment. No 'special' functions which are necessary for cryogenic controls can be easily integrated. As an example cryogenic temperatures need dedicated calibration curves of a 5th to 7th grade polynomial. There's no easy way to these running in a commercial environment. Another example are

dependencies on the operating system, if not on the frontend controllers then at least on the operator screens. Security updates on the Windows operating system may cause problems in the operator's programs or the commercial software may define the time of system updates or hardware changes while the rest of the Window systems run on a different upgrade pace.

Using PLCs

PLCs may be a solution to reach stability. Predictive continuous processing speeds for all processing blocks in a PLC ensure stability in this respect. Thus PLCs are ideal candidates for machine protection systems. Full process controls on PLCs have certain limitations in processing functionality (see temperature calibration) and transparent data exchange with the operator interface. By default the PLCs do not support transparent data exchange for each and every property of the processing blocks. Each possible connection must be explicitly configured.

Redundancy in PLC controls is partly available but the selection of possible candidates is limited.

Using embedded (real-time) Systems

There is a lot of experience at DESY using embedded real-time systems for process controls. We are using vxWorks on the front end processors and EPICS as the process control system. This combination is typically used for machine controls. At DESY we added and/ or improved the processing block in EPICS to make them process control ready. EPICS comes with transparent access to all properties of the processing blocks. No specific configuration is necessary to gain access to these properties. This is a big advantage over PLCs. On the other hand EPICS did not come with redundancy support.

Redundancy

Redundancy is an important feature when 24/7 operations shall be guaranteed over periods of more than a year. Hardware problems on the processor, on I/O boards or on the control system Ethernet may cause system failures. These hardware problems can be overcome by adding a second processor running the same process database and sequence program in parallel. In case of a failure the second processor will take over without shutting down the controlled cryogenic plant. This kind of redundancy support has been added to the EPICS based control system. This kind of setup shall also help during software updates in the future.

PROCESSORS

We started with VME processors when we started using EPICS at DESY. This was the initial platform for EPICS IOCs. The disadvantage of VME is that a complete VME crate is necessary for a single CPU. And this is typical layout of a front-end processor which controls cryogenic components. All of the I/O is connected to field-buses at the beginning CAN and nowadays Profibus (DP). Just one to three field-buses comprise the full I/O system of a major cryogenic plant. A full blown VME system would be overdone. In addition these systems come with active cooling of the power supplies and of the processor boards. This should be avoided. Any moving parts should be removed from a stable long running front-end system.

Compact PCI for Front-End Systems

We have chosen Compact PCI (cPCI) as the backplane bus for the front-end systems. The processor itself should also run without active cooling. It is hard to find low power consumption processors. Low processor speeds are a rare requirement. We have chosen a CPU board from Kontron Company in Germany the CP-305 which comes with enough processing speed and more than enough memory. The power consumption is about 10W total.

No moving parts – no disks

Using vxWorks as the operating system eases the implementation of a diskless front-end. The operating system is downloaded at boot time from the load host. It is memory resident and has only a small footprint. No disk is required to operate such a system.

Power Supplies

Besides the low power processor boards it was difficult to find low power and (of course) redundant power supplies for the cPCI crates. A special design provides good efficiency at low power consumption and can therefore be operated without active cooling.

This way all the necessary components for the cryogenic process controller can be operated fan-less.

LOAD HOST

Because of the diskless character of the front-end controllers it is important to provide a stable and reliable load host. Otherwise the controllers cannot be started since they are diskless. We have installed a redundant system consisting of a two node Sun cluster with a third node as an adjacent test node (Figure 1). Sun clusters are the most reliable cluster setups on the market. Linux clusters cannot compete with them. The most important service on the cluster is the tftp load service. This is also used as a cluster service which causes a failover if it or the cluster node fails.

Disk Storage

Initially the storage was connected directly to the cluster nodes. Nowadays it is more convenient to use network attached storage (NAS). This way the cluster node as well as other computer on the control network can access the disks directly. We are currently using a redundant set of Netapp nodes. For now they reside in the same cabinet but it is planned to separate them to increase the availability in case of a major problem in the



Figure 1: Redundant IOCs connected to a redundant load host.

computer room.

The same kind of separation can also be applied to the Sun cluster. Availability and reliability are trade off of cluster separation. This is still under negotiation.

NETWORK

The controls network is directly connected to the global DESY network. The main difference being that the connection is not implemented on layer two but on layer three the - routing layer. This way any kind of



Figure 2: Redundant Network - Redundant Power (A | B).

management traffic of the routers cannot propagate into the controls network. Any connection between switches or routers in the network is redundant (Figure 2). Spanning tree helps to organize the proper paths. Redundancy is also implemented on the core switch level. Each building hosting control components is connected by a redundant set of core switches. These are configured in a way that any one of these can fail with seamless takeover by the redundant partner.

POWER

Wherever possible components are equipped with redundant power supplies. This applies for front-end controller, major network components, the Sun cluster and the NAS disks. If redundant power is not feasible then we will make sure that e.g. the network switches for the redundant front-end controllers are connected to different power sources. Power sources are in general battery backup systems which keep up for at least ten minutes. These UPS systems themselves are connected to a reliable main supply. This will be powered by a diesel generator within half a minute after a power fail.

Keeping the control system - including the I/O components - up and running after a power fail has proven to be very useful. This way the operators can still see what's going on in the process when the power is down. Especially when pressure rises or valves are open (which should not) it is useful for the operators to get support from the control system.

SUMMARY

Over the course of more than twenty years we have gained experience how to set up control systems for cryogenic plants. Redundancy is mandatory in many areas. Beginning with redundant UPS systems and ending at the redundant temperature sensor. We even developed a package ourselves to support redundancy in the EPICS toolkit. So far our experience with this setup is very positive.

PLC CONTROLLED SEARCH & SECURE SAFETY INTERLOCK SYSTEM FOR ACCELERATOR

V. Sharma[#], R. Rajan, Seema, S. Acharya, K.C. Mittal, BARC, Mumbai, India

Abstract

A PCVM type 3MeV DC Electron beam accelerator has been developed at Electron Beam Centre, BARC, Mumbai, India. A PLC assisted search and secure system has been designed, developed and installed as the human safety system for radiation protection in the accelerator area.

INTRODUCTION

The electron beam accelerator has various subsystems. All the subsystems have to work together in a pre-defined sequence to generate the desired accelerated electron beam from the accelerator. All the subsystem such as vacuum system, sweep scan magnet unit, high voltage unit, chiller water unit, air circulation unit, search-secure, safety interlock unit, steering/focussing magnet unit has to be controlled. Each group of similar units has been provided with a PLC controller in order to perform the fully automatic operation of that subsystem. Finally all the PLCs are connected together on Modbus TCP-IP network to achieve a single point control of the accelerator. A touch screen panel has been provided at the control panel as an institutive user interface.

SYSTEM OPERATION

The 3 MeV electron beam accelerator is an industrial accelerator developed for the various industrial electron beam processing applications. The accelerator generates high dose X-ray radiation as by product when operational. The Accelerator is placed inside a hall made of thick concrete walls. The accelerator hall is called as the cell area. All the power supplies and other support systems are placed outside the cell area. Cables and electrical connections from the accelerator to the power supplies are made through S bands holes created in the walls of the cell. Since the accelerator generates large amount of radiation, which is actually a required feature for the irradiation but at the same time all the human being working in that region has to be protected against the radiation. The cell area needs to be fully evacuated for any human presence before starting the accelerator. A PLC based search and secure system has been installed to ensure all the regions of the cell area has been thoroughly checked for human occupancy and there is no human present before we can start the accelerator. The whole search and secure operations is carefully designed to check each and every corner of the cell area. Besides this the system provides an option of emergency shutdown of the accelerator. The emergency switch off button has been provided at various locations in the cell area. If a person is #vijay9819420563@gmail.com

trapped then he can press the emergency button and the high voltage in the accelerator will switch off immediately and source of radiation will cease off.

The cell area has one shutter and one entry/exit door. Before starting the search operation the shutter is closed and its status is monitored using a limit switch. The Entry/exit door has been provided with a mechanically close but electrically locks type of door lock. The door lock also provides the door open/close feedback signal for interlock purpose. The door lock operates automatically and controls the entry of any new visitor before the search operation starts. In order to locate any visitor or operator inside the cell search operation is performed. Seven no of search and secure units have been installed at carefully selected locations inside/outside the cell area. Each of this unit consists of two button and two indicators. Green Button as 'secure' (it is to be pressed during search operation), Red button as 'Emergency OFF' (It is to be pressed by the person trapped inside the cell to switch off the accelerator), 'Search ON' indicator indicates that the unit is active and 'secure' button can be pressed. If the secure button is pressed, the second indicator indicating 'Secure' will turn on indicating the unit is secured. Performing the 'secure' operation on these field units ensures the operator who is performing the search and secure operation physically goes to the location where the unit is located and ensures that he has personally ensured that nobody is present in that region.

All the seven Search & Secure field units are connected to the PLC controller unit for control and monitoring. The limit switch to sense the shutter close and electrically lockable lock and a hooter is also connected to the controller unit. Before the search operation starts, the shutter is closed in order to stop any new visitor entry. The Search & Secure (S&S1) unit get power automatically when the shutter closer switch turns ON. The operator will secure the unit S&S1. This operation also starts a hooter for 20seconds inside the cell to indicate the search operation has started so that if anybody present in the cell can move out. Securing the S&S1 will unlock the door for 10 seconds so that the operator can go inside the door and close the door. The door gets automatically locked after 10seconds. This closure disables any new entry. After the door is closed S&S2 unit gets the power from the PLC. The operator secures the unit S&S2, unit S&S3 gets the power from PLC and so on. After all the six S&S units have been secured the door gets unlocked for 20 seconds so that the person performing the search operation can come out of the cell area. The operator clears the S&S7 in order to finish the

search operation. The timing between each of the S&S operation has been set by PLC such that the total operation has to be performed within in a fix time of 180seconds. This time duration has been a comfortable period to perform the operation. If the operator spends more time then he adds the vulnerability of a visitor located inside the cell to move into other secured areas. If the operator fails to finish the operation and come out of the cell within 180seconds, the PLC will reset all the S&S units and the whole operation has to be done again. Output from all the S&S units and door interlocks are connected in series to generate a hardwired interlock signal. This signal is connected in series to the high voltage of the accelerator. If any of the series connected interlock fails the HVDC will switch off and make the accelerator off. The sequence of search operation can also be seen on the HMI panel. Other safety interlocks such as radiation meter. Ozone monitor are also connected to this safety system. The accelerator control system has been designed using industrial PLC in such a way that an operator with short duration training can operate the accelerator. The control system has been equipped with the self diagnostic features for quick finding of faults in the failed subsystem. This feature reduces the down time of the accelerator by giving type and location of fault hence helps in quick recovery of the accelerator.

CONCLUSION

The safety control system has been commissioned and it is working satisfactorily. Operation of the accelerator has been done on trial basis. Different mock trials have done to check the effectiveness of the safety system. The system has been cleared for operation by local safety committee as well DSRC (design safety review committee). This system has the merit that it is offers timing and sequence flexibility but retains the safety merit of hard wired circuit.

ACKNOWLEDGMENT

We gratefully acknowledge the technical support offered by the High voltage, vacuum, beam optics and other auxiliary system groups in giving the clear understanding of the accelerator. The information provided by all these people has helped us a lot in the system development, commissioning and operation.

MACHINE THROUGHPUT IMPROVEMENT ACHIEVED USING INNOVATIVE CONTROL TECHNIQUE

V. Sharma[#], S. Acharya, K.C. Mittal, BARC, Mumbai, India

Abstract

In any type of fully or semi automatic machine the control systems plays an important role. The control system on the one hand has to consider the human psychology, intelligence requirement for an operator, and attention needed from him. On the other hand the complexity of the control has also to be understood well before designing a control system that can be handled comfortably and safely by the operator. As far as the user experience/comfort is concerned the design of control system GUI is vital. Considering theses two aspects related to the user of the machine it is evident that the control system design is very important because it is has to accommodate the human behaviour and skill sets required/available as well as the capability of the machine under the control of the control system. An intelligently designed control system can enhance the productivity of the machine.

INTRODUCTION

The control system has been developed for subsystems of three different electrical machines. The present works describes how the throughput of a machine can be enhanced by innovative control system design.

SYSTEM OPERATION

Control system for three different types of electrical machines has been designed/ developed and installed. Following is the description of the machine productivity improvement by control system design:

1] Beam utilisation factor improvement from 50% to 98% by modulating the speed of the convey system: A 10MeV, Electron beam, RF Linac is operational at EBC, Kharghar, Navi Mumbai. The beam output scans one meter length in a scan horn. The product under irradiation is placed below the beam horn on a conveyor trolley. The trolley is one meter long and there is one meter gap between each of the trolley. With the highest possible (limited by trolley track conditions and angle of turn since the trolley follows the zigzag path) constant speed of 5mtrs/min operation of the trolleys, the beam utilization by the product kept in trolley will be 50% since same amount of the beam falls in the gap between the trolleys. We have modulated the speed of the trolley. The trolley speed is reduced from 5mtrs. /min to 0.1mtrs/min when it is under the beam otherwise when the beam falls between the trolleys the conveyor speed is kept at 5mtrs. / min. This speed modulation operation is performed by sensing

vijay9819420563@gmail.com

the arrival and departure of the trolley using a limit switch which is activated by the base of the trolley. Using this method the beam utilization for the irradiation goes up to 98% from 50% hence 48% rise in productivity.

2] Productivity improvement of the Electromagnetic machining facility: APPD/BARC has developed a 20kV 10KJ Electromagnetic machining (EMM) facility. In this EMM facility a large value capacitor is charged by a DC supply to a set voltage. This charged capacitor is then quickly discharged using triggered spark gap into a coil to generate an intense magnetic field. This magnetic field generates the eddy current into the job piece to do the mechanical forming. We have developed and installed a PLC based control system to control the EMM machine. We used to charge the capacitor bank manually before the control system was installed. A DC voltage source made using motorised variac was being used for the capacitor charging. When the voltage is set the capacitor takes time equal to five RC times constant by charging the capacitor exponentially to charge to the set input voltage.

With the implementation of PLC based control system in the EMM facility the capacitor voltage has been conditioned /isolated and fed to the PLC. When the operation starts the PLC sets the double voltage as the charging voltage to the capacitor bank and trip voltage as the capacitor voltage desired. When the power is switched ON the capacitor starts charging very fast since the input voltage is two times to that of the desired voltage. As soon as the desired voltage on the capacitor bank is reached the control system isolates the charging supply to the capacitor bank. This technique reduces the total charging time to 1/7th to that of the earlier technique. Since the EMM facility is an industrial production machine, the productivity is a direct function of the capacitor charging time. Hence it can be stated that the total throughput of the EMM machine can be enhanced by 700%.

CONCLUSION

The control system has been commissioned and it is working satisfactorily. The capacitor charging time has been reduced but the full utilisation of this reduced time can only be done when the job piece under machining is loaded on the machine using automatic feeding system.

ACKNOWLEDGMENT

We gratefully acknowledge the technical support offered by the developer team of the EMM facility/ accelerator design team in making us understand the machine operation and the control process. This has helped us a lot in the development, commissioning and operation of the control system.

DESIGN AND ANALYSIS OF SECOND HARMONIC MODULATOR FOR DC CURRENT TRANSFORMER

Reju K, Kuldeep Joshi[#], BARC, Mumbai 400 085, India

Abstract

DC Current Transformers (DCCT) are widely used in particle accelerators. DCCT is a device which produces even harmonics, predominantly second harmonics corresponding to DC beam current flowing through two toroids. The second harmonics is detected by digital synchronous detector implemented in programmable logic. Current proportional to the detected second harmonic is passed through the toroids in a feedback loop such that the flux due to the DC beam current is cancelled by it. This feedback current is the measure of average beam current. The high permeability toroid's, excitation and output windings are collectively called magnetic modulator, which is a key component of DCCT. Design and analysis of a second-harmonic magnetic modulator used as a detector for DC Current transformer for high resolution current measurement is presented.

INTRODUCTION

Ion current in a particle accelerator is a key performance measurement parameter. Based on the requirement of a particular experiment various configurations of the particle beam are required, thus the characteristics of the beam are different for these configurations. In order to have a measure of performance the average beam current forms a useful parameter for measurement. DCCT is non-destructive current measuring instrument in particle accelerators which can be calibrated. We have been involved in a project of technology development for Accelerator Driver Subcritical Systems [1] and as a part of development of high resolution DCCT, a second harmonic magnetic modulator for DCCT was designed and implemented. The following sections describe principles of second harmonic modulator. Processing algorithm magnetic was implemented in programmable logic. Experimental results of the magnetic modulator in a feedback loop have also been presented.

SECOND HARMONIC MODULATOR

A second harmonic magnetic modulator in its simplest configuration consists of a single high permeability toroid core with excitation and output winding. The particle beam current through the core is considered as input signal. If the toroid is excited to saturation by an antisymmetric waveform, the resulting magnetic flux and hence the signal induce in the output windings contains fundamental and odd harmonics of the excitation frequency. When a dc current is passed through the toroid, even harmonic contents also appear in the output. The main disadvantage of this configuration is the presence of large amount of odd-harmonics in the output. These odd-harmonics are comparatively much larger than the second harmonics and there is a probability of overloading the succeeding stages of electronics. So double core modulators are normally used for this applications as shown in Fig. 1. Two identical cores arranged in an opposition manner so that the odd harmonics would cancel each other. Ideally, output V_d will be zero if there is no input current, provided the cores are matched. In presence of input signal even harmonic components appear in the output and when the input signal changes sign V_d undergoes a phase reversal. But in practical conditions, imperfections in core matching and the presence of even harmonics in excitation signal causes zero error in all types of magnetic modulators. The earth's magnetic field and any other stray fileds, thermal e.m.f.s in circuit connections are the other causes of zero error and drift. The zero error caused by memory effects should be removed by proper demagnetization [2]. Either the peak value or second harmonic component of the output can be a measure of input signal. One of the advantages of second harmonic detection over the peak detection method is that, the succeeding amplifier gain can be more as the amplifiers are not loaded by other harmonic components [3]. Toroidal cores are the most critical components of the modulator. Magnetic properties of these cores are the main factors which determine the resolution and the zero stability of the instrument.



Figure 1: Schematic of second harmonic magnetic modulator.

[#] kuldeep@barc.gov.in

(3)

Core Selection and Characterization

The magnetic cores were selected based on criteria as discussed in paper by Unser [4]. Core dimensions were selected according to the beam pipe dimension.

For further identifying the matched pair from the set of toroids, B-H parameters were measured. Excited each cores up to saturation by sinusoidal signal and measured voltage across the resistor in series with primary winding (V_1) , which is the measure of excitation current and voltage induced in the secondary terminal (V_2) . From V_1 and V_2 , magnetizing force (H) and magnetic flux density (B) were calculated as per the Equations (1) and (2) respectively.

$$H = \frac{NV_1}{R \, l} \tag{1}$$

$$B = \frac{V_2}{4.44fNA_C} \tag{2}$$

where N is the no. of turns, l is the effective path length of the core and R is the series resistance, f is the frequency of the input sinusoidal signal and A_c is the cross sectional area of the core.

It is necessary to determine the permeability at different frequencies (as shown in Fig. 2). Inductance of the core was measured with LCR meter and permeability was calculated by using Equation (3).

 $L = N^2 \frac{\mu A_C}{l}$

Figure 2: Permeability vs. frequency curve.

MODELING AND SIMULATION

In order to study the behaviour of second harmonic magnetic modulator, the selected toroidal cores were modelled in PSPICE based on Jiles-Atherton model of a ferromagnetic core [5]. Core models were made according to the core vendor data sheet as well as the experimental B-H curve data.

B-H curve in gauss vs. oersted was plotted by running the transient analysis on the test circuit. Jils-artherton parameters were extracted from the B-H loop and analysed the mismatches between cores. The magnetic modulator circuit was modelled (shown in Fig. 3) by using the two core models core1 and core2. L1 and L2 are the excitation windings of core1 and core2 respectively. L3 and L4 function as the output windings of core1 and core2 respectively. L5 and L6 are single turn signal windings on core1 and core2 respectively. Both the excitation windings are in parallel and there is a 10hm resistor in each branch for limiting the current at the time of saturation. A sinusoidal voltage is used as excitation source.



Figure 3: Magnetic modulator circuit.

The core1 and core2 (cores having mismatch) were replaced with ideal cores. If two cores are identical the combination doubles the even harmonic output components and reduces the odd harmonic output components to zero. If the cores are unbalanced there will be odd harmonics and hence a non-zero voltage in the modulator output even if it is operated with zero input signal. Figure 4 shows the magnetic modulator peak output for ideal and non- ideal core configurations. Here, it is seen that the unbalance in the cores causes noticeable zero error. But for the second harmonics zero error is comparatively small as shown in Fig. 5.



Figure 4: Peak modulator output vs. input signal.



Figure 5: Second harmonics output vs. input signal.

DESIGN AND DEVELOPMENT

The toroid cores used for magnetic modulator are made up of amorphous magnetic alloy tapes. Core dimensions were decided according to the beam pipe diameter. By analysing the BH curve and permeability variations with respect to frequency, we selected the operating frequency and number of turns for excitation coil winding. The mismatch in excitation windings were adjusted by adjusting the number of turns. The magnetic modulator in a closed loop is shown in Fig. 6.

The major criterion of designing excitation generator is its capability to provide sufficient mmf to saturate the core. A sinusoidal excitation signal of 10 kHz frequency with the help of programmable logic and 16 bit DAC module was generated. DAC output was filtered and amplified with power amplifier.



Figure 6: Magnetic modulator in a closed loop – low frequency channel of DCCT.

The second harmonics of the modulator output was extracted by a digital Lock-in Amplifier implemented in programmable logic. Hardware also detects the phase of second harmonics and it was adequate for the control action and feedback loop implementation

A digital PID controller was implemented using programmable logic. Final PID output was fed to the amplifier which provides a current in order to nullify the effect of beam current.

RESULTS AND CONCLUSIONS

The magnetic modulator was tested with both primary and secondary side pick-ups. Primary side output was taken across the series resistors $R_{\rm A}$ and $R_{\rm B}$ as shown in Fig. 1 and summed up by an instrumentation amplifier. In secondary side pick-up, the output was directly taken from the output winding. The modulator having secondary side pickup showed better common mode rejection and fundamental harmonic suppression than that of primary side. The memory effect observed in the transfer curve was reduced by proper adjustment of excitation voltage. The output of PID controller is shown in Fig. 7. There is noise below 30 µA is visible. DCCT implemented with the second harmonic modulator was tested in laboratory with the help of a calibrator kit which is capable of supplying DC current with a resolution of 1 μ A. We achieved a 30 μ A resolution of measurement in the range of ± 30 mA. Bandwidth of the measurement was DC to 0.1Hz. Figure 8 shows the final output of DCCT vs. the input DC current.



Figure 7: PID output for zero input signal.



Figure 8: DCCT output vs. input DC Current.

ACKNOWLEDGMENT

We thankfully acknowledge the advice and encouragement provided by Dr. S.K. Gupta, Mr. T.K. Saha, Dr. M.S. Bhatia and Dr. Chandrashekhar Rao for this work. We also wish to acknowledge the assistance given by Shri. Prashant Davange, Shri. P. D. Thomas and Ms. Shobha Jagtap for undertaking the experiment.

REFERENCE

- [1] S.B. Degweker et al, "The physics of accelerator driven subcritical reactors", Pramana-journal of physics, February 2007, vol.68, p.161-171.
- [2] E.H. Frost Smith, "The theory and design of magnetic amplifiers", 1966.
- [3] F.C. William et al, "The fundamental limitations of the second harmonic type of magnetic modulator as applied to the amplification of small DC signals", Proceedings I.E.E.-Part II, 1950, p.445.
- [4] K.B. Unser, "The parametric current transformer, a beam current monitor developed for LEP", AIP Conference Proc. April 1992, vol.252, p.266.
- [5] U.D. Annakkage, "A current transformer model based on the Jiles-Atherton theory of ferromagnetic hysteresis", IEEE Transactions on Power Delivery, January 2000, Vol.15.

FLOGBOOK: FROM CONCEPT TO REALIZATION

B.S.K. Srivastava*, R.K. Agrawal, K.G. Barpande, P. Fatnani, C.P. Navathe RRCAT, Indore, India

Abstract

Indus-1 and Indus-2, the Synchrotron Radiation Source (SRS) facilities at RRCAT Indore are national facilities and being operated on round the clock basis to provide synchrotron radiations to users as well as carrying out machine studies. Both of these accelerators are widely distributed systems and employ many sub systems for their operation. These sub systems are also made up of heterogeneous type of hardware and software modules. Indus-2 Control System is presently controlling approximately 10,000 input/output parameters for its operation. To keep the whole system up and running the faults & failures encountered during machine operations are attended at site and all observations and rectifications information are to be recorded electronically by the crewmembers. FLogbook has been conceived & developed to meet such needs. This web based software operates in the intranet environment over three tier software architecture. It mainly uses JavaServer Pages (JSP), JavaBeans and SQL databases for designing its building blocks. Using relational database features have been provided in the package for logging, e-mailing, searching & commenting the faults of various sub systems. Recently we have deployed the FLogbook in the field and machine operation crewmembers have been asked to use it as & when required.

INTRODUCTION

The main motivation of this software is to constantly monitor and improve the performance of Indus-1 & Indus-2 accelerators.

Indus-1 & Indus-2 are big & complex machines and involves various types of hardware and software for running it smoothly at round the clock basis. Seeing the complexities of various parts of both accelerators, it was felt to have a system that would systematically record the fault & failure information of accelerators encountered during its operations. This information would be recorded by accelerators crewmembers.

REQUIREMENTS

Based upon the previous experiences and feedback received from various system experts following basic requirements were considered before developing the FLogbook:

(1) It should be possible to use the FLogbook by multiple users simultaneously connected over campus network.

*bsks@rrcat.gov.in

- (2) The FLogbook should be easy to use for especially non computer experts.
- (3) Only authenticated users should be able to use the FLogbook. It should be possible to log the textual as well as non textual data e.g. image, documents etc into FLogbook
- (4) It should be possible to transmit the logged information with attachment to concerned subsystem experts electronically.
- (5) It should be possible to query & retrieve the stored historical data with different search arguments.
- (6) It should be possible to add comments on logged faults by concerned sub-system persons.

SOFTWARE DESCRIPTION

Access to FLogbook has been protected with password so that only authenticated users could use the system. System checks user's credentials using institute's central e-mail server. Hence users may use the software using his/her official e-mail login and password.



Figure 1: FLogbook Features.

Essential features of FLogbook have been shown in Figure 1. FLogbook is equipped with a software module for saving the computer's desktop in popular image files. This module has been developed in java and packaged in an executable jar file. On clicking the link provided in application this software module downloads & installs on user's machine & sits silently in system tray. This module is very useful for instant grabbing of machine control console desktop if any abnormality or unusual is noticed by the crewmembers.

Data Input

Using HTML & JavaServer Pages (JSP) Flogbook provides the form for logging the fault related information. Some fields in this form are automatically filled by the system and file attachment (e.g. doc, excel, jpg, png, etc) along with textual data may be logged by the crewmembers for future reference. Logged textual data with attached document (if any) may also be sent immediately to system concerned persons through emails. FLogbook gives unique fault-id to each logged fault so that historical fault could be diagnosed in future by its fault-id.



Figure 2: FLogbook Information Searching Form.

Data Output

Flogbook provides query form (Figure 2) for retrieving the logged historical faults with different search arguments e.g. timestamps, system name, text string. Feature has been provided by the software to log comments on retrieved specific historical faults and email it to selected concerned persons.

Software Design

Java Server Pages (JSP), JavaBeans and SQL databases have been used for designing & developing the various components of FLogbook. JSP with JavaBeans are used for developing the applications for web based environments. Java is main programming language used here for developing the complete FLogbook.

SOFTWARE ARCHITECTURE

The FLogbook follows the three-tier software architecture for designing & executing its building blocks. Here Web Browser resides on client machine and work as first or client tier. In our working environment we mainly use Microsoft Internet Explorer (IE) for accessing the web sites and web based applications hence we have developed and tested the Flogbook for IE users. FLogbook uses JavaServer Pages (JSP) & JavaBeans for

developing the presentation/view and business/application logic. JavaServer Pages (JSP) technology enables Web developers and designers to rapidly develop and easily maintain, platform independent, information-rich, web applications that leverage existing business systems.



Figure 3: FLogbook Architecture.

JSP technology separates the user interface (content presentation) from content generation, enabling designers to change the overall page layout without altering the underlying dynamic content. Tags for content *access and presentation* reside in the webpage. Logic and programming code for content *generation* reside in reusable components. After receiving the client request, the JavaServer Page requests information from a JavaBean. The JavaBean can in turn request information from a database (Figure 3). Once the JavaBean generates content, the JavaServer Pages can query and display the Bean's content. JavaBeans components (beans) are reusable software programs that we can develop and assemble easily to create sophisticated applications.



Figure 4: Execution of a JavaServer Page (JSP).

FLogbook uses JDBC (Java Database Connectivity) inside JavaBeans components for accessing the FLogbook database for inserting & retrieving the information.

The Type 4 Driver for JDBC has been used which provides JDBC access through any java-enabled applet, application, or application server. It delivers highperformance point-to-point and n-tier access to SQL database across the internet & intranets.

The JavaServer Page is identified to the server by a .jsp extension; this tells the server that special handling is required. As shown in Figure 4, the first time a request is made for such a file, the *isp* file is translated to a servlet & compiled into an object. (For that reason, there can be a slight delay on the first request for a .jsp page.) The output from the object is standard HTML which the browser interprets and displays as usual. After compilation, the compiled-page object is stored in memory on the server. On subsequent requests for that page, the server checks to see if the .*jsp* file has changed. If it has not changed, the server uses the compiled-page object stored in memory to generate the response to the client. (Because the object is stored in memory, the response is very fast.) If the .jsp file has changed, the server automatically recompiles the page and replaces the object in memory.

We have used Apache Tomcat as a web server for executing & serving web components of FLogbook. Apache Tomcat (or simply Tomcat) is an open source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Oracle Corporation, and provides a "pure Java" HTTP web server environment for Java code to run. FLogbook uses JavaMail API for sending the e-mails composed of the information logged by the operation crewmembers into FLogbook database. The JavaMail API provides a platform-independent and protocolindependent framework to build mail and messaging applications.

Microsoft SQL Server based relational database was designed to implement the data tier of FLogbook, which stores the complete information in related tables. Subsystem names of both the accelerators (Indus-1 & Indus-2) and concerned persons details are stored in the database so that logged fault information could be mailed electronically.

SOFTWARE DEPLOYMENT

The FLogbook has been deployed on Apache Tomcat web server configured on Gateway machine. This computer as shown in Figure 5 is connected with accelerators' technical network (AccNet) as well as campus network (RRCATNet). The URL of the application has been mapped in DNS of both the networks so that it could be accessed uniformly from the machines of both networks. DNS Server of accelerators' technical network (AccNet) is running on Primary Domain Controller (PDC) Server machine.



Figure 5: Deployment of FLogbook.

CONCLUSION

The first version of FLogbook has been deployed in the field and being used by the operation crewmembers of both accelerators (Indus-1 & Indus-2). The system is very useful for not only the accelerators operation crewmembers but also for machine/subsystem experts for tracking the faults and improving the overall machine performance.

Presently, we are using Microsoft Windows based infrastructure and if we switch to Linux based infrastructure then very little modifications are required in codes of the system during porting.

DEVELOPMENT OF A MONITORING SYSTEM FOR THE FL-net PROTOCOL

M. Ishii[#], T. Masuda, S. Ueda, JASRI/SPring-8, Hyogo 679-5198, Japan T. Fukui, RIKEN/SPring-8, Hyogo 679-5148, Japan

Abstract

FL-net is being used as a communication protocol between the front-end computers and PLCs in several control systems at SPring-8 and SACLA. It is an Ethernet-based open standard protocols used for a factory floor networks. Furthermore, it is an UDP/IP-based master-less token passing protocol and supports cyclic transmissions. SACLA has experienced certain problems in terms of data acquisition when using FL-net for a beamline machine protection system. To resolve these problems, we developed a monitoring system for the FLnet protocol that captures and analyzes all packets of an FL-net network segment, detects protocol failure events, and stores the event information in a relational database. We can easily refer to the stored information in the database using a Web browser. The software-based monitoring system is highly portable and does not require a dedicated hardware-implemented protocol stack. This paper presents the monitoring system design for FL-net.

INTRODUCTION

FL-net was first introduced at SPring-8 in 2005 as a field-bus for the communication between the VME system and PLCs. The control systems using FL-net are as follows: a status monitoring of linac interlock system at SPring-8 [1]; a vacuum, RF high-power, and low-noise power supply control system; an undulator control system; a precise water temperature control system at SACLA [2]; and a facility control system at both SPring-8 [3] and SACLA [4]. Figure 1 shows a schematic view of a typical control system using FL-net at SPring-8 and SACLA. In our control system, FL-net shares Ethernet switches through the use of virtual LAN technology (VLAN). The SACLA control system currently utilizes 45 types of VLAN segments for FL-net. Packet capturing and protocol analyses are helpful for troubleshooting the network. On the other hand, analyzing a very large amount of captured data requires a lot of time and effort. In addition, since problems may occur at any time, we have to monitor the packets over the long term. For the SACLA control system using FL-net, a problem occurs at least once a month. While an FL-net protocol analyzer has been released on the market, it was not designed for a long-term monitoring.

We therefore developed a monitoring system for the FL-net protocol in March 2012, and started monitoring the system in May. Herein, we describe the design of the system and the effects of its installation.



Figure 1: Schematic view of a typical control system using FL-net at SPring-8 and SACLA.

FL-net

The FL-net protocol was authorized by the Japan Electrical Manufacturers' Association (JEMA), and was established as a JEM standard in 2000 [5] and as a Japanese industrial standards (JIS) in 2004 [6]. Since several PLC vendors provide FL-net interface modules, we can build a control system using multiple vendor products. The FL-net protocol has the following features.

- The standard UDP/IP Ethernet communications protocol is used. Cables, hubs, and other networking components are readily available.
- The network operates at 10 Mbps (version 2.00).
- Up to 254 nodes can be connected to the physical network layer.
- Message transmissions for asynchronous data communication and a cyclic transmission are supported.
- Message transmissions sends/receives up to 1024 bytes of data.
- A cyclic transmission uses the common 8-Kbits and 8-Kwords memory shared by all nodes.
- Nodes can be automatically connected to or disconnected from the FL-net network by adopting a master-less token method.

SYSTEM DESIGN

Our design policy was to capture and analyze all packets flowing through the Ethernet of the FL-net network, and to record all problematic events. In FL-net, and to record all problematic events. In FL-net, and to record all problematic events. In FL-net, and the record all problematic events are of ~ 4 Mbps. If all packets of the captured data are saved as files in pcap format, which is a standard format treated by open-source network analyzers such as tcpdump [7] and Ξ

[#] ishii@spring8.or.jp

Wireshark [8], the disk usage reaches up to $\sim 30 \text{ GB/day}$. We therefore have to store only the necessary data to save disk space. Our requirements for the monitoring system are as follows:

- The system automatically saves only several 10-s long data packets including the problem event as a pcap format file.
- We can select parameters such as the type of detection events, and save directory using a Web browser.
- The system stores information on a problem event with a time stamp in a relational database. We can easily refer to this information using a Web browser.
- The system has no influence on the operation of the FL-net system.
- The system monitors several FL-net network segments, and we can access the monitoring system



Layer configuration

Network frame configuration

Figure 2: The layer configuration and network frame configuration of the FL-net protocol.

through the Ethernet.

• The system has the highly portability and is independent from the hardware.

We developed a software-based monitoring system running on a Scientific Linux. In the FL-net protocol, the FA link protocol layer, including the FA link header and user data, is located above the transport layer. The monitoring system analyzes only the FA link header. Figure 2 shows the layer configuration and network frame configuration of the FL-net protocol.

Figure 3 shows a schematic view of the monitoring system. The black line indicates a data flow, and the green line is a process management flow. The monitoring system consists of an administration process (Adminproc), a packet capture process (CP), a protocol analysis process (AP), a MySQL database, and a Web server in a computer.

The CP captures all incoming packets and saves them as a temporary file with 3 to 4 MB in size for each 10-s of data, and writes the temporary file name to the database. The file size depends on the packet capture period. From the database, the AP acquires the file name of the temporary file to be analyzed. The AP then analyzes the temporary file according to the detection events such as the participation and removal of a node, a token skip, and the duplication of the node address. If the AP detects an event, it saves the packet data including the event as an archive file and writes the event information with a time stamp to the database. The AP then sets an analysis complete flag for the temporary file to the database. The CP acquires the temporary file name set the analysis complete flag from the database, and deletes the

Ethernet for control system



Figure 3: Schematic view of the monitoring system for the FL-net protocol.

temporary file. We can check the packet flow before and after a problem event off-line using the archive files. If the FL-net is normal, an archive file is not generated.

The Admin-proc manages the creation, termination, and alive monitoring of the CP and AP. The CP and AP update a time stamp as keep alive to the database.

A Web server creates and terminates the Admin-proc, and displays lists of information, such as the node status, event history, and saved archive files from the database. Figure 4 shows the main page on a Web browser. We can access the connection history, archive file list, timer information, event history, parameter configuration, process status, and help from the menu bar. In addition, we can obtain the node number, connection status, 4th octet of the IP address, the offset address and size of the allocated common memory, and the token watchdog timer for each node.



Figure 4: The main page of the Web browser. The language is set to Japanese. The red line around the box No. 1 area indicates the menu bar used for browsing The No. 2 area is the FL-net select bar, which can start/stop the monitoring by the click of a button Finally, the No. 3 area shows the node status and configuration list.

APPLICATION

We built the monitoring system on a computer with a quad core Xeon 2.4 GHz CPU, 16 GB of memory, a 1 TB HDD, and five Ethernet ports. One Ethernet port is for Web access and the other Ethernet ports are for monitoring four FL-net network segments. The average CPU load is less than 0.1.

From the start of the SACLA operation, there were problems with the status monitoring of the beamline machine protection system (MPS) during data acquisition, such as the status of the main beam shutter opening/closing, and the gate valve status at the front-end and transport channel vacuum system. The MPS is consisted of PLC modules. The VME system acquires data from the PLCs using FL-net (Fig. 1). On one occasion, the VME FL-net board was suddenly blocked, and data acquisition was ceased. Before the installation of the monitoring system, we were unable to determine whether the source of the problem was a network switch, VME computers, PLC modules, an application running on the VME system, the PLC ladder software, or burst traffic. After its installation, the monitoring system detected that the source of the data transmission blockage was the VME FL-net board. We then examined and upgraded a firmware of the VME FL-net board. Since this firmware upgrade, the data acquisition of the MPS has been running stably.

Data acquisition for precise temperature control system of undulators at SACLA also involved some issues. The VME system acquires data from the PLCs through FL-net as an MPS. By using the developed monitoring system, we found that many token skips had occurred, and the token cyclic time exceeded the token watchdog timer set for each node. We were able to determine the setting configuration of each node based on the node status and configuration list of the monitoring system. We will modify the configuration to the appropriate token watchdog timer during the next shutdown period.

SUMMARY

Several control systems using FL-net have been built at SPring-8 and SACLA. We developed a monitoring system for the FL-net protocol, which captures and analyzes all packets of an FL-net, detects problem events, and stores the event information into a relational database. Through the installation of this monitoring system, we have resolved certain issues in data acquisition for the beamline MPS and precise temperature control system for undulators at SACLA. The monitoring system is effective in identifying the source of system problems.

REFERENCES

- [1] T. Fukui et al., "Development of a communication with PLC by using the FL-net as an open standard PLC link," Proc. of PCaPAC2005, Hayama, Japan, 2005.
- [2] R. Tanaka et al., "First operation of the SACLA control system in SPring-8," Proc. of IPAC2011, San Sebastian, Spain, 2011, p. 2325.
- [3] M. Ishii et al., "The Gateway of Facility Control System for SPring-8 Accelerator," Proc. of ICALEPCS2011, Grenoble, France, 2011, p. 473.
- [4] T. Masuda et al., "Facility Utility Control System of XFEL/SPring-8," Proc. of ICALEPCS2009, Kobe, Japan, 2009, p. 286.
- [5] http://www.jema-
- net.or.jp/Japanese/standard/opcn/opcn07.html
- [6] http://www.jisc.go.jp
- [7] http://www.tcpdump.org
- [8] http://www.wireshark.org

MODULAR BEAM DIAGNOSTICS INSTRUMENT DESIGN FOR CYCLOTRONS

N. Chaddha[#], R. B. Bhole, S. Sahoo, P. P. Nandy, S. Pal VECC, Kolkata, India

Abstract

The Cyclotrons at VECC, Kolkata i.e. Room Temperature Cyclotron (RTC) and Superconducting Cyclotron (SCC) comprise of internal and external Beam Diagnostic systems. These systems provide the beam developer with position, intensity, beam profile, a visual impression of the size & shape of ion beam, and operational control over diagnostic components like 3finger probe, Beam Viewer probe, Deflector probe, Faraday cup, X-Y slit, Beam viewer etc. [1]. Automation of these components was initially done using customised modules for individual sub-system. An expansion of this facility and various levels of complexity demand modular design to cater easy modification and upgradation. The overall requirements are analysed and modular cards are developed based on basic functionalities like valve operation, probe/ slit/ viewer control, position read-out, Interlock, aperture control of beam line and communication. A 32-bit Advanced RISC Machine (ARM) based card with embedded EPICS is chosen as the master controller and FPGA/ microcontroller is used for functional modules. The paper gives a comprehensive description of all modules and their integration with the control system.

INTRODUCTION

Various upgradations of different sub-systems are being done in both the cyclotrons at VECC, Kolkata are engaged with different kinds of experiments. RTC is getting modernized with new breed of automated beam diagnostics equipments whereas upgradation of internal beam diagnostic system is being done to facilitate internal beam tuning for better extraction at SCC. All the beam regions (injection, acceleration, extraction & external line) are employed with many beam diagnostic stations as per the beam tuning and transportation requirements [2].

The beam diagnostic stations are equipped with different set of components which are either electrical or electropneumatically controlled. Development of customized instruments with uniform control scheme, as employed in other sub-systems is required to control and monitor these diagnostic components.

DESIGN CRITERIA

The modular design of the diagnostic control instruments has undergone through several modifications for multiple times due to fast changing and customized requirements. The earlier versions of these instruments had common data and control lines [3]. This design had

the restriction of sharing hardware resources due to interfacing compatibility. The recent development introduces an EPICS embedded main controller card, interfacing with other functional modules through dedicated serial lines on backplane. This design has given liberty to the development of individual module while keeping the same software architecture. The developer gets a choice of selecting his own tool chain according to the complexity of functional requirements of the module.



Figure 1: Block diagram of the basic scheme of module.

Each functional module is designed to have four building blocks: μ controller, functional component, communication and power supply as shown in Fig 1. AVR and C51 family controllers are used for the μ controller part. Functional part is designed as per the system requirements. Interface of these modules to main controller card is through dedicated TTL UART communication lines i.e. Rx (Receive), Tx (Transmit) and Ground. An optional RS232 line driver is also kept on each module for stand-alone operation.

HARDWARE MODULES

The newly designed instrument has a main controller card communicating with other functional modules and PC for controlling diagnostic devices. The description of developed hardware cards/modules are as follows:

Controller Card

The main controller card is designed using a Cavium ARM9 CPU based SBC (Single Board Computer) running on 250 MHz. This board has features like 64MB RAM, a bootable 4MB on-board flash, a microSD card slot and 5000 LUT (Look up table) Lattice FPGA. Interface to external devices can be done via Ethernet, USB host, USB device, or I2C ports as well as DIO, UARTs, and SPI which are implemented in the standard FPGA load. The SBC boots to Linux 2.6 from either an SD card or 4MB on-board flash having a bootable kernel image and an initial ram-disk image (Fig 2).

[#] nchaddha@vecc.gov.in



Figure 2: ARM-9 based Single Board Computer.

Stepper Motor Module

This card is designed to drive single high torque, high current stepper motor using FET based driver (Fig 3). The basic function of this module is to control the movement towards a given direction (in full/ half step) and to read the status of position switches & motor.



Figure 3: Block diagram of Stepper Motor module.

The module is designed to drive unipolar, bipolar and hybrid type of stepper motors. Optical isolation is provided between logic and driver part of the module to avoid unwanted glitches in communication lines.

X-Y Slit Module

The beam slits are used to control the beam aperture in the injection and extraction lines.



Figure 4: Block diagram of X-Y Slit module.

As the slits are installed in a pair {horizontal (X) and vertical (Y)} directions, the module is designed to control two slits as shown in Fig 4.

The movement of the slits is controlled by operating geared DC motors, one at a time, whereas position is read by potentiometric position encoder, fixed with the moving arms. Speed of the motors is controlled using built-in PWM in on-board controller. Positional switches are read during movement to restrict the motion beyond its limits.

Encoder Read-Out Module

All the major diagnostic components like three finger probe, viewer probe, magnetic channels etc. in SCC uses optical (Incremental & absolute) encoders, for position measurement. The encoder read-out module is designed interfaces with incremental type encoders to read 32-bit of absolute positional data (Fig 5).



Figure 5: Block diagram of Encoder Read-out module.

A count with specific multiplication factor of decoded pulses from quadrature clock converter provided positional data. Other features like auto correction of position using index pulses and local/remote calibration are also introduced to avoid cumulative error in position. The position data is saved in local memory and calculated absolute position is displayed in an LCD panel.

Relay and I/O Module

The basic function of the relay module is to operate different electro-pneumatic valves. Electromagnetic relays are used as switches for actuating diagnostic components like Faraday cup, Beam viewer, Gate valve, Track valve etc.



Figure 6: Circuit diagram of Input-Output module.

The I/O module comprises of relays operation with interlock inputs as shown in Fig 6. This card is useful for small beam diagnostic stations with a few components like combination of beam viewer & faraday cup in external beam line. Fig 7 shows example of an assembled module.



Figure 7: An assembled module.

CONTROL ARCHITECTURE

The control system for beam diagnostic in both the cyclotrons follows a 2-layer architecture. The operator interfaces (OPIs) are developed using MEDM (Motif Editor and Display Manager) which runs on standard Intel X-86 PCs. As per EPICS standard, OPIs interact with the IOC (Input-Output Controller) through control LAN following Channel Access (CA) protocol.

The main controller card, having ARM based Single Board Computer, runs the EPICS IOC. The IOC contains the required process variable list (database) for beam diagnostics purpose. The controller card communicates with functional modules through dedicated serial links. The device drivers for all theses hardware modules are written separately and included into the IOC.

CONCLUSION

Presently two sub systems are under operation with such newly designed instruments. The inflector positioning in the Superconducting Cyclotron is done with the combination of a linear and an azimuthal motion control. The external beam line diagnostic system in RTC is being upgraded from manual to automated controls using these new modules.

REFERENCES

- Suvadeep Roy, S. Bhattacharya, T. Das, T.K. Bhattacharya, G. Pal, C. Mallik and R.K. Bhandari, "Fabrication of Beam Diagnostic Components at Superconducting Cyclotron", Rising Expectations & Mounting Challenges, VECC, Kolkata, June 25-26, 2008.
- [2] Niraj Chaddha, et. al. "Control And Monitoring System For Internal Beam Diagnostics Facility Of SCC", InPAC, RRCAT, Indore, February 10-13, 2009.
- [3] Niraj Chaddha, et. al. "Embedded Systems for Diagnostics & Control Used in VEC and SCC", SACET, VECC, October 28-29, 2009.

FACILITY MONITORING SYSTEM USING STORAGE AREA NETWORK FOR VEC AND SCC

T. Bhattacharjee*, A. Roy, R. B. Bhole, G. Saxena, K. Datta, T. Samanta, S. Pal, D. Sarkar, VECC, Kolkata, India

Abstract

The facility monitoring system of cyclotron operational parameters at VECC is developed and commissioned recently. Storage Area Network (SAN) is used to isolate the control LAN and office LAN which ensures secured access of the control systems from outside world. EPICS gateway service and modified channel access save/restore tool have been used to integrate EPICS based control system of VEC and SCC with office network. This paper describes the implementation details of the overall facility monitoring system.

INTRODUCTION

The control systems of K130 variable energy cyclotron (VEC) and K500 superconducting cyclotron (SCC) at VECC, Kolkata, are isolated from each other as they run separately in their respective control network. However, online and historical monitoring of many important control parameters of both the facilities are required to be viewed from both the control rooms for proper diagnosis, maintenance and operation. Moreover, the facility managers, sitting at office network, require the automatic update of statistical information e.g. number of hours of beam time supplied to user, planned /unplanned shutdown etc. and hence, availability of control parameters from office network for automatic generation of operating reports and statistic in an application like spread-sheet for both the facilities are required. A facility monitoring system has been developed to cater all these requirements while maintaining access security aspects of the control networks as described in the subsequent sections.

PHILOSOPHY

A storage area network (SAN) is a dedicated network which is primarily used to make storage devices, accessible to computer servers. These storage devices appear like locally attached drive to the operating system of the server. A SAN network of storage devices is not accessible through the local area network by other devices and hence transaction of data from one network to other by means of SAN storage reduces vulnerability of cyber security threats. The facility monitoring system at VECC has been developed by adopting this idea where all the important parameters of both the facilities are transferred to the Office network for facility managers. A EPICS Process variable (PV) Gateway server [1], which can be * btanu@vecc.gov.in

configured to collect PVs from isolated networks and transmit them to another network, is installed. A software-based Firewall is also installed in the same Gateway server to provide the required access to perform CA monitor process to collect PVs from the IOCs running in the control networks. As all the required parameters can be accessed from the gateway server, one SAN server (SAN-1) is connected with the gateway server in a separate network. A channel access save/restore tool [2] is modified and installed in SAN-1 to save all parameters, to be forwarded in office-network, in the common storage. The other SAN server (SAN-2) which is connected in office network can access data-file containing the control parameters from the common storage. With this philosophy, one would ensure security of the control networks while passing all the required parameters in a separate network.

ARCHITECTURE

The overall system is divided into two layers where the first layer is the PV gateway layer (Dell PowerEdge) and second one is a storage area network where two servers (HP Server-1 and HP Server-2) are connected with a common storage through a point-to-point fibre channel as shown in figure 1. The gateway service is responsible for collecting all required parameters from SCC and VEC networks using EPICS channel access protocol. A channel access server process is available inside the gateway service which transmits all control parameters collected from the SCC and VEC control networks towards the SAN system. The gateway process has the ability to apply access security in addition to the access security configured in the firewall, running in the same gateway server, and hence, all incoming PVs are configured with read only access in the PV gateway.

The process running on SAN-1 server collects the monitoring parameters from the gateway server using CA protocol and saves the data into the common storage. The office-side server i.e. SAN-2 contains an duplicate EPICS IOC with all the PVs similar to the PVs collected from VEC and SCC control networks for monitoring. A modified channel access restore process on office-side retrieves the data from the common storage and update the duplicate IOC. An Apache-based web server running in SAN-2 collects the data from the IOC using EPICS CA-plugin application. The user can view the parameters in a web browser from the office LAN as shown in figure 🚍 2 and figure 3. The common storage (2.1 TB) is \bigcirc configured with RAID 5 and is divided into three logical



Figure 1: Architecture of SAN-based Facility Monitoring System.

units. One logical unit is currently used for storage of online data and two units are kept for future use of historical data trending from office LAN. The file locking and sharing functions are handled by Red Hat Global File System (GFS) for each server in the SAN system. The inside server of the SAN storage system has read/write access to the common storage whereas the outside one has been configured with read access only as shown in figure 1. In addition to this, the access security in the duplicate IOC is configured in such a way that it will only allow SAN-2 to update its database. These features prevent the write access of control data from office LAN to ensure additional data security.

VEC Live Status

25-4-2012, 02:58pm

Current Status Trim coil Power Supply Beam line Power Supply Deflector Power Supply Vacuum
Main Machine Parameters
Main Magnet(Ampere) 842.5

Gas Type	Helium	
Gas Flow(seem/s)	7.41	
Arc Current(Ampere)	0.5	
Arc Voltage(Volt)	379	
Filament Current(Ampere)	381	
Dee Voltage(kVolt)	0	
Main Magnet(Ampere)	842.5	

Figure 2: On-line control parameters of VEC in web browser.

SCC Live Status

25-4-2012, 03:00pm

SCC Current Status Beam line Vacuum RF Power and Phase RF Forward/Reflected Power Cryogenic Load

Main Machine Vacuum					
Main Macine-BC2(mbar)	5,99	E-07			
Main Macine-BC3(mbar)	5.49	E-07			
Main Macine-BC4(mbar)	7.61	E-06			
Main Macine-BC5(mbar)	NaN	E-00			
Main Macine-OVC(mbar)	4.44	E-07			
Main Macine-Lower Liner(mbar)	6.99	E-02			
Main Macine-Upper Liner(mbar)	7.54	E-02			

Figure 3: On-line control parameters of SCC in web browser.

SCOPE OF EXTENSION

So far, the facility monitoring system from the office network has been described, however, EPICS and MySQL interface, developed in-house [3] can also be installed and integrated to monitor the control parameters of VEC from SCC control room and vice-versa. The webtool support available for historical trending and online monitoring of archived data as shown in figure 4 and figure 5 in the EPICS-MySQL tool-set will enable to achieve this facility very easily. The control parameters can be collected from the Gateway server and Firewall will be configured in such a way that it will allow access of MySQL database from both the control room.

Copyright © 2012 by the respective authors

	ŀ	PICS	AySql A	Arcl	hiver	
		Started on: 1	2:34:22.633	, 2011	-01-09	
			Historic plot			
		Sv	stem: Vacu			
			ACTIN: YACIN			
		Sub-syst	tem: Main_r	nachin	e	
Name	Value	Time stamp	Date stamp	State	Log type	Description
Name VAC MM BC2	Value 6.60Se-07	Time stamp	Date stamp 2011-02-17	State	Log type scan	Description
VAC_MM_BC2 VAC_MM_BC3	Value 6.60Se-07 6.900e-07	Time stamp 13:36:08.236 13:36:08.236	Date stamp 2011-02-17 2011-02-17	State OK OK	Log type scan scan	Description
VAC_MM_BC2 VAC_MM_BC3 VAC_MM_BC4	Value 6.608e-07 6.900e-07 9.462e-06	Time stamp 13:36:08.236 13:36:08.236 13:36:08.236	Date stamp 2011-02-17 2011-02-17 2011-02-17	State OK OK	Log type scan scan	Description
Name VAC_MM_BC2 VAC_MM_BC3 VAC_MM_BC4 VAC_MM_BC5	Value 6.60Se-07 6.900e-07 9.462e-06 0.000e+00	Time stamp 13:36:08.236 13:36:08.236 13:36:08.236 13:36:08.236	Date stamp 2011-02-17 2011-02-17 2011-02-17 2011-02-17	State OK OK OK	Log type scan scan scan scan	Description
VAC_MM_BC2 VAC_MM_BC3 VAC_MM_BC4 VAC_MM_BC5 VAC_MM_BC5	Value 6.60Se-07 6.900e-07 9.462e-06 0.000e+00 8.6S9e-06	Time stamp 13:36:08.236 13:36:08.236 13:36:08.236 13:36:08.236 13:36:08.236	Date stamp 2011-02-17 2011-02-17 2011-02-17 2011-02-17 2011-02-17	State OK OK OK OK	Log type scan scan scan scan scan	Description
Name VAC_MM_BC2 VAC_MM_BC3 VAC_MM_BC4 VAC_MM_BC5 VAC_MM_DVC VAC_LOLINER	Value 6.60Se-07 6.900e-07 9.462e-06 0.000e+00 8.6S9e-06 8.179e-02	Time stamp 13:36:08.236 13:36:08.236 13:36:08.236 13:36:08.236 13:36:08.236 13:36:08.236	Date stamp 2011-02-17 2011-02-17 2011-02-17 2011-02-17 2011-02-17 2011-02-17	State OK OK OK OK OK	Log type scan scan scan scan scan scan	Description

Figure 4: EPICS-MySQL Archiver.



Figure 5: Historical trending in EPICS-MySQL interface.

CONCLUSION

The MySQL database can also be installed in the SAN-2 server which will collect all control parameters from duplicate IOC. It will provide historical trending facility from office network for facility managers as well for respective system personnel. The in-house CCTV system can also be integrated with this system for campus-wide monitoring facility of on-line cyclotron parameters.

ACKNOWLEDGMENT

We thank all our colleagues, related to various subsystem of the cyclotron, who made the commissioning successful by providing required supports. We are also thankful to Cyclotron Control Section for their constant support during the installation and commissioning.

REFERENCES

- [1] The Process Variable Gateway, http://www.aps.anl.gov/epics/extensions/gateway/index.ph
- [2] The EPICS Channel Access Save Restore Tool, http://www.aps.anl.gov/epics/meetings/2002-111/talks/maclean1.pdf
- [3] A. Roy et al., "EPICS MySQLArchiver Integration Between EPICS and MySQL", THCB02, these proceedings.

DESIGN AND IMPLEMENTATION OF AN IEEE 802.15.4 / ZIGBEE BASED STAR NETWORK FOR DATA ACQUISITION AND MONITORING

S. K. Guha[#], P. Y. Nabhiraj, T. K. Bhaumik, C. Mallik, VECC, Kolkata, India

Abstract

ZigBee based wireless technology is used to provide a low cost, low power, secured, PAN solution for monitoring of parameters from several distributed vacuum pumping modules installed in the SCC injection line. The parameters include module's pump RPM, input current and pressure reading of different vacuum gauges.

The ZigBee stack is written in a simplified form so that each node can create a network and can join to any established network when powered on. End nodes can be replaced through a little modification in the firmware codes. End node consists of sensors, signal conditioning circuits, microcontroller and ZigBee Transceiver whereas the Central node consists of microcontroller, Transceiver and UART interface. This paper highlights the future approach of utilizing this network for data acquisition related with environmental temperature, relative humidity, noise, water leakage from inaccessible areas of Cyclotron Vault, Pit, Basement and ECR Highbay for the ease of maintenance also the development of an environment monitoring system powered by solar cells covering a wide area.

INTRODUCTION

An Extremely Low Power (XLP) microcontroller based wireless network has been designed for several vacuum pumping modules using 2.4 GHz ZigBee [1] protocol on STAR topology. These pumping modules are installed at different locations of the SCC Injection line in a distributed manner. Two different network devices are designed to operate in this network, one is Coordinator and another is End device. Each End device acquires both analog and digital information from pumping module, generates a Zig Bee data frame for each sample and finally sends to the Coordinator. The frame structures follows IEEE 802.15.4 standards to keep the complexity to a minimum while at the same time making the network sufficiently robust for transmission in a noisy environment.

IMPLEMENTATION OF ZIGBEE

ZigBee is one of the global standards of low data rate wireless communication protocol formulated by the IEEE 802.15.4 working group. The ZigBee protocol has a transfer rate around 250 Kbps and 16 channels in the 2.4 GHz band [2], refer Fig.1, fully hand shakable protocol for transfer reliability, low power consumption (Tx or Rx mode or sleep mode) facilitating remote controlled battery operated systems and sensor network. The Higher Protocol Layers have been defined by ZigBee Alliance

#suman@vecc.gov.in



Figure 1: IEEE 802.15.4 PHY overview Operating Frequency band (Industrial, Scientific and Medical).

Group whereas IEEE 802.15.4 working group only utilizes the lowest two layers of OSI which are Physical Layer (PHY) and Data Link Sub Layer (MAC – Media Access Control Sub Layer) refer Fig. 2.



Figure 2: IEEE 802.15.4 Stack Architecture.

IEEE 802.15.4 standard supports several topologies refer Fig. 3 and the intention was to build a medium ranged (70 - 300 mtrs.) low data rate, low complexity wireless Personal Area Network to operate in an unlicensed international frequency range. ZigBee based wireless technology is implemented to provide such a solution.



Figure 3: IEEE 802.15.4 standard Combined Topology, Highlighting STAR.

SCHEME OF THE SYSTEM

The star network has been developed by using two kinds of Zigbee nodes. Several End nodes are designed for each of the vacuum pumping modules and one

authors

Network Coordinator it also interfaces with control PC refer Fig. 4.



Figure 4: Schematic of the network along with other beam line elements.

Each End node receives information regarding DC current drawn by the turbo pump, pumping speed and the pressure information as read by the adjacent vacuum gauges. The information is being sent to the coordinator against a request being received from it. At the coordinator end, after successful receiving of a frame from different end nodes the data part is being separated from the complete frame and the data bytes are being sent to Lab VIEW user end application program through serial communication.

HARDWARE CONFIGURATION

Both the nodes are using Microchip 8 bit PIC 18F46J50 MCU [3] a member of nano watt extremely low power (XLP) technology microcontroller family with 64K program memory and IEEE 802.15.4 standard compliant RF transceiver module which is based on 2.4GHz RF Transceiver IC MRF24J40, [4] 20 MHz Integrated Crystal, Internal Voltage regulator, Matching Circuitry and PCB Antenna refer Fig. 5.



Figure 5: Block diagram of End node and Coordinator.

The RF Transceiver communicates with MCU via 4 wire serial SPI protocol along with Interrupt, Wake and Reset Signal refer Fig. 6. While configuring RF transceiver the operation channel, data rate, address and other related issues has been specified by accessing the different Short address and Long address control registers through SPI interface. The end node acquires analog signals (0 – 10 V DC) from turbo pump controller and vacuum gauges. After being passed through the signal conditioning circuit the three analog data are fed to the ADC of the MCU through different channels. These three analog signals are sequentially sampled and the 10 bit digital equivalent data are stored in 6 subsequent memory

locations of an array which are appended in the Data frame in the MAC Sub layer by the stack program.



Figure 6: Connection details between RF Transceiver and MCU.

SOFTWARE STACK PROGRAM

The software protocol stack is modified by using MPLab C18 Compiler for both types of devices in a simplified form. It is basically based on MAC and PHY layers of IEEE 802.15.4 specification. This Stack provides a feature to find an existing network, form a new network or to join in a network. Specific 8 byte Extended Organizationally Unique Identifier (EUI) or MAC address and 2 byte PAN address have been loaded into respective control registers of the RF chip. It also enables to identify Received Signal Strength (RSSI) for all the received frames. Refer Fig. 7 for ZigBee Data frame format as used.



Figure 7: Data Frame format in MAC and PHY Layer.

The MAC layer program provides the information of the channel to be accessed (here in this case channel 24 has been utilized), generates the address information and append the data bytes into the MAC layer data frame. The protocol stack also supports the broadcast or unicast addressing mode. The network coordinator access each end node by pointing different arrays of the Destination address field sequentially only after receiving the response form an end node it address the next one. Because of the less number of nodes the entire network is designed in such a way that all the nodes are having same PAN (Personal Area Network) ID field. For simplicity only Data frames are utilized here. In case of changing any defective node, only 8 numbers of Extended Address Registers (EAD) has to be configured. User end Lab VIEW application program receives data bytes from a particular end node via the network coordinator through

RS 232 serial (19200,N,8,1 format) bus. The program flow diagram has been shown in Fig. 8.



Figure 8: Flow chart of the Firmware program.

THE OPERATOR INTERFACE

The user interface is being performed by application program developed using Lab VIEW [5] software (version 7.1) refer Fig. 9.

Be got gernie Jook Broese y	indow 1980						1 = X 1
30000000,0000 3000 9000 9000 9000 1000 1000	The state	100	reachuffer 72 decoral 1114 114 114 13.522353 7400000 3.55E-6				
2 start ⊇z wedw	• E)PC alac 12	urithed.bn	C 2 Microsof •	To MP20404	Single Coa	D 1 14975	C C C C C C C C C C C C C C C C C C C

Figure 9: Operator GUI.

The front panel shows the vacuum information as received from the coordinator. It displays the actual voltage signal as collected and transmitted by the end node. Fig. 10 shows the block diagram of the program.



Figure 10: Block diagram of GUI program.

CONCLUSIONS

The present network has been tested with the vacuum pumping modules kept at a distance of 150 ft. in a closed, non line of sight environment of SCC (Superconducting Cyclotron) Highbay area. The transmitted data has been checked with local vacuum gauge reading in an un interrupted manner and found working satisfactorily. As extension of the present job, we are engaged in the Design of 802.15.4 / ZigBee based Personal Area Network for data acquisition of environmental temperature, relative humidity, noise, water leakage from the inaccessible areas of Cyclotron Vault, Pit, Basement and ECR Highbay for the ease of fast fault finding and decrease of down time of both Cyclotron and ECR Ion source.

REFERENCES

- [1] ZigBee_Specification,
- http://www.zigbee.org/en/spec_download/download_request.asp
- [2] IEEE 2003 version of 802.15.4 MAC & Phy standard.
- [3] Data Sheet Microchip PIC18F46J50 MCU.
- [4] Data Sheet Microchip MRF24J40, RF Transceiver
- [5] National Instruments LabVIEW, http://www.ni.com

Socket-CAN DEVICE SUPPORT FOR EPICS IOCS*

C. Burandt[†], U. Bonnes, J. Enders, M. Konrad, N. Pietralla, Institut für Kernphysik, TU Darmstadt, 64289 Darmstadt, Germany

Abstract

This contribution describes an EPICS device support for CAN bus. It makes use of the Socket-CAN framework and is thereby independent from the API of a specific vendor. The device support has been used successfully in a production environment at the superconducting Darmstadt linear electron accelerator (S-DALINAC) since almost two years.

INTRODUCTION

CAN bus (Controller Area Network) has been chosen as the preferred field bus connecting in-house developed hardware to the S-DALINAC's accelerator control system. CAN bus is a rather robust communication bus. It allows a bit rate of 1 Mbit/s for distances up to 40 m. The underlying protocol is message-based. Every frame carries up to 8 byte of user data.

CAN interface controllers for personal computers are commercially available from different manufacturers, but although they all share the same basic functionality, most of them have a vendor-specific application programming interface (API). Moreover traditional CAN drivers are usually accessed by only one process at a time, which precludes the use of sniffer programs for debugging. In contrast to that the Socket-CAN network stack [1], included in recent Linux kernels, provides access to the CAN bus via network devices (BSD sockets). Those can be accessed by multiple applications at the same time via a vendorindependent interface. A set of open source CAN drivers provides access to controllers of different vendors.

For development purposes we use USB interfaces. They are easy to transport and can therefore be utilized to hook a laptop into a CAN bus segment for on-site diagnostics. The Linux computers running the IOCs (input output controller) are equipped with PCI/PCI express slot cards.

Since the S-DALINAC's accelerator control system is currently being migrated to EPICS [2], a CAN device support module has been developed to provide access to our devices from EPICS IOCs.

SOCKET-CAN

Opposed to a *character device driver* based *device support*, our solution here described relies on the Socket-CAN framework. The latter is included in the kernel main line since Linux kernel version 2.6.25, which is exceeded even by most of the conservative Linux distributions. On the



Figure 1: The Socket-CAN framework presents itself in the same fashion as the Linux kernel's network support. Different devices can be accessed by different applications at the same time. This is achieved by the network layer and appropriate protocol families.

one hand it brings along a growing number of CAN device drivers, on the other hand a network layer is implemented, which allows CAN interfaces to be treated in a similar fashion to ethernet devices [3]. The connection endpoint on the PC is represented by the Linux kernel as a BSD socket. The described functionality is combined in a so-called protocol family named PF_CAN. It is a common analogue of the PF_INET protocol family, which allows ethernet interfaces to be accessed with protocols like TCP and UDP. Both protocol families are provided by the kernel simultaneously. Figure 1 shows this architecture.

According to this concept, a CAN adapter does not just show up as character device /dev/canX, but needs to be brought up by

ip link set can0 up type can bitrate 1000000

as usual with ethernet devices. A virtual CAN device can be configured easily using the vcan kernel module. This is useful when doing IOC development without a physical CAN interface.

A central feature of the Socket-CAN framework is to reflect its bus property on the PC. While in hardware several participants can be connected to a CAN bus segment, the same should be possible on the PC. The Socket-CAN network stack makes this possible. Therefore, multiple applications, *e. g.* several IOCs and diagnostic tools, can be run

^{*} Supported by DFG through CRC 634.

[†] burandt@ikp.tu-darmstadt.de



Figure 2: The device support keeps two queues. One is used by the writer thread as a buffer. The other one tracks the time which has passed since the last arrival of an incoming frame to allow a second thread to identify timeouts. The reader thread maps incoming frames to the different records. It uses an allocation table for this task.

on one PC and can use a single interface to connect to a CAN segment.

DESIGN OVERVIEW

Figure 2 shows the structure of the device support module. The detailed design of the device support depends on the capabilities of the different devices which will be connected to the CAN bus. Asynchronous communication with the devices is possible. Nevertheless, we decided to use polling, that is requesting all data periodically. The repetitive transmission makes sure the information is never totally outdated or changed unnoticed by the control system. Polling allows broken devices, which do not respond anymore and that cannot even signal errors, to be identified quickly. A timeout can be triggered when a certain device does not send frames anymore. A feature shared by many of the in-house developed devices is the ability to transmit certain values periodically without the need for repeated requests. This makes the polling with timeouts approach even more convenient.

Since CAN frames may carry more than one piece of information, the device support needs to be able to distribute them to multiple records. A record table holding the mapping between the CAN address IDs and the corresponding input records is created during the IOC startup process.

All records sensitive to timeouts are stored in a separate table. Writing to the bus is handled by a queue. It holds all the frames intended to be sent on the bus. Multithreading allows each table/queue to be processed with equal priority.

DEVICE SUPPORT MODULE

Software Engineering Features

The Socket-CAN device support has been written in plain C. About 1300 lines of code are necessary to interface the Linux kernel and to deliver the data to different types of EPICS records.

Two build dependencies exist apart from some standard C library header files and from EPICS header files. The headers, describing the Linux kernel's Socket-CAN network stack, are needed. This is nowadays trivial, since it is part of the kernel main line. The second dependency concerns the firmware embedded in all devices that are equipped with a CAN interface. This header file defines the addressing scheme which is used for sending messages to specific devices.

There is no need for any include file specific to the CAN interface manufacturer. Therefore every CAN interface which is supported by the Linux kernel's socket-CAN framework can be used with this device support.

The IOC has to supply the name of the interface which is supposed to be used. During IOC initialization the INP fields of all records using Socket-CAN device support will be parsed. Every interface listed in one of these fields is being connected to and listened for incoming frames.

On arrival of a matching frame the requested bytes are extracted from the data frame and interpreted within the meaning of C data types. Subsequently they are handed to the record support for further conversion.

User Features

The Socket-CAN device support currently supports the basic record types:

- analog in/out
- binary in/out
- multi binary in/out
- multi binary direct in/out
- long in/out

There was no need for the more complex record types yet, but in principle these could be implemented also.

Template Development

The following is a showcase for a definition of a record which uses the Socket-CAN device support:

```
record(ai, "rf:rdSupplyVoltADC") {
  field(LINR, "SLOPE")
  field(ESLO, "9.3e-9")
  field(EOFF, "0")
  field(EGU, "V")
  field(EGU, "I/O Intr")
  field(DTYP, "tudSocketCan")
  field(INP, "@can0 07 01 06 519 02 1 sl 50")
}
```

The field DTYP makes the record use our Socket-CAN device support. To define a valid INP field, the template designer needs to collect certain information about the device the record is meant to communicate with. The line starts with the interface which is going to be used (can0). The further properties are as follows:

- 07: marks the CAN frame as being sent from a device (as opposed to frames being sent to a device)
- 01 06: address of the device
- 519: defines the specific request, in this case some ADC's value is being transmitted
- 02: specifies the ADC (there are more than one)
- 1: omit the first data byte, when converting the value, since this is the one used to identify the ADC
- sl: interpret as a signed long number
- 50: timeout in seconds

Usually the address is defined as a macro configured in a substitutions file. The number which specifies the request (519 in this example) is taken from a header file belonging to the firmware running on the devices. The user can use mnemonics to specify the request, and a perl script will substitute those automatically with their numerical code.

Socket-CAN Tools

A significant advantage of the Socket-CAN approach is testability. The Socket-CAN project provides a set of basic command-line tools. The following tools have proven as very convenient during template development:

- cansend: send a single CAN frame to the specified interface. Address and data content are given in hex notation and can be arbitrarily chosen.
- candump: dump the traffic of one or many CAN interfaces to stdout. Precise timing information can also be obtained. Filtering can be achieved by specific command-line options or piping to a grep command.

Considering that the CAN bus uses a message-based protocol, one can easily monitor the traffic between the IOC and the hardware. Utilizing caput and camonitor additionally, the whole chain from the CAN bus to the operator interface can be evaluated on a minimal setup.

FUTURE WORK

The current implementation has been used successfully at the S-DALINAC for almost two years. Still some minor issues remain unresolved.

The architecture of an EPICS IOC with its scanning implementation can lead to bursts of CAN frames sent to the microcontrollers. These can cause overruns of the receive buffers of the devices. Since dealing with such an issue needs changes deep inside the device support, a complete rewrite in the C++ programming language is being considered. The object-orientated design of this language plus a large set of functionality provided by the Standard Template Library (STL) would allow for a much cleaner design, from a software engineer's point of view. The table which holds the connecting information between the EPICS records and the corresponding CAN messages is currently implemented as a static array. During a rewrite this would be replaced by a dynamic data structure which consumes only as much memory as needed. Additional functionality could also be implemented, for example the logging of bus error frames to the IOC log or to a certain record.

The extension of the supported record types is possible. Some of our hardware components can send or receive strings. Thus string in, respectively string out, device support is needed. As long strings can require more than one CAN frame to be transmitted, this proves to be different from the already implemented records. The same applies to waveform records which could be used to transfer larger binary data sets.

CONCLUSION

The presented device support has been in use at the S-DALINAC for nearly two years. In particular the low-level rf control system relied on this module [4]. Since then it proved to be quite suitable for the given task. However the effort of a rewrite may be reasonable.

The characteristics of the device firmware are met well and allow an easy definition of large numbers of records, that share a pattern in nomenclature and the same firmware request.

The concept is simple enough to allow studends to write template files for yet unsupported firmware features after a short time of familiarization.

ACKNOWLEDGMENT

Important advice for a successful implementation of the Socket-CAN device support from members of the controlsystem group at BESSY is gratefully acknowledged. This work has been supported by DFG through CRC 634.

REFERENCES

- [1] The Socket-CAN project at berliOS.de http://developer.berlios.de/projects/socketcan/
- [2] C. Burandt et al., "Status of the Migration of the S-DALINAC Accelerator Control System to EPICS," PCaPAC'12, these proceedings.
- [3] Readme file for the Controller Area Network Protocol Family http://www.kernel.org/doc/Documentation/ networking/can.txt
- [4] M. Konrad et al., "A digital base-band RF control system," ICALEPCS'11, Grenoble, October 2011, MOMMU012, p. 82, http://www.JACoW.org

STATUS OF THE MIGRATION OF THE S-DALINAC ACCELERATOR **CONTROL SYSTEM TO EPICS***

C. Burandt[†], U. Bonnes, J. Enders, F. Hug, M. Konrad, N. Pietralla Institut für Kernphysik, TU Darmstadt, 64289 Darmstadt, Germany

Abstract

The S-DALINAC has been in operation for more than two decades. The control system has ever since been an inhouse development. This contribution overviews the current status of the migration to a new EPICS-based control system. Several important subsystems have been adapted, but the process has not been completed yet. The general network infrastructure has been restructured in context of the ongoing migration and is also presented.

INTRODUCTION

The superconducting Darmstadt linear electron accelerator S-DALINAC was taken into full operation in 1991 [1]. Two separate control systems, one for beam transport and one for the rf controls, both based on VMEbus computers, were used. The former was upgraded once, with a VMEbus Single Board Computer equipped with a DEC Alpha CPU and running VxWorks [2]. The user interfaces had been developed on a OpenVMS cluster. Communication was carried out via a custom protocol over Ethernet. As the S-DALINAC is an accelerator exclusively developed by students, the life cycle of many subsystems is longer than the time of a full change of the staff. Moreover students can not be expected to have previous knowledge of the VMS operating system or VxWorks which made maintenance and further developments quite difficult.

When extension of the old control system became impossible, the development of a new control server was initiated. Since no real-time requirements for high level control need to be met, an architecture relying on standard PC hardware and TCP/IP was chosen for its low price and the good availability [3]. The resulting client/server control system is still in use. It controls most of the magnet power supplies and the movable scintillation screens used as an optical feedback for the operators. Due to its highly specific hardware, the rf control system still was stuck with the old computer hardware and software.

When the analog low-level rf control system was replaced by a modern digital solution in 2010 [4], a control system which could cope with the challenges of thousands of available parameters and diagnostic values was needed. For its Linux compatibility and its active community EPICS [5] was chosen. Already without writing program code it can be adapted to different needs, since the design allows extensive configuration of all aspects. Even the integration of proprietary hardware interfaces requires only

* Supported by DFG through CRC 634.

[†] burandt@ikp.tu-darmstadt.de

minimal programming effort, as the device support is well separated from the other parts of the system. The EPICS framework and most of the related tools are open-source software. There are no license issues at all.

After first on-site experience with the new control system the process of integrating more equipment unrelated to the rf control system started. A growing number of devices has ever since been integrated and existing singlepurpose computer programs diminished leaving an increasingly cleaner operator environment.

HARDWARE

CAN Power Supplies

The S-DALINAC is an electron LINAC which is designed to provide beams of energies up to 130 MeV and $60\,\mu$ A. A low number of dipole magnets require currents between 50 and 300 A. The lenses, quadrupole magnets and steerers can be supplied with small power supplies (\leq 10 A of output current) which therefore contribute to the total number of approximately 200 magnet power supplies by 95 %. For replacement, an in-house development was prefered for cost reasons. The question for the hardware interface could be approached freely. A communication bus allows to connect multiple devices with a single cable and a single connector to the PC. The CAN (Controller Area Network) bus was chosen for its simplicity and costefficiency. Microcontrollers whith built-in CAN controller are available. In contrast to ethernet-equipped controllers these are quite low-priced and need less configuration. Figure 1 shows the basic concept of the in-house developed hardware family.

Further Hardware Development

When different hardware was developed in-house later on, the basic design of the power supplies was adopted at its best. That is, not only do they share a CAN bus interface but even the same microcontrollers running the very same firmware. Even though the devices are fundamentally different in purpose, they share basic functionality like setting the power status. The firmware itself determines the type of hardware it is running on and adapts its behaviour accordingly. That is, the individual commands for each kind of hardware are provided.

Among others, the following hardware has been developed taking the same structure as basis:

• A multi-purpose measurement system which uses different modules to digitize voltages and currents, but



Figure 1: Schematic layout of the hardware. The CAN bus is connected to the crate controller, a dedicated unit which has administrative functionality only, and can be daisy chained to the next crate. Inside the crates, the bus is extended on the backplane to the single slots. Thus, the wiring is kept clear. A PC running an appropriate IOC communicates with the devices and provides an EPICS interface for remote clients via Ethernet.

also allows four-terminal sensing for precise resistance measurements.

- Controllers triggering the pneumatics driving the scintillating screens.
- The most complex type of hardware in this development line is the above-mentioned rf control system. The fast feedback control algorithm is implemented in an FPGA.

EPICS AT THE S-DALINAC

Custom Device Support

The described hardware is connected to the PC by a commercial CAN interface. A so-called device support software module connects it to EPICS.

Although CAN bus device support modules for cards of different vendors are available, a new module has been developed. The socket-based approach used with our IOCs (Input Output Controller) is described in more detail in a dedicated contribution [6]. The major advantage over other implementations is the possibility to connect the IOC to a certain CAN interface while still being able to connect diagnostic tools to the very same software interface. Apart from that, independence from the PC CAN adapter manufacturer is gained.

Another type of device support is required for the digital low level rf system. It allows bit level diagnostics at the full sampling rate of 1 MS/s and a resolution of 18 bit. A USB 2.0 interface is used to transfer the data to the PC. The corresponding device support process provides streaming data via TCP/IP to arbitrary clients. After reducing the data rate the data is also fed into EPICS records. These are used for monitoring by the operator and can also be archived.

Input Output Controller

Several IOCs are in use at this time. They are executed either on physical or virtual machines. The rf control system, the thermionic electron gun and the multi-purpose measurement system have completely been migrated to EPICS. Each system has a dedicated IOC.

The large dipole power supplies and many vacuum gauges are equipped with serial ports like RS-232. These are connected to commercial device servers that make the serial interfaces accessible over Ethernet. An IOC utilizing the StreamDevice device support [7] runs on a virtual machine.

The IOC configuration is obtained from a relational database. A PostgreSQL Database holds information about all accelerator-related devices. It is divided into tables, each containing the equipment of a single domain, e.g. *beam diagnostics* or *beam transport equipment*. The information includes the electrical connection data, the type and name of the associated magnet and the current limits. When building an IOC this information is retrieved by a Perl script and is used to create substitution files automatically.

User Interface

As graphical user interfaces (GUI) the Eclipse-based Control System Studio (CSS) is used. A custom product has not yet been created. We so far rely on css-nsls2, which offers the *Best Operator Interface Yet* (BOY) and the DataBrowser plugin [8]. Due to the fact that programming is unnecessary, Operator Interfaces (OPIs) are changed and optimized on a regular basis and many are created to simplify special activities like the calibration of rf systems.

In future, a custom product will be built from the source repository. It might be quite similar to css-nsls2, but will carry all the individual configuration, which is for example necessary to access the archived process variables.

For the controls of the magnet power supplies, operators prefer rotary knobs during beam optimization. The knobs used at the S-DALINAC combine the features of setting a value to a particular channel as well as switching between different devices. An appropriate EPICS client application is under development and will complete the migration of the magnet power supply controls.

SERVICES

EPICS Specific Services

An archiver solution with PostgreSQL back end on has been put into operation recently. It is described in detail elsewhere [9].

State machines are implemented using the EPICS Sequencer module [10]. Since the S-DALINAC uses rf superconductivity, a fine-tuning system is necessary. It is based on magnetostrictive elements which are mounted next to the cavities inside the cryostat. To avoid frozen magnetic fields in the cavities during the next cooldown, the tuners need to be degaussed before warming up the cryostat, when maintenance is scheduled. A state machine cycles the tuners power supplies in a convenient way on demand and keeps track of the usage of the tuners. Further applications of the sequencer are under development.

General Services

A build-server retrieves source code from a revision control system and provides many applications as Debian packages. Together with a Debian FAI server (Fully Automatic Installation [11]) the setup of ready-to-use machines is a matter of minutes. This technique is used on a regular basis when hardware is replaced or the installation procedure for a certain machine has changed considerably.

A MediaWiki is hosted for arbitrary user documentation. Electronics engineers and control-system programmers share various information with users and eventually update it while keeping the version history. This is especially helpful with rapidly-changing student personnel.

The documentation of bugs is achieved by a bugtracker. They can be categorized and assigned to a certain developer.

INFRASTRUCTURE

Network Layout

A subnet holds all devices and computers which are directly required for the operation of the accelerator. Therefore all IOCs are operated in this subnet. The control systems of the various experiments use a separate network segment. The exchange of information, held as EPICS process variables, between the subnets is required. Two *CA Gateways* [12] are in operation facilitating read-only access to process variables that are available in the accelerator's subnet from the experiments' subnet and vice versa. Read-only access is also possible from office computers.

To further improve the network security, the setup of a so-called demilitarized zone (dmz) is convenient. A number of services needs to be available in both, the accelerators subnet and the public network. This includes the database server, the build-server, the Fully Automated Installer and potentially a webOPI server. These will be placed in the dmz which is a separate network segment.

Virtualization

When no direct hardware access is needed a service can run on a virtual host. This applies to the above-mentioned Wiki software, the bugtracker, the build server, the Fully Automatic Installation server and the *CA Gateway*. Even the IOC which uses the StreamDevice device support is solely connected via Ethernet to the device servers. Thus it is running on a virtual machine.

FUTURE WORK

There is no appropriate handling of warnings and errors from the beam-line equipment up to now. An alarm system toolkit like BEAST is a possible solution here.

Scintillating screens can be lowered into the beam pipe to allow determination of beam position and beam shape. Both, the controllers switching the pneumatics and the video multiplexer switching the corresponding video signal, are not yet integrated into EPICS.

CONCLUSION

During two years of operation in a production environment at the S-DALINAC the EPICS based control system proved to be a stable solution. The progress of migration is still ongoing, but is approximately half-finished.

ACKNOWLEDGMENT

This work is supported by DFG through CRC 634.

REFERENCES

- A. Richter, in: Proceedings of European Particle Conference 1996, Barcelona, Spain (Institute of Physics, London, 1996), pp. 110-114
- [2] S. Richter, "Design and implementation of a new local control system at the S-DALINAC," Dissertation D17, TU Darmstadt, December 2000
- [3] O. Patalakha, "Design and implementation of a modular client/server control system for the S-DALINAC," Dissertation D17, TU Darmstadt, July 2006
- [4] M. Konrad et al., "Digital base-band rf control system for the superconducting Darmstadt electron linear accelerator," Phys. Rev. ST Accel. Beams 15, 052802 (2012)
- [5] Experimental Physics and Industrial Control System, http://www.aps.anl.gov/epics/
- [6] C. Burandt et al., "SocketCAN device support for EPICS IOCs", THPD13, Proceedings of PCaPAC'12, Kolkata, India.
- [7] EPICS StreamDevice, http://epics.web.psi.ch/software/streamdevice
- [8] Control System Studio, http://cs-studio.sourceforge.net
- [9] M. Konrad et al., "Control System Studio Archiver with PostgreSQL backend: optimizing performance and reliability for a production environment," WEPD03, Proceedings of PCaPAC'12, Kolkata, India.
- [10] EPICS sequencer, http://www-csr.bessy.de/ control/SoftDist/sequencer
- [11] Debian, Fully Automatic Installation, http://fai-project.org
- [12] EPICS: The Process Variable Gateway, http://www.aps. anl.gov/epics/extensions/gateway

MULTIPLE HIGH VOLTAGE POWER SUPPLY CONTROLS SOLUTION USING COMPACT, DISTRIBUTED ETHERNET BASED PC BOARDS AND LINUX/WINDOWS BASED GUIS

J. Antony, R. Kumar, R. Suman, S. Venkatramanan, P. Sugathan, IUAC, New Delhi, India

Abstract

Compact Ethernet based High voltage PC boards have been developed, tested and produced to use as an integrated HV power supply unit to generate and control voltages varying from 0 to 2000 V dc from any OS independent PC platform. The Neutron gamma array (NAND) project at IUAC will need distributed control of at least 120 such units over a private Local Area Network to bias detectors. These Power supplies are being made as five independent boxes, each box consisting of 24 such HV PC boards and they will be interconnected using network switches. Presently, a compact two layer board with the PICO make DC-DC HV converter mounted on PCB, put together in a group of 24 of them, have been built and fully tested. The advantage of such a system is that, it is easily expandable to a large number of power supplies with low cost, globally accessible, multiple users in a network can set or read any power supply value through an OS independent PC. Control GUI applications are developed using C, IUAC PCLI [2], Ot c++ etc. and have been successfully tested.

INTRODUCTION

A large array of neutron detectors, named NAND [1] is being developed as a nuclear physics experimental facility to use with the beam delivered by the booster LINAC at IUAC. NAND will consist of about 120 detectors in the final phase in which each detector will have organic scintillator liquid cell coupled to Photonics photomultiplier tube. The detectors need to be biased at a dc voltage somewhere below 2000 V. A large number of compact power supply boards developed indigenously (approx. 70), has combined analog & digital circuitry in each PC board which can be addressed with unique MAC (Media Access Control address) and IP (Internet Protocol) address so that each can be specifically selected at a time for read write operations over distributed LAN.

THE HARDWARE

The COTS design of the H/W board for the HV generation and control, using the DC-DC converter module, is given in Fig.1 below. The board used SMD components due to its compact size required to fit-in 24 of them in a single box. The digital section consists of an 8 bit microprocessor (ATMEGA168/328) and an SPI driven Ethernet controller (ENCJ60). The analog section consists of a precise 12 bit DAC (AD 7541) connected to

the microprocessor using the parallel bits. A precision Voltage reference of 2.5 V is used externally to generate 0 to 2.5 V at its output. This voltage is amplified using a low offset high precision single supply OPAMP, CA3140, to generate 0-5 V for the DC-DC conversion. The small size PICO make HVP2N is used to generate 0 to -2000 V from a 0 to +5 V input. RJ45 connectors are used for the Ethernet connections to outside world. The block diagram of the HV control unit is given in Fig. 2.



Figure 1: The single compact HV board with Ethernet.



Figure 2: The block diagram of a HV control unit.

THE FIRMWARE

The boards use a small TCP/IP firmware stack. The TCP is used as the HTTP works on TCP protocol. The idea is to build an easiest user interface which brings in HTML code from the web server board into the browser client to perform simple read write operations using a

small size single data packet TCP. The IP address is firmware coded and can be altered. Also one can program the boards to work as static or dynamic servers.

DISTRIBUTION OVER LAN



Figure 3: Networking architecture over LAN.

The interconnection of Ethernet based HV boards are done using a 24 port CISCO network switch as given in Fig. 3. Each board has its own MAC and IP; therefore it becomes unique in a network. The Ethernet based boards can be very useful for interconnecting using WiFi, which transforms any IEEE 802.3 device (wired network device) into an 802.11b/g/n wireless client. The inside view of complete 24 channel box is shown in Fig. 4.



Figure 4: The inside view of a 24 channel unit.

WINDOWS & LINUX CONTROL GUI

A high level socket program in c or c++ will allow programmer to easily connect to any HV power supply unit through TCP services by implementing a simple HTTP client which will get request a web page given the hostname and the page name, then read the server answer and output the HTML content of the reply which can $\stackrel{\bigcirc}{\sim}$ contain the HV read-out data. To be able to connect to a Service built on top of TCP, we first need to create a socket for the TCP protocol, fill in a network address structure representing our Power supply destination ip and the port to connect to. From there, we will be able to send and receive data over the network.

IUAC uses an in-house developed Linux-c based Client-sever based control system known as PCLI [2] for the entire control of pelletron parameters via. CAMAC for last 25 years. The server of PCLI now has been modified to test NAND power supplies to its hardware support by mapping all the High voltage units to C, N, A, F commands and has been tested successfully and is given below in Fig. 5.



Figure 5: The PCLI control page for NAND power supplies.

Ot is a cross-platform GUI development tool. A Ot application program has also been developed in C++ to control and monitor 24 such power supplies using the Network manager libraries and has been tested from windows & Linux PCs and the screen shot is given below in Fig. 6.



Figure 6: Qt interface for controls & readout.
SUMMARY

Out of many boards under fabrication, the first phase of 24 such boards housed as a single 19" mounted box has been tested successfully. The advantage of such a system is, it is easily expandable to a large number of power supplies at a low cost, globally accessible as they have unique addresses; multiple users in a network can set or read any power supply value simultaneously. The disadvantages being the real read back issues due to lack of negative voltage read back circuitries at the load farend.

ACKNOWLEDGMENT

The authors would like to acknowledge Mr. Ajith Kumar B.P of IUAC for his support to make it compatible to PCLI interface and the entire IUAC magnet lab support for their contributions to integrate it as a single hardware unit out of the 24 HV boards supplied to them.

- [1] http://www.iuac.res.in/refac/np/nand/nand main.html
- [2] B. P. Ajith Kumar, J. Kannaiyan, P. Sugathan and R. K. Bhowmik, "The NSC 16 MV tandem accelerator control system", Nuclear Instruments and Methods in Physics Research Section A, Volume 343, Issue 2-3, p. 327-330.

FAST DIGITAL FEED BACK CONTROL SYSTEMS FOR ACCELERATOR RF SYSTEM USING FPGA

Pritam Singh Bagduwal*, Dheeraj Sharma, Nitesh Tiwari, M. Lad, P.R. Hannurkar Raja Ramanna Centre For Advanced Technology, Indore, India

Abstract

Feedback control system plays important role for proper injection and acceleration of beam in particle accelerators by providing the required amplitude and phase stability of RF fields in accelerating structures. Advancement in the field of digital technology enables us to develop fast digital feedback control system for RF applications. Digital Low Level RF (LLRF) system offers the inherent advantages of Digital System like flexibility, adaptability, good repeatability and reduced long time drift errors compared to analog system.

To implement the feedback control algorithm, I/Q control scheme is used. By properly sampling the down converted IF signal using fast ADC we get accurate feedback signal and also eliminates the need of two separate detectors for amplitude and phase detection. Controller is implemented in Vertex-4 FPGA. Codes for control algorithms which controls the amplitude and phase in all four quadrants with good accuracy are written in the VHDL. I/Q modulator works as common actuator for both amplitude and phase correction. Synchronization between RF, LO and ADC clock is indispensable and has been achieved by deriving the clock and LO signal from RF signal itself. Control system has been successfully tested in lab with phase and amplitude stability better then $\pm 1\%$ and $\pm 1^{\circ}$ respectively. High frequency RF signal is down converted to IF using the super heterodyne technique. Super heterodyne principal not only brings the RF signal to the Low IF frequency at which it can be easily processed but also enables us to use the same hardware and software for other RF frequencies with some minor modification.

INTRODUCTION

A typical RF system of an accelerator consists of RF synthesizer, chain of amplifiers, RF cavity and other required sub systems. All these subsystems are realized using various active and passive components to fulfil the requirement for beam acceleration. Change in amplitude and phase characteristic of any of the component could lead to change in phase and amplitude of the EM field inside the cavity. To keep the amplitude and phase stable LLRF feedback control systems is used.

Block diagram of a typical Digital LLRF system is shown in the Fig. 1 which consists of four main components

- Synchronised LO and CLK generation
- RF to IF down conversion
- Digital controller
- Actuator

RF signal is first down converted to IF using mixer and appropriate LO signal. Synchronization between LO, RF and CLK is required for proper amplitude and phase detection. Controller is implemented in FPGA based digital signal processing board having fast ADC and DAC. The I/Q Modulator works as actuator to correct the amplitude and phase of RF signal. Digital signal processing schemes like CORDIC (COordinate Rotation DIgital Computer), PI (Proportional Integral) controller and digital filters are used to implement the proper algorithm. These schemes are implemented using Vertex-4 FPGA.



Figure 1: Block Diagram of Digital Feed Back Control System.

*pritam@rrcat.gov.in

SENSING AND DETECTION

Performance of the feedback control system strongly depends on the accuracy of the detection, hence proper sensing and detection of the amplitude and phase information of the RF signal is very important. As the processing of RF signal at very high frequency is difficult, super heterodyne scheme is used where the high frequency signal is down converted in to the Intermediate Frequency (IF) preserving the amplitude and phase information. This scheme allows us to use same system for different frequency with only minor modification.

Selection of IF frequency is done considering the maximum expected error rate and available hardware. We have chosen IF to be 31.6MHz for 505.8 MHz RF signal. Amplitude and phase information from the down converted IF signal is extracted using digital I/Q detection scheme. In Digital I/Q detection scheme two consecutive samples are taken 90° apart and separation of alternative samples gives us I and Q signal. This eliminates the need of two separate detectors for amplitude and phase detection. Selection of sampling frequency (f_s) is critical; which is governed by the relation shown in the Figure 2.



Figure 2: I/Q detection scheme.

Taking clock jitter, maximum error rate and available resources in consideration, a sampling frequency of ~25 MHz is selected.

SYNCHRONIZED LO AND CLOCK GENERATION

Synchronised RF, LO and clock signals were used to detect amplitude and phase of RF signal with good accuracy. To achieve this synchronization both LO and clock are derived from the main RF signal. LO (505.8+31.6 MHz) signal is required for Up/Down conversion of RF signal. Clock (25 MHz) signal is required for operation of ADC/ DAC. Using a pre-scalar 31.6 MHz (505.8/16) signal is produced which when mixed with 505.8 MHz RF signal gives the required LO signal. The same 31.6 MHz signal is used to generate the 25 MHz clock using DCM. This makes the LO and Clock both synchronised with RF signal. Performance of this scheme was tested by applying known phase shift to the RF signal and detecting this phase shift. Test results shown in Fig. 3 depict close resemblance between applied and detected phase shift.



Figure 3: Phase detection characterization.

CONTROLLER

Purpose of the controller is to generate the control signal on the basis of set and detected signal. Controller is implemented in the vertex-4 FPGA board with 100 MSPs ADCs and DACs. Flow diagram of digital controller is shown in the Fig. 4



Figure 4: Controller flow chart.

Input IF signal is sampled by the ADC at a rate of \sim 25MHz to give the 'I' and 'Q' components of the IF signal. These sensed 'I' and 'Q' components of the IF signal are compared with the set 'I' and 'Q' components. Control algorithm is written in VHDL to generate the required I and Q control signal. Single ended to differential converter is also implemented to make I/Q control signal compatible to I/Q modulator. These differential 'I' and 'Q' control signals are fed to I/Q modulator to generate the corrected RF output signal. To rotate the control signal in the circle for producing correction CORDIC algorithm is used. Movement of the control vectors are shown in the Fig. 5



Figure 5: Rotation of vectors for control system.

I/Q MODULATOR

In order to correct the amplitude and phase error actuator is needed, which should be able to change the

amplitude and phase as per the control signal. I/O modulator is used to control the amplitude and phase.



Figure 6: Typical Block diagram of I/Q modulator.

As depicted in Fig. 6 change in base band I and Q signal will change the amplitude and phase of the RF signal allowing the control of amplitude and phase of the RF signal. These base band I/Q signals are generated by the controller to correct the error in the signal.

TEST SET UP AND RESULTS

For testing of the digital LLRF control system test bench was set up in the lab. The block diagram of test setup is shown in Fig. 7.



Figure 7: Block diagram of the test setup.

Amplitude and phase errors are deliberately introduced and performance of the system is checked under closed loop and open loop condition. Amplitude correction for the 20dB dynamic range and phase correction of the full 360 degree has been successfully achieved. Results along with the photograph of actual test bench are shown in the Fig. 8.

Test on the amplitude and phase stability of the system > was also performed. Amplitude stability of better than 0.5% and phase stability better than 0.5° has been achieved. Results are shown in Fig. 9.



Figure 8: Photograph of test bench and test results.



Figure 9: Amplitude and Phase Stability Results.

CONCLUSION

Digital Low Level RF control system has been developed. Testing of the closed loop system has been successfully done in the lab. This Digital LLRF system will be implemented in Indus-2 RF system which is being upgraded. With slight modification same system can be used for machine at different frequency. Use of digital processor like FPGA will also allow us to incorporate different diagnostic for RF system in future.

ACKNOWLEDGMENTS

We acknowledge Shri S.J. Buhari, Shri N. Srinivas, Shri Arnab Das, Shri Ashif Reza and Shri Joy deep Jana for assistance in this work.

- [1] Mahendra Lad, Nitesh Tiwari, Pritam S. Bagduwal and P.R. Hannurkar, "Commissioning & Optimization of Indus-2 RF System for 2GeV/100mA", InPAC-11, IUAC, New Delhi, 2011.
- [2] Lawrence Doolittle, "Low-Level RF Control System Design and Architecture", APAC, RRCAT, Indore, India, 2007.

API MANAGER IMPLEMENTATION AND ITS USE FOR INDUS ACCELERATOR CONTROL

B. N. Merh[#], R. K. Agrawal, K. Barpande, P. Fatnani, C. P. Navathe, RRCAT, Indore, India

Abstract

The control system software needed for operation of Indus accelerators is coupled to the underlying firmware and hardware of the control system by the Application Programming Interface (API) manager. In the threelayered architecture of Indus control system, PVSS-II SCADA is being used at the layer-1(L1) for control and monitoring of various sub-systems. The layer-2(L2) consists of VME bus based system. The API manager plays a crucial role in interfacing the L1 and L2 of the control system. It has to interact with both the PVSS database and the L2. In order to access the PVSS database it uses the PVSS API, a C++ class library, whereas in order to access the L2 custom functions have been built. Several other custom functionalities have also been implemented. The paper presents the important aspects of the API manager like its implementation, its interface mechanism to the lower layer and features like configurability, reusable classes. multithreading capability etc.

INTRODUCTION

PVSS-II [1] has a highly modular structure. Various functionalities are handled by modules specifically created for different tasks. These modules are called *managers*. The Database (DB) and Event (EV) manager are the core managers that handle and manage all process variables of the system.



Figure 1: Software Layers of Indus-2 Control System.

#bhavna@rrcat.gov.in

The software layers of Indus-2 control system are as shown in Fig.1. PVSS system works at user Interface (UI or L1) Layer. The lower layers are Supervisory Control (SC or L2) and Equipment Controller (EC or L3) layers. The API manager is interfaced to the SC layer over Ethernet.

API MANAGER

What is API Manager?

PVSS offers a C++ application interface which enables it to extend its control functionality [2]. This interface allows the developer to implement his own custom functions together with full access to PVSS database. The self contained manager so implemented is called the API manager. It is also an interface for the integration of external programs [2]. Any software can be integrated in a PVSS System via class libraries provided by PVSS API. Implementation of these managers has been done for Indus-2 controls specific tasks. API managers have been developed for all sub-systems of Indus-2 viz. Magnet Power supply (MPS), RF, Beam Diagnostics (BDS), Timing (TCS), Vacuum (VCS), Radiation Safety (RSS), Machine safety and Interlock (MSIS), Beam line Frontend (BLFE) and Beam Orbit Correction.

Internal Structure of API

The API manager communicates with EV and DB



Figure 2: Event or Message handling by API.

using messages [2]. The manager processes incoming messages in the doReceive() function. doReceive() is automatically called by the *dispatch()* function for each incoming message. The *dispatch()* function is called at regular interval in the API manager.

Each message is composed of group(s) (DpMsgAnswer, DpHLGroup class) of DpVCItem (Data Point(DP) value changes) [3].

Fig. 2 shows the occurrence of events and data flow between EV, API Manager and L2.

As in Fig. 3 the DpMsgAnswer class object is returned if a request for DP values is made, whereas DpHLGroup object is returned on any spontaneous DP event.

In case of most functions regarding the access to data points, messages are handled with the aid of WaitForAnswer objects [4].

So, in order to receive notification for any DP event HotlinkWaitForAnswer is specified as:

Static PVSSboolean dpConnect(const DpIdentifier &dpId, WaitForAnswer *wait, PVSSboolean del = PVSS TRUE); [4]



Figure 3: DP Message Handling in API.

Indus-2 API Manager Functions

When the API manager is first started it executes its initialization routines.

- *Load device configuration information* Loads device related information like signal name, L3 card number, channel number and corresponding Data Point information from the configuration file.
- Interface to PVSS DB Connect to data and event manager and for each data point get the identifier.
- *Establish connection to L2* The API manager interfaces to L2 over custom TCP/IP application layer protocol.

Once initialization process is run successfully, the API manager proceeds further with its routine tasks as follows:

• **Polling** - It periodically polls the L2 for status and current operating values of the various devices of a sub-system. The polling is done at 1 Hz. The polling command is sent on receiving a hotlink (Fig. 2) from the global timer which is used system-wide to

maintain synchronization among all the API mangers.

- **Data filtering** The API manger after parsing the data, does the comparison between old and new values of incoming data from L2 and sets only the changes in order to reduce the network traffic and PVSS processing.
- **Data transformation** Data is first transformed from protocol specific type to basic C++ types and then to PVSS data type.
- *Conversion* All incoming data values from L2 are scaled to their specified ranges.
- *Handle control commands* Any device settings command or control command is received by the API manager and a command is framed according to the decided protocol. This is sent to L2 (Fig. 2) and acknowledgement received.
- *Event log* All events like control commands, set commands, error received from L2, communication break from L2, operator actions from GUI are logged in the event log file in chronological order. The event log file is periodically renewed after it reaches it size limit.

Indus-2 API Manager Features

• *Object oriented approach* – The object oriented programming of the API allows making full use of its many benefits like maintainability, reusability, extensibility etc. The code is easy to understand and manage. In Indus control system API managers the TCP class, the timer class, the event log class all are reusable classes used system–wide for all API managers. This facilitates quick development. Fig. 4 illustrates the class diagram of API manager for magnet power supply system.



Figure 4: Class Diagram of API.

• *Easily configurable* - At the start-up the API manager loads information from the configuration files. These files have device related information viz. signals coming from the device, the physical

mapping of signal by card type, card number and channel number, the data point mapping of the signal by DP name. Thus any addition/modification /deletion of any signal need not require any change in API code. The manager is re-run to load a new configuration.

- *Multi-threading to publish data* The API manager also serves the programs external to PVSS system with required data like beam current, beam energy, beam position from all BPIs etc. It caters to multiple clients by following a multi-threaded design.
- *State Based* For a special requirement like ramping the magnet power supplies, API manager code was implemented such that it retains the state of the last operation. The various states maintained are INIT, RAMP-ON, RAMP-PAUSE, RAMP-RESUME and RAMP-OVER. So even if the API Manager is rerun during ramping process, it will not affect the clock generation and ramp operation.
- **Data Refresh on request** The data in the respective Data Points is usually set from the API manager only on change. All changes are reflected on the operator panel. At times it is required to refresh data which effectively will set the same data in DP but renew its timestamp. At any instance the API sets/refreshes all data on request from the operator.
- *Error handling* In case of error in communication between L1 and L2, the API manager disconnects from the global timer which stops polling. It then tries to periodically establish connection to L2.
- **Diagnostics** The API manager provides different diagnostic information like, the status of L3 stations, status of connection between L1 and L2, status of any control command sent from the operator interface and sequence of actions being taken.

Table 1 illustrates the DP load handled by various API manager of Indus-1 (I1) and Indus-2 (I2) control system.

System Name	No. of Devices (approx.)	Total DP handled by API (approx.)
I2- MPS	174	6960
I2-RF	25	320
I2-VCS	220	1350
I2-TCS	10	145
I2-BDS	80	235
I2-RSSS	81	378
I2-BLFE	270	756
I2-MSIS	-	165
I1-MPS	97	112

Table 1: Total DP Load handled by API managers.

CONCLUSION

The Indus-2 API managers have been developed and commissioned in 2005. Since then augmentations and new additions have been made. All features and functionalities mentioned in the paper have been implemented all through these years. API managers have been running with no crash event being reported. The load of the over all system has been nearly constant and lies between 17-21% with API manager load is maximum 2%. Possible extensions to the API are developing a generic API which caters to all future up-gradations of the Indus control system.

- [1] PVSS-II is a SCADA package from ETM, Austria.
- [2] PVSS Driver Development by ETM.
- [3] H. Milcent, P. Burkimsher, W. Salter, "How Do I Write a Driver? Ctrl Managers, API Managers and Drivers", Release 2.1, 2003.
- [4] PVSS-II API Documentation by ETM.

ADAPTIVE FUZZY CONTROL FOR TRANSFER CHANNELS IN PARTICLE ACCELERATORS

S. Berlik, University of Siegen, Siegen, Germany H. Ehrlichmann, DESY, Hamburg, Germany

Abstract

Long-term objective of this work is to develop a fuzzy technology based control framework to be applied in particle accelerators. Main motivation for this is the promise of fuzzy systems to exploit the tolerance for imprecision, un-certainty, and partial truth to achieve tractability, robustness, and low solution cost. Intended areas of application are manifold: we think on automatic operation, optimization of the operating conditions and yields; applied to various stages in the processing of circular and linear accelerators. As a first step towards this goal a fuzzy control system for a transfer channel in a particle accelerator has been developed. For it we built up the machinery, i.e. algorithms, data structures, integration in the existing control system and did a first proof-of-concept. Special emphasis is given on handling high dimensional data streams and the immanent challenges as sparsity and equidistance of the data.

INTRODUCTION

Model-based techniques for process control can provide valuable advantage over conventional approaches. They may yield a better understanding of the underlying system behavior than pure black box models and offer the opportunity to directly include expert knowledge. Such models, however, are costly to build and maintain when manually designed. Furthermore, they usually need to be continuously adapted since process conditions may change dynamically. Because of that evolving data-driven models with automatic adaptation techniques are becoming more and more attractive since they reduce maintenance costs while keeping high precision and interpretability.

Such a system that learns and evolves with every single sample from the data stream shall be developed for the use in a particle accelerator. A special problem here is the high dimensionality and homogeneity of the incoming data. For it, an additional projection step is implemented to reduce the dimensionality and filter out the most interesting ones.

Key ideas of the algorithm and the state of the work in progress are to be presented in the following. First we outline the current area of application within the DESY particle accelerator complex and sketch the algorithm's overall structure. In the following sections essentials of adaptive, i.e. evolving fuzzy systems are introduced and the projection mechanism is presented.

AREA OF APPLICATION

As a sandbox for its initial release the 450 MeV electron/positron transfer line between the accelerators PIA and DESY II within the pre-accelerator chain for PETRA III has been chosen. This decision was motivated by the ideal



Figure 1: Particle accelerators at DESY.

test condition, since tests can be carried out parasitically to the regular operation at a quite high beam transfer frequency of 6 Hz. For a short overview over the DESY facility (Fig. 1) and its accelerators see [1]. The fuzzy control algorithms were integrated into the TINE-based accelerator control system via MATLAB codes.

OVERALL STRUCTURE OF THE CONTROL ALGORITHM

A modern method for the control of complex processes is model predictive control (MPC), also referred to as receding horizon control (RHC)[2, 3]. Model predictive control uses a time-discrete dynamic model of the process to be controlled to calculate its future states and output values as a function of the input signals. Using this prediction, in particular suitable input signals for desired output values can be found. While the model behavior will be predicated several steps ahead till a certain time frame, the input signal is usually only searched for the next time step and then the optimization is repeated. For the calculations of the next time step then the actual measured state is used, resulting in a feedback and thus a closed loop. Model predictive control technology offers a number of advantages that have made it one of the most widely used advanced control methods in the process industry. It can calculate control variables where classical nonlinear control techniques fail, is easily extended to multivariable problems, can take restrictions on the actuators into account, permits the operation near the constraints, allows a flexible specification of the objective function, delivers optimum control devices and is last not least model based. A typical controller structure to build a



Figure 2: Internal model control scheme.

model predictive controller is shown in Figure 2.

EVOLVING TAKAGI-SUGENO FUZZY MODEL

In the present work, the model shown in Figure 2 is a fuzzy one. Fuzzy set theory provides structured ways of handling uncertainties, ambiguities, and contradictions which made systems based on fuzzy set theory the approach of choice in many situations. Since its introduction in 1965 [4], fuzzy set theory has found applications in a wide variety of disciplines. Modeling and control of dynamic systems belong to the fields in which fuzzy set techniques have received considerable attention, not only from the scientific community but also from industry. Their effectiveness together with their ease of use compared to systems based on classical two-valued logic paved the way to countless practical applications of fuzzy systems.

Compared with other control strategies, e.g. based on neural networks, the use of fuzzy models offers significant advantages: Already existing expertise can be directly fed into the system, in contrast to neural networks the knowledge is explicitly, thus self-explanatory and the acceptance of the procedure higher.

The modeling and control of nonlinear systems using fuzzy concepts is described in [5]. Current methods for identification are data-extraction techniques on the one hand and expertise on the other, but also mixed forms of both approaches. Basis of the data-driven approach is a clustering algorithm whereby fuzzy models are derived, preferably of the Takagi-Sugeno (TS) type [6].

A Takagi-Sugeno fuzzy model is characterized by its crisp, most often linear functions in the consequent part. Thus, it combines linguistic and mathematical regression modeling in one. The antecedents describe fuzzy regions in the input space in which the consequent functions are valid. The TS rules take the following form:

$$R_i: \text{ If } \vec{x} \text{ is } A_i \text{ then } \hat{y}_i = f_i(\vec{x}); \qquad i = 1, \dots, C.$$
(1)

Formally, the underlying Takagi-Sugeno fuzzy system with its multiple input variables (x_1, \ldots, x_p) , one single output variable y, and C rules can be defined as follows:

$$\hat{\mathbf{f}}(\vec{x}) = \hat{y} = \sum_{i=1}^{C} l_i \Psi_i(\vec{x})$$
 (2)

with normalized Gaussian membership functions

$$\Psi_{i}(\vec{x}) = \frac{\exp\left(-\frac{1}{2}\sum_{j=1}^{p}\frac{(x_{j}-c_{ij})^{2}}{\sigma_{ij}^{2}}\right)}{\sum_{k=1}^{C}\exp\left(-\frac{1}{2}\sum_{j=1}^{p}\frac{(x_{j}-c_{kj})^{2}}{\sigma_{kj}^{2}}\right)}$$
(3)

and consequent functions

$$l_i = w_{i0} + w_{i1}x_1 + \dots + w_{ip}x_p, \tag{4}$$

where x_j denotes the *j*th input variable, c_{ij} the center, and σ_{ij} the width of the Gaussian fuzzy set in the *j*th premise part of the *i*th rule.

In principal, learning in this setting can be implemented in two places: a) fuzzy-rule-based structure design and b) parameter identification of the rules consequents. The rule base may evolve in the number of rules C and the number of fuzzy sets per input dimension, each premise in turn in its center c_{ij} and width σ_{ij} . Consequent parameter estimation is realized via the output weights w_{i0}, \ldots, w_{ip} .

Evolving Fuzzy Model

As already mentioned, the fuzzy model in the kernel of the controller shall evolve over time as data samples arrive from the data stream. Several algorithms have been proposed for this use case, e.g. the FLEXFIS model [7] or the eSensor approach [8]. Generally speaking - while varying in details - all of these approaches follow the same main idea to partition the learning problem into the subproblems structure identification and consequent-parameter estimation. The first subproblem is usually approached using some clustering technique in the data space. This partitioning creates basic information granules, described linguistically by fuzzy sets, that transform the raw data into primitive forms of knowledge. For the second subproblem typically some form of weighted recursive least squares algorithm (WRLS) is applied to estimate the output weights per rule. In general, our implementation is inspired by the eSensor approach with some major modifications in the data preprocessing phase (see next Section).

As can be seen in Figure 3 the fuzzy rule base of the controller works in two main modes, the calibration and estimation mode depending on the kind of data sent to the model. If the input data comes with measured output data the evolving fuzzy model uses both to recalibrate the actual model. In case of input data coming alone, the output is estimated by the fuzzy model.

PROJECTED CLUSTERING OF HIGH DIMENSIONAL DATA

The input data stream of the given problem consists of measurements of the several devices in the accelerator and is hence high-dimensional in nature. High-dimensional data however is inherently more complex in clustering, classification, and similarity search. Among others this is because of the sparsity of the data in the high-dimensional case. Moreover, in high-dimensional space, all pairs of



Figure 3: Calibration and estimation flowchart of the evolving Takagi-Sugeno fuzzy model.

points tend to be almost equidistant from one another. This makes it difficult for common distance-based clustering algorithms to find meaningful clusters.

An algorithm tailored to this domain is the HPStream algorithm [9]. It implements a high-dimensional projected stream clustering by continuous refinement of the set of projected dimensions during the progression of the stream. Dimensions to be projected are selected depending on the variance of the data in it. Those with the smallest variance are chosen. The updating of the set of dimensions associated with each cluster is performed in such a way that the points and dimensions associated with each cluster can effectively evolve over time.

We use a methodology inspired by the HPStream algorithm to preprocess the high dimensional data stream. It differs in one essential way from its model in that it prefers dimensions with high to small variance. This is due to the rather unconventional environment the algorithm is used in. Measurements are in most dimensions more or less constant. Since the original projection mechanism clusters data samples with the highest similarity it would mask out the few dimensions in which variations happen. The modification aims at finding the dimensions with high 'liveliness'.

CURRENT STATUS

Currently the core algorithms with its data structures have been developed and integrated in the existing control system. Critical turned out to be the model building due to the special nature of the data. With the eSensor concept no meaningful model could be achieved. Either only one single cluster was created for all incoming data samples or every sample made up a new own cluster. Transition between both states turned out to be extreme sharp. Configuring the algorithms parameter such that a meaningful number of clusters is maintained was not possible.

Because of this the projection mechanism has been integrated. With it a configurable number of fuzzy clusters can be supported. The problem than has shifted to the poor extrapolation quality of the model. The optimization algorithm that evaluates the model in order to find the next control variables is miss leaded by the linear consequent functions to the borders of the domain. For this reason the current work focuses on the integration of a confidence factor in the blending procedure of the consequent functions.

CONCLUSION

Presented has been an architecture that addresses the design of an adaptive fuzzy model predictive control system, comprising a data stream driven evolving fuzzy model with data projection mechanism to reduce the input dimensionality and an optimization component to find optimal control quantities.

ACKNOWLEDGMENT

We gratefully acknowledge funding from Deutsches Elektronen-Synchrotron, Hamburg (DESY).

- [1] SPEED MACHINES. Deutsches Elektronen-Synchrotron DESY 2010. http://pr.desy.de/sites2009/site_ pr/content/e113/e48/column-objekt60802/lbox/ infoboxContent60803/SPEED_MACHINES_eng.pdf
- [2] Jan Maciejowski. *Predictive Control with Constraints*. Prentice Hall, June 2001.
- [3] Coleman Brosilow and Babu Joseph. *Techniques of Model-Based Control*. Prentice Hall, 2002.
- [4] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8:338– 353, 1965.
- [5] Robert Babuška. *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.
- [6] Tomohiro Takagi and Michio Sugeno. Fuzzy identifiation of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116– 132, February 1985.
- [7] Edwin David Lughofer. FLEXFIS: A robust incremental learning approach for evolving Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, 16(6):1393–1410, December 2008.
- [8] Plamen P. Angelov and Arthur Kordon. Adaptive inferential sensors based on evolving fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(2):529–539, 2010.
- [9] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for projected clustering of high dimensional data streams. In *Proceedings of the 30th VLDB Conference*, Toronto, Canada, 2004.

DRIVE SYSTEM CONTROL FOR KOLKATA SUPERCONDUCTING CYCLOTRON EXTRACTION SYSTEM

T. K. Bhattacharyya[#], T. Das, S. Bhattacharya, C. Nandi and G. Pal, VECC, Kolkata, India

Abstract

The K500 Superconducting Cyclotron at VECC, Kolkata uses two electrostatic deflectors, eight passive magnetic channels, one active magnetic channel and two compensating bars as its extraction elements. Except the active magnetic channel, all the other elements can be moved radially, typically by ± 6 mm around a centre position. This maneuverability is due to the fact that not all the ions, spanning the operating region of the cyclotron, will have the same optimum beam extraction radius. At the end of the beam extraction channel, the beam is shaped and aligned by a pair of water cooled slit. The slit movement is pneumatically controlled as it has to be operated in high magnetic field. The computer controlled drive system can move the elements precisely. The paper will describe the drive system and its control mechanism.

INTRODUCTION

The extraction system in superconducting cyclotron [1] is required to extract the beam from the cyclotron. The extraction system layout is shown in fig. 1.



Figure 1: Extraction system layout.

It is inherently difficult to extract beam because of the high magnetic field and small turn separation. The high magnetic field itself exerts a strong centripetal force that has to be encountered to deflect the beam out of the cyclotron [2]. Because of the very restricted space in the cyclotron, the deflectors must be built with utmost care with proper

#tamal@vecc.gov.in

surface finish to achieve a high electric field gradient. After deflecting the beam to suitable radius using electrostatic deflectors (Fig. 2), magnetic devices called magnetic channels (Fig. 3) can complete the task of deflection. Compensating bars have to be judiciously installed to compensate addition of magnetic material due to magnetic channels.





Beam dynamics demand that extraction components viz. electrostatic deflectors, magnetic channels and compensating bars should have accurate movement mechanism.



Figure 3: Magnetic channel.

SYSTEM DESCRIPTION

The magnetic channels are required to be moveable to suit extraction conditions of different beams. The movement mechanism has to be reproducible and precise. The drive system (Fig. 4) of a magnetic channel is consisting of a synchronous motor, gears, encoder, a lead screw and nut. The synchronous motor drives a lead screw and nut. The lead screw nut is fixed to the magnetic channel element using a stainless steel tube. At the vacuum barrier, edge welded bellow has been used. The motion of the lead screw is transmitted to an angular encoder, which gives the position of the extraction element.



Figure 4: Motor drive for magnetic channel M4.

General Layout

The layout of the control system is shown in fig. 5. The drive system for the magnetic channel is fully computer controlled. A geared synchronous motor precisely controls the position of the magnetic channel using hardware modules and dedicated HMI software, developed in-house. An absolute encoder is attached to the motor shaft for monitoring the channel position remotely.

The drive system is having the following features:

- All the magnetic channel and compensating bar drives can be operated both manually and remotely from the control console.
- Mechanical limit switches are placed to restrict the movement of the magnetic channels in addition to the predefined software limits.
- The maximum range of travel of the single drive is 15 mm with an accuracy of $\pm 2\%$ of full scale range with a resolution of 0.1 mm.
- Optical encoder is calibrated to provide the actual displacement of the drive. It provides digital output collected by individual electronics with local display developed at VECC. A centralised DAQ server communicates via RS-485 to collect all the position values and send them to the HMI on the console.
- Relays located in the drive control module (Fig. 6) are used for two purposes; firstly, to execute the interlocks for the limit switches and secondly, to facilitate

operation of the drives both from locally and remotely through control console.



Figure 5: Block diagram of drive system.

Drive Specification

The drive is having the following specification:

- Maximum range: 15 mm for the single drives
- Resolution: 0.1 mm
- Accuracy: $\pm 2\%$ of full scale
- Drive Speed: 1.35 mm/s
- Drive load: 40 Kg

$$V = \frac{P \times S}{R} = \frac{1.5 \times 1.2}{1.33} = 1.35 \, mm \, / \, s$$

where V (mm/s) is the drive speed, P (mm/rev) is the lead screw pitch of the drive, S (rps) is the motor speed, and R is the gear ratio.

Software

HMI for the drive control system (Fig. 7) is developed using LabVIEW and installed on an industrial computer at central control room. The software communicates with the remote distributed DAQ modules via control LAN. The same software also collects drive position feedback values and beam current values from the encoder data server and beam current server respectively. In case of any communication error, it displays the error in blink mode.



Figure 7: HMI for the drive control system.

The software takes care of the individual movement of each channel drive in three modes. In the 'press to run mode', the user can freely drive the channel between limits. As long as the run button of the particular channel is pressed, the drive runs until it reaches the limiting point. In the 'preset mode', user can set a position value and the software does the rest to position the channel within ± 0.2 mm of the set value. In the 'auto set mode', user can choose a pre assigned data file containing the position value of all the channels and the software takes care of positioning each channel one by one. It displays a message after setting the position. The software can also provide a bird's eye view of the all the drives with actual positions. It features a protected service mode and an on-screen help for easy explanation of the operational methods, thus providing user friendly environment for reliable operation.

Table 1: Typical accuracy test result performed on Oct 21, 2009*

Drive	Encoder display	Vernier value
E1A	13.4 mm	13.5 mm
E1A	7.1 mm	7.1 mm
E1B	7.1 mm	7.3 mm
M2	9.7 mm	9.7 mm
M2	8.0 mm	7.8 mm
M5	10.6 mm	10.6 mm
M8	10.2 mm	10.14 mm
M8	10.3 mm	10.4 mm

*Some position values could not be measured due to inaccessibility of the drive locations.

CONCLUSIONS

The superconducting cyclotron extraction element drive assembly and its control system are fabricated in-house. The complete control system with associated hardware and software is successfully commissioned in October, 2009 and since then is in continuous operation. At present, this combined instrumentation and control system is running smoothly up to our satisfaction. Efforts are also being made to upgrade some of the field components in accordance with the feedback received from different operators at later stage.



Figure 6: Internal & external view of the control module.

ACKNOWLEDGMENT

The authors are grateful to members of superconducting cyclotron beam development team for their constructive remarks and feedbacks which helped to define and integrate this system meeting user requirements. The authors gratefully acknowledge the sincere effort and active support rendered by the staff members of I&E Section of VECC in helping fabrication and successful commissioning of the system in time.

- S. Bhattacharya, S. K. Mishra, R. N. Sharma, T. K. Bhattacharyya, G. Pal, C. Mallik and R. K. Bhandari, "Fabrication of Extraction System for Superconducting Cyclotron," Proceedings of Indian Particle Accelerator Conference - 2005, p. 515 – 516.
- [2] J. Debnath, S. Paul, M. K. Dey, U. Bhunia, J. Pradhan, P. S. Chakraborty, B. Shoor, Md. Z. A. Naser, C. Mallik, G. Pal, S. Bhattacharya, T. K. Bhattacharya and R. K. Bhandari, "Beam Extraction System and External Beam Line of Kolkata Superconducting Cyclotron," Proceedings of CYCLOTRONS 2010, Lanzhou, China, p. 242-244.

RF DISTRIBUTION AND CONTROL SYSTEM FOR ACCELERATORS OF THE VEC-RIB FACILITY

H.K. Pandey, T.K. Mandi, D.P. Dutta, S. Basak, A. Chakrabarti, VECC, Kolkata, India A. Kumar SAMEER, Kolkata,India K.P. Ray, SAMEER, Mumbai, India

Abstract

RIB facility at VECC has several heavy ion linear accelerators like RFO, two IH-LINACs and one buncher cavity operating at 37.8 MHz and two IH-LINACs with one buncher cavity at 75.6 MHz. Some more RF cavities are being designed at the third harmonic of 37.8 MHz and will be added in the RIB beam line [1]. All the cavities have separate RF power amplifiers with proper amplitude, phase and resonance frequency tuning and control system for efficient and stable operation. The Low Level RF control system has been operational for the power amplifiers of the existing RF cavities and improved design and development is carried out. An embedded controller based data acquisition and processing system is being used for control and local/remote operation. The RF distribution system as well as the design details of RF control system and remote control system will be presented in this paper.

RF DISTRIBUTION SYSTEM

The RF sources for the RIB accelerators consist of three 30 kW RF transmitters operating at 37.8 MHz, one 2 kW RF transmitter operating at 37.8 MHz and a 5 kW RF transmitter operating at 75.6 MHz [2]. A single oscillator at 37.8 MHz drives all the RF power amplifiers of the RIB facility. This is done using a 6-way power splitter

which divides the RF signal to the respective power amplifiers. For LINAC 3, which operates at 75.6 MHz, a frequency doubler is used to convert the 37.8 MHz signal to 75.6 MHz [3]. The forward and reflected power samples and pick-up signal from each of the RF accelerator cavities are fed to the low level RF control modules of the RF amplifiers. The schematic diagram for RF distribution is shown in Figure 1.

LOW LEVEL RF CONTROL

The Low Level RF (LLRF) control system regulates the amplitude and phase of the RF cavity voltage through fast electronic feedback mechanism. The frequency of the cavity is regulated through mechanical movement of a tuner loop in appropriate direction by sensing the frequency deviation. At present the control is based on the conventional amplitude and phase loop method. The scheme of the LLRF control is shown in Figure 2.

Amplitude Control

The important components in amplitude control comprises of the amplitude detector (realized with Schottky diodes HP5082-2835), PI controller and voltage controlled attenuators (realized with pin diodes based π configuration circuitry).



Figure 1: RF Distribution scheme for VEC-RIB accelerators.

The amplitude detector senses the level of the feedback signal (the power sample obtained from pickup port of cavities) and then it is compared with the reference level signal to produce the error signal which is fed to a PI controller. This low-cost ultra stable amplitude control circuit for each RF transmitter has been designed and tested. The measured performance of amplitude stability of the control circuit is $\pm 0.5\%$.



Figure 2: Block diagram of RF feedback control unit.

Phase Control

The important components in phase control comprises of the phase comparator (realized with double balanced mixer configured in phase comparator mode), PI controller and voltage controlled phase shifter (realized with two 0-180° varactor diode based phase shifters in cascade). The phase detector senses the phase difference between the RF input signal and the feedback signal (the power sample obtained from pickup port of cavities) to produce an error signal which is fed to a PI controller which provides an appropriate control voltage. The control voltage is input to a phase shifter which brings the phase of the RF signal to the desired value. For accurate measurement of phase difference, the high frequency RF and feedback signals are mixed with a LO signal to produce an IF signal at 455 KHz. This IF signal is converted to TTL signal and the phase difference is detected using a flip flop based digital phase detector. This stable phase control circuit for each RF transmitter has been designed and tested. The measured performance of the phase satiability is $\pm 0.5^{\circ}$ and phase control range is 360°.

Resonance Frequency Control

In the resonance frequency control, the frequency deviation of the RF cavity is sensed by measuring the phase difference between the forward power sample obtained from a dual directional coupler and the pick-up signal from RF cavity. The phase difference is converted to proportional output voltage using a phase detector (AD8302). The output of the phase detector is fed to a microcontroller (ADuC841) wherein a control logic program generates corrective output signals (direction and/or stepping sequence) which drive a stepper motor holding the tuner loop. The motor rotates the tuner in an appropriate direction until the RF cavity is brought back to resonance. The resonance frequency controller operates only when the VSWR in the input transmission line exceeds a particular value, during which, it switches the level controller to open loop state to prevent two loops from operating at the same time.



Figure 3: Schematic of the planned LLRF control.

It is planned to upgrade the present control system to analog/digital IQ based LLRF control system. A schematic diagram of the low level control unit is shown in Figure 3. To this end, the simulation studies of the analog IQ based LLRF control has been carried out, the details of which are presented in another paper in this conference.

REMOTE CONTROL FOR RF SYSTEM

The remote control system is designed to control and monitor all parameters for the RIB beam line components including the RF transmitters at RIB Facility. This unique system "Distributed Data Acquisition and Control System (DDACS)" has been developed and installed at VECC. The details of DDACS are presented in another paper in this conference. For reliable and fast RF operation, the data acquisition and control system has a 3 layered design as shown in Figure 4, namely: a) Equipment Interface Layer b) Supervisory Layer and c) Operator Interface Layer [4-6].



Figure 4: Schematic diagram of control system.

Equipment Interface Layer

The Equipment Interface Layer consists of Remote Interface Modules (RIM) and Equipment Interface Modules (EIM). The main controller used in RIM is 32bit LPC 2478 ARM7TDMI with analog/digital front-end electronics (ADCs, DACs, Optoisolators, Multiplexers etc.) and RS232 line driver. It also consists of a touchscreen display from which equipments, connected to it, can be locally controlled and monitored. RIM is directly connected to the different analog and digital input/output signals and the local controller for the RF transmitters. The local controller of the RF transmitter is equipped with 8 bit microcontroller, relays, buffers etc. All interlock systems are checked by the local controller. It takes the necessary actions and sends the status of the interlocks to the RIM for display as well as transmission to the Equipment Interface Module (EIM). Some screenshots of the RIM display is shown in Figure 5.



Figure 5: Snapshots from the RIM.

The RIM is interfaced to the EIM through RS232. Each EIM controls a specific RIM along with a specific section of beam line support subsystems. EIM also supports multiple analog/digital input/output channels and RS232/RS485 ports.

Supervisory Layer

The Supervisory layer has been realised using Embedded Controllers (EC1) designed around Single Board Computer (SBC) with Embedded XP operating system. This controller, interfaced with EIMs through fiber optic cables, performs supervisory task of continuously sending command and acquiring data from lower level EIMs and reporting to the operator interface layer as and when requested. The EC1 has higher priority as compared to the EIMs.



Figure 6: GUI interface at operator console.

Operator Interface Layer

Operator interface layer consists of operator console formed with another embedded controller (EC2) and high performance PCs/Workstations for controlling and monitoring machine parameters. The data acquired and analysed at the supervisory layer can be displayed at the operator console and operators can control the whole facility from user friendly graphical interfaces shown in Figure 6. EC2 has higher priority as compared to EC1 and EIMs.

Remote Setting of Control Parameters

The important controlling parameters of the RF transmitter are level controller and phase controller. In the level controller the RF switch and voltage variable attenuator are the main controlling parameters which are used to vary the RF power of a transmitter. The phase controller consists of phase shifter and phase comparator module. The digital signals are used to control the level and phase controllers ON/OFF and a 10-bit DAC output is used to set the level and phase of the transmitter. The corresponding output values are read back using 10-bit ADC in the RIM module. The snapshot of the level control and phase control are shown in Figure 7.



Figure 7: Snapshots of Level/Phase Controllers in RIM.

SUMMARY

The RF distribution scheme and a remote control scheme for RF transmitters have been presented. The remote control system for the RF transmitters has been tested and further tests and refinements are in progress. The present low level RF control system for the RF transmitters are described in this paper. The design of the LLRF system is being improved to IQ based LLRF control.

ACKNOWLEDGMENT

The authors are grateful to Mr K. Mourougayane and his teams from SAMEER, Chennai for their efforts to development of remote control for RF system.

- A. Chakrabarti et al, "The radioactive ion beam project at VECC, Kolkata – Its present status and future plans", Nucl. Instru. & Meth. of Phys. B 261 (2007) 1018.
- [2] H. K. Pandey et al, "RF Systems of the VEC-RIB Facility", Proc. IPAC'10, Kyoto, Japan.
- [3] P. S. Chauhan et al, "Design and Development of 75.6 MHz LINAC for VECC-RIB Facility", InPAC'11, New Delhi, India.
- [4 K. Dutta et al, "Development of Distributed Data Acquisition and Control System for Radioactive Ion Beam Facility at Variable Energy Cyclotron Centre, Kolkata", ICALEPCS'11, Grenoble, France.
- [5] D-DACS Technical Manual, SAMEER, Chennai.
- [6] RF Transmitter Manual, SAMEER, Mumbai.

TESTING OF INDUCTIVE OUTPUT TUBE BASED RF AMPLIFIER FOR 650 MHz SRF CAVITIES

A. Mandal, S. Som, S.K. Manna, S. Ghosh*, S. Seth , S.K. Thakur, S. Saha, U.S. Panda, VECC, Kolkata, India

Abstract

A 650 MHz IOT based RF amplifier has been developed in VECC. It can be used to power several cavity modules in high energy high current proton linear accelerator to be built for ADSS programme in India and in Project-X at Fermilab, USA. The IOT based amplifier requires different powers supplies, water cooling and forced air cooling for its operation. A Programmable Logic Controller (PLC) based interlocks has been incorporated to take care of systematic on/off of the power supplies and driver amplifier, water flow, air flow and other interlocks for the safe operation of the RF System. In addition to that EPICS based RF operating console and data logging/monitoring system has been added.

INTRODUCTION

IOT based high frequency amplifier has been developed in VECC to power several 650 MHz high beta (B=0.61) RF cavities in ADSS programme and Project-X at Fermilab, USA. The amplifier uses the TH 793-1 Inductive Output Tube that delivers an output power of typically 85 kW in CW operation [1]. Specifications are given in Table 1. In addition to this, the amplifier is equipped with various impedance matching and biasing networks and two resonant cavities for input and output tuning. This water cooled amplifier requires five different power supplies naming filament PS, ion-pump PS, Grid PS, Collector PS and focus Coil PS. Among these auxiliary power supplies filament, ion-pump and grid are floating are in high voltage potential, i.e. collector/beam voltage (36 kV) and therefore mounted in an isolated frame in a separated cabinet called high voltage deck. The whole scheme is shown in Figure 1. The mains inputs of High voltage deck are electrically isolated by means of an isolating transformer. The amplifier is cooled by low conductivity water (<2 µS) and forced air. Water flow switches and air flow switches are installed at proper locations to have safety interlocks. In addition to that a PLC works as main control device. It acquires and controls all the analogue and digital readings and settings, supervises and interlocks the slower signals and handles the interfaces to the remote interlocking unit and the local control consol.

CONTROL SYSTEM OVERVIEW

Control architecture of our IOT based amplifier test system is a three layer architecture comprising of device

Table 1: IOT Specifications

Output Power	70	kW
Frequency Range	650	MHz
Bandwidth	6	MHz
Beam Voltage	36	kV
Beam Current	3.4	А
Body Current	50	mA
Grid Current	-40	mA
Filament Current	24	А
Focusing Current	18	А
Efficiency	69.4	% typ
Gain	23.2	dB typ

layer, IOC server layer and user Interface layer. The device layer consists of PLCs which controls the automatic process sequence operations. The PLCs and the process components are configured to satisfy fail-safe operation. The user interface layer consists of the control computers where the operators issue the set points and the mode of operation commands. The system is controlled by means of Siemens S7-300 PLC which is handling all the field I/Os. PLC takes care of all safety interlocks of the amplifier like over-voltage, over-current, water flow and airflow. The PLC is programmed using Simatic manager Step7 [2]. All PLC programmes are written in STL (Statement List) i.e. the program consists of a sequence of mnemonic codes of the commands executed one after another by the PLC. Three auxiliary power supplies for the amplifier are located at high voltage deck floating at 36 kV. These power supplies and other instruments located at high voltage deck are powered through an isolation transformer (isolation: 40 kVolt). An ethernet to optical and optical to ethernet channel is used to navigate the control signals from HV deck to PLC. Two ethernet-optical fibre media converters and a 3 meter fibre optic cable are used to have the isolation between HV deck and PLC located at ground level. A siemens distributed I/O module (ET200) with ethernet takes care of all the I/Os at HV deck. The scheme used for HV isolation is given in Figure 2.

SUPERVISORY CONTROL

Supervisory control is the user interface layer part. In this layer the PLCs are being monitored and controlled

^{*} surajitg@vecc.gov.in



Figure 1: The IOT Amplifier: Schematic overview.



Figure 2: HV Isolation scheme.

through Ethernet based control LAN.We felt the requirement of a common platform as well as exchange data between systems. Also the user interface computers were required to be put at different locations for system integration. After evaluation of requirements, we selected Experimental Physics Industrial Control System (EPICS) [3] as suitable system which fulfil our need. Experimental Physics and Industrial Control System (EPICS) is a set of open source software which provide a software platform for building distributed control systems to operate devices such as Particle Accelerators, Large Experiments and major Telescopes, and is used worldwide for such applications. EPICS is free, hence there is no license fees, no new payment to be made for every upgrade. EPICS architecture shown in Figure 3 has primarily three parts, the operator interface (OPI), the IOC (input-output controller) and the communication network (LAN) which allows the IOCs and OPIs to communicate.

After testing successfully we implemented this in a linux PC in our network to run the IOC to communicate with Siemens PLC with TCP communication. The Graphical User Interface (GUI) is made by using EPICS Motif Editor and Display Manger (MEDM) tool. All the process parameters of all the systems are displayed here for monitoring with a proper navigation. Same GUI can

the respective authors

q

20

0

Filament (V) 12.V Filament (I) 23.8.A Ion Pump (V) 3.2.8.V Grid (I) 30 mA Focus Coil (V) 4.0V Focus Coil (V) 5.000 Collector (V) 2.0.0V Collector (V) 5.000 Bard Daver Walt Li BC.000 Filament Mark Isi (E) BC.000 Bard Daver Walt Li BC.000 Collector Walt Li BC.000 Bard Daver Walt Li BC.000 Collector Walt Li	RF_IOT.adl -	• ×
	Filament (V) 12V Filament (I) 23.8.4 Ion Pump (V) 3.2.8V Ion Pump (I) 10 uA Grid (V) 60V Grid (I) 30 mA Focuss Coil (V) 4.0V Focuss Coil (V) 4.0V Focuss Coil (I) 3.0.8V Collector (V) 23.0.8V Collector (V) 20.000 Filament Gener Walk Lk Exc.000 Collector (V) 20.000 Collector (V) 20.000 Collector	

Figure 4: GUI for Remote operation.

be run in different PCs so that different systems can be monitored together. In our standalone test setup IOC and remote computer where MEDM based GUI is running, is in the same computer. Operators use the GUI shown in Figure 4 to turn on the system step by step.

CONCLUSION

We had operated the system continuously round the clock delivering power to a RF dummy load. Presently total control system is running undisturbed and without any change. The amplifier has delivered up to 14 kW of RF power. Going to higher power levels is restricted due to crowbar problem in main collector power supply. The amplifier system can be easily integrated with other EPICS based system. In future we are looking forward to power SRF cavities with this amplifier.



Figure 3: EPICS control architecture.

- [1] IOT TH 793-1, http://www.thalesgroup.com/ Portfolio/ Documents/Science IOT TH793-1 pdf
- [2] Working with STEP 7, https://moodle.dce.fel.cvut.cz/ file.php/17/Manualy/S7gsv54_e.pdf
- [3] EPICS: http://www.aps.anl.gov/epics

CONTROLS FOR A 10 PETAWATT CLASS LASER FACILITY

D. Pepler, A. Boyle, P. Holligan, A. Kidd and D. Robinson STFC Rutherford Appleton Laboratory, Harwell Oxford, Didcot. Oxon, UK

Abstract

Computerised controls are vital to the operability and flexibility of large-scale physics facilities (such as accelerators, synchrotrons and high-power lasers) in providing fundamental services, for example: automatic configuring of specialist hardware; motion control; firing of shot sequences; enabling precision trigger distribution; vacuum monitoring and control, data acquisition and analysis.

The proposed 10 PW Laser Facility (300 J, 30 fs, 10^{16} W), in line with other major physics facilities around the world, will require a complex computer control system. This is expected to be modelled on the existing Vulcan Laser [1] computer control system and consist of a dozen or so Windows based PCs each of which will be running a separate and dedicated application to control a particular area or function of the facility.

This paper will present an overview of the existing Vulcan laser and provide a status report on the development towards the 10 PW which will require the control system to be designed to allow autonomous operation of the 10 PW Facility as well as to be fully integrated with the existing Vulcan laser controls for combined and synchronised 10 PW plus 1 PW operations.

INTRODUCTION

The Vulcan Laser

Vulcan is an high-power Nd:glass near-infrared laser with a physical footprint 75 m by 50 m (about the size of a couple of Olympic-sized swimming pools). It has roughly 250 m of optical beam path from the various laser oscillator sources through the numerous laser amplification stages and on to the target areas. The laser is a multi-beam pulsed system with 'long' pulses of 0.3 ns to 6 ns duration synchronised with 'short' pulses of 500 fs to 20 ps duration. The initial short pulses start at the nJ level with 1 mm ~ 2 mm diameter beams and are amplified up to 500 J at 600 mm beam diameter providing a peak power of 10^{15} W i.e. a Petawatt. Using a large (1 m diameter) off-axis parabola, this beam is then focussed down to 5 microns onto various targets, where an intensity of up to 10^{21} Wcm⁻² is obtained. The facility is primarily used by university researchers working in the field of plasma physics to conduct fundamental research in particle acceleration, astrophysics and fusion energy.

The Vulcan Control System

The present control system as indicated above allows for a significant number of operational features to be manipulated. This is done either locally to the equipment or, more normally, remotely from the main control area as personnel are not permitted to be inside the laser area when high-power shots are being fired. In the main control room the laser operator is able to interrogate or change the status of equipment, change beam line parameters and initiate shot sequences. For this, the operators are presented with a number of control screens (generally touch activated) which gives them access to a real-time overview and control of the current state of the laser hardware. Figure 1 shows two of the central control interfaces used by the operators. These and other key control applications have been written in-house using Delphi 7 (visual Pascal) a choice that has allowed the development of a very robust and flexible operational system.



Figure 1: Laser Operators central control screens for (left) the main control, data acquisition and diagnostic interface; and (right) the system schematic showing overall active status of individual components.

Behind the screens is a private, isolated Ethernet network with a distribution of thirteen PCs and nine CAMAC crates. These are variously populated with analogue and digital electronics to drive, for example, pneumatic slides and oscillator timing controls or, to capture flashlamp discharge waveforms and provide interlock monitoring.

10 PW DEVELOPMENT

Overview

Physically the 10 PW will be another large-scale laser system and it is proposed to be built as a major extension alongside the existing Vulcan system. The 10 PW will be driven with a new front-end oscillator to generate the much shorter optical pulse length of 30 fs but for the main amplification to 300 J it will use flashlamp-pumped Nd:glass amplifier technology similar to that already used in Vulcan.

The 10 PW will be designed to operate either standalone into a dedicated target area or to be synchronised with Vulcan to provide a 10 PW + 1 PW capability into an existing target area.

10 PW Controls

It is acknowledged that CAMAC although proven and very robust, is an aged technology and options for the new local chassis of electronic real-world interfaces (such as analogue to digital converters, digital delay generators, and digital input / output registers) are being investigated. Options include Advanced Mezzanine Card technologies, Telecommunications Computing Architecture systems (such as Advanced TCA or µTCA), PCI extensions for Instrumentation (PXI), compact PCI (cPCI) crates and Industry Pack modules.

Because of the close proximity, similarities and linkage between the two systems the 10 PW could be seen 'simply' as an extension of Vulcan, so far as the controls are concerned. As the Vulcan control applications are written in-house and are somewhat modular in form the 10 PW computer controls are expected in principal to be copied from Vulcan and modified to suit the new hardware. Keeping the same modules allows integration between the two laser systems to be readily achieved. Keeping the same forms of display also allows for an easier operational integration with Vulcan as display screen familiarity will have an impact on minimising training needs.

Control Synchronisation

In the tandem operational mode where Vulcan and the 10 PW laser systems are required to fire together, it will be important to have the shot sequencing of both systems operating in parallel at the same time. It also is expected that the 10 PW in this mode will be operated remotely from the Vulcan control area. This will need the Vulcan control system operating as a master controller and the

10 PW running as a slave system. Communication between the two systems will generally use standard ethernet TCP/IP or UDP protocols which will enable Vulcan to have the 10 PW interlock status checked alongside its own.

With both systems declared safe and ready, the Vulcan control system will instruct 10 PW to initiate a charge sequence. Depending on which amplifiers are selected, the 3 MJ, 20 kV capacitor banks take about 30 s \sim 40 s to charge so will need to be independently started at a time appropriate for all Vulcan and 10 PW amplifiers to reach their respective end-of-charges (EOCs) at the same time. When both systems indicate EOC the Vulcan control system will initiate a cascade of electrical triggers for both Vulcan and 10 PW shot usage.

Trigger Requirements

A range of diagnostic and data acquisition systems will require specifically timed triggers. Triggers one second early can easily be provided programmatically as part of the shot sequence but later or more precise triggers will require separate hardware. Triggers 10's of millisecond early can be readily created through the use of digital delay generators whether these are stand-alone units or part of the chassis electronics; nanosecond triggers will need more sophisticated hardware (such as the use of precision delay generators from Bergmann [2], or Micro-Research [3]) which allows for a master trigger cascade to be produced synchronous to an RF cycle derived from the optical pulse train output of the 80MHz source laser oscillator.

Optical Synchronisation

For the combined operation of Vulcan with 10 PW there will need to be precise temporal synchronisation at the target of both the 500 fs pulse from Vulcan and the 30 fs pulse from 10 PW. This will be a challenge and will require extremely finely tuned optical phase-locking together of the oscillator sources from the Vulcan and 10 PW front-ends.

Conclusion

The 10 PW as proposed is fully designed and scientifically reviewed but the new building required awaits some £6 M of capital funding. Nonetheless, development and testing of the 10 PW new front-end and pre-amplification stages are being actively pursued with the expectation that funding will materialise in due course.

- [1] www.clf.stfc.ac.uk/Facilities/Vulcan/12248.aspx
- Bergmann Messgeräte Entwicklung KG, www.bme-[2] bergmann.de
- [3] Micro-Research Finland Oy, www.mrf.fi

INTEGRATED CONTROL SYSTEM FOR LEHIPA

Sandeep Bharade, Paresh Motiwala, Gopal Joshi, T.S. Ananthkrishnan, Chandra Kant Pithawa, Electronics Division, BARC, Mumbai, India Arindam Basu, Sudheer Singh, Pitamber Singh, IADD, BARC, Mumbai, India

Abstract

The Low Energy High Intensity Proton Accelerator (LEHIPA) is a 20 MeV 30 mA proton accelerator which will be achieved in multiple stages [1]. LEHIPA consists of several sub systems/devices located at different positions of the beam path which includes ION source, RF Power, RF Protection Interlock System, Low Conductivity Water plant, Low Level RF control Systems, Vacuum System, Beam Diagnostics & Beam Line Devices. All these subsystems have their own local control systems (LCS) which will coordinate the operation of the corresponding subsystem. The control system for LEHIPA is thus being designed as a Distributed Control System with different teams developing each LCS. The control system will assist the operator to achieve a beam of desired characteristics by interacting with various sub systems of the accelerator in a seamless manner, protect the various parts of machine by generating the necessary interlocks, keep track of various parameters monitored periodically by suitably archiving them, alarms annunciation and trouble shoot from the control room. This paper describes approach to system design of ICS.

SYSTEM DESCRIPTION

The Low Energy High Intensity Proton Accelerator (LEHIPA) is a 20 MeV 30 mA proton accelerator. An overall design of the LEHIPA is given in Fig. 1. Electron Cyclotron Resonance (ECR) ion source gives DC proton beam of energy 50 keV and current of 30 mA. The lowenergy beam transport (LEBT) system will consist of a magnetic system comprising of two solenoids that will transport the beam from the ion source, and match the beam into the acceptance of the RFQ. It will also have various beam diagnostic systems. For the acceleration of the proton beam at low energy, a four-vane RFQ (3 MeV, 30 mA, 350 MHz) is used. A Drift-Tube Linac (DTL) structure will accelerate the beam up to 20 MeV. The DTL is preceded by medium-energy beam transport (MEBT) section. The MEBT section will consist of four magnetic quadrupoles and a rebuncher cavity that will transport the beam from the RFQ to the DTL structures that follow, and match the beam into the acceptance of the DTL. For both the RFQ and the DTL high-power CW klystrons are used to generate the CW power of around 2 MW CW, at 350 MHz, along with high-power microwave lines to transport the RF to the accelerating structures.



Figure 1: LEHIPA subsystems.

INTEGRATED CONTROL SYSTEM (ICS)

The integrated control system is designed as a 3 tier architecture depicted in Fig 2. The OWS layer hosts the presentation applications and run manager which is used by operators for machine run. The middle layer is the server layer where command control, parameter and configuration servers are hosted. The command control server is responsible for permitting commands to the equipment server depending upon macro level machine state logic. Parameter server is a data concentrator which collects data from different equipment servers, arrange data as per properties and present it to OWS layer. The equipment layer is the lowest layer. It is presented to the server and OWS layer through standard interface, it receives commands from higher layers and multicast its parameters to OWS layer. It hosts the access to the hardware. At the equipment level, the various actuators, sensors and measurement devices are interfaced to the control system through three different types of front-end computers.

CPCI computers dealing with high performance acquisitions and real-time processing; these employ a large variety of I/O modules. Typically, the LEHIPA beam instrumentation, RF Electronics System and the LEHIPA beam interlock systems use CPCI front-ends.

Systems like LCW System, ECR Injector uses PC based gateways interfacing systems as a large quantity of identical equipment is controlled through field buses.

Programmable Logic Controllers (PLCs) are used to drive various sorts of industrial actuators and sensors for systems such as the Klystron System or the Vacuum system.



Figure 2: ICS Architecture.

INTERLOCK FOR ICS

Interlocks are needed for the protection of the machine and personnel where, during an unexpected event a safety action has to be implemented. The action taken will depend on the subsystem. The worst case scenario will include shutting of the beam and all the relevant power supplies. The interlocks will be of two types for each subsystem: an interlock generated by the subsystem and an interlock which is supplied to the subsystem. Each system will define the significance of the interlock generated by that system. These interlocks are made available to other subsystems through the ICS, which will monitor and process the interlocks and activate the various interlocks which are connected to individual subsystems, as required. The processed interlocks are then distributed to each sub systems. Subsystems will decide what action has to be taken on the activation of interlock provided to it from the ICS.

A search and secure operation philosophy will also be provided for personal safety and the status of search and secure system will be given to ICS. Depending upon various searches and secure status appropriate operation permission will be decided by ICS. The interlock operation will depend upon mode of the accelerator viz with beam, without beam. Also interlock system has to be programmable as during commissioning stages a reduced set of interlock operations will be required.

An emergency shutdown command generated from emergency console, in absence of loss of connectivity with the OWS, will be part of the global interlock in which case all the systems are expected to shutdown as an emergency shutdown. The 'beam hall open' condition will provide input to the ICS as well as a hardwired interlock directly to the Ion Source Subsystem for shutting down the beam.

TIMING AND PULSING FOR ICS

One of the major requirements for the Accelerator operation is the pulsed beam operation. LEHIPA, being a high current accelerator, will not be operated in Continuous Working (CW) mode in the initial stages of operation and also during start-up of the accelerator. For this purpose, beam will be operated in pulsed mode whenever required with facility to switch over from one mode to another mode which can be set from the OWS. In the initial phase of commissioning and also during each start-up after a shut-down, LEHIPA will work in a pulsed mode. During the pulsed mode a sequence of actions takes place. During the initial phase of the 'on' period, RF is set-up and stabilized in different resonators of the linac. A timing signal is provided to the LLRF and high power RF system, which perform the action of setting-up and stabilizing of the RF field in different resonators. After stabilizing the fields, the beam is injected into the resonators. After a predetermined time, signalled by a trigger, the beam is removed and after this RF is removed from the Resonator. The beam diagnostic system also receives suitable timing signals to provide beam parameters during the beam on period. This operation is repeated for each cycle of pulsing. Before going to the continuous mode of operation the average beam current is slowly increased by changing the duty cycle. All this action sequence is managed by a pulsing system which emits timing pulses to different sub-systems (ion-source, LLRF and RF power systems of different resonators and beam diagnostic system) of the linac. This pulsing system is essentially a number of programmable pulse generators. where the parameters of pulse sequences are under operator control.

In addition to the pulsing system a RF master signal is distributed to different subsystems. In order to work as an accelerator, the RF field in different resonators of the LEHIPA has to be phase locked with this RF reference signal. This reference phase line is provided to the LLRF system, which stabilizes the field in the resonators with respect to it. The phase of the beam at any point in the LEHIPA is obtained by comparing the phase of the beam induced RF in the pick-up type of monitors along the LEHIPA, with respect to the RF phase reference. Phase information from two pick-up helps to determine the energy of the beam. In addition to serving the phase reference for the LLRF and beam pick-up devices, this reference phase also constitutes the basic clock from which other clocks for example, for down/up conversion, sampling are derived. The ICS will provide this RF reference phase signal for the LEHIPA.

ALARMS ANNUNCIATION

Each subsystem will be monitoring a large number of parameters, the important parameters and operation condition will be announced as alarms to draw the operator attention. Whenever a parameter which is monitored is not in limits (low, high, very low and very high) then alarm has to be announced by the ICS. All the necessary interlock actions local to the subsystem will be taken care of by the subsystem. Thus all the parameters which are monitored for alarm annunciation should be made available to the ICS. ICS will annunciate the alarm condition generation and removal on dedicated display. Also at a later stage the alarm annunciation process has to provide intelligent conclusions which may lead to revision of the requirements. Also the time sequence of alarm generation/removal has to be archived for future analysis.

DATA ARCHIVING FACILITY

Data servers have to be provided as many of the sub systems will keep their periodically monitored parameters for diagnosis and design improvement purpose in the archives. Also parameters which are used for alarm generation will also be available in the archives. Sequence of alarm generation and removal, diagnostic messages and operator logs will be parts of the archives.

OWS PROTOTYPE

The operator workstation should provide a facility that operator can easily operate the accelerator and focus on important jobs for operation the OWS prototype has been developed. The LEHIPA simulator which displays parameters of accelerator of different subsystem and provides operational parameter change facility has been developed. This simulator along with OWS prototype was tested with Server software which connects to different subsystem and fetches parameters. This prototype has provided as design inputs for software design phase.



Figure 3: OWS Prototype.

REFERENCES

 P. Singh et al, "Accelerator development in India for ADS programme," Indian Academy of Sciences, Vol. 68, No. 2, Journal of Physics. p 331 (2007); http://repository.ias.ac.in/15620/1/333.pdf

CONTROL SCHEME FOR REMOTE OPERATION OF MAGNET POWER SUPPLIES FOR INFRARED FREE ELECTRON LASER

L. Jain[#], M.A. Ansari, V.P. Bhanage, C.P. Navathe, RRCAT, Indore, India

Abstract

Infrared Free Electron Laser (IRFEL) is under development at MAASD, RRCAT Indore. The IRFEL machine consists of 90keV thermionic gun as electron source, beam transport line, 25MeV Linear Accelerator (LINAC) and an undulator magnet. There are fifty magnets on beam transport line. These magnets are energized by precision power supplies. These power supplies have local as well as remote control and will be located at equipment hall. The control room and equipment hall are at approximate distance of 300 m. We have planned a three layer structure for centralized operation of Beam Transport line Magnet Power Supplies (BTMPS). These layers are device interface layer, the equipment control layer and the presentation layer. Presentation layer is linked with equipment control layer on Ethernet. Whereas equipment control layer will be linked to device interface layer by RS-485. Device interface layer consist Magnet Power Supply Controllers (MPSC). Each MPSC has one master and five slave controllers linked on isolated SPI bus, which will control five BTMPS. We have developed slave controllers and a master as prototype of MPSC. This paper describes MPSC prototype and proposed control scheme.

INTRODUCTION

The Free Electron Laser (FEL) laboratory is developing Infrared Free Electron Laser (IRFEL) as part of their ongoing development activity. The machine is having a 90Kev thermionic gun as an electron source, beam transport line, 25Mev linear accelerator (linac) and an undulator magnet. The beam transport line for the FEL has two parts: a low energy part & a high energy part. Low energy part has solenoids to transport a beam from the thermionic gun to the linac entrance. A high energy beam transport line consisting of dipoles, quadrupoles and steering magnets for transporting beam from the linac exit to the entrance of the undulator. These magnets are energized by Beam Transport line Magnet Power Supplies (BTMPS). There are fifty numbers of such BTMPS. These BTMPS have local as well as remote control. BTMPS are located in equipment hall. BTMPS will be controlled from control room located at a distance of approximately 300 meters from equipment hall. These BTMPS are remotely operated by Magnet Power Supply Controller (MPSC) connected to control computer on RS-485 network. Inside MPSC, there is one master and five slave controllers linked on isolated SPI bus. Each MPSC is designed to control five BTMPS.

BEAM TRANSPORT LINE MAGNET POWER SUPPLY (BTMPS)

Several types of magnets are required to transport the beam through beam line. For these magnets, numbers of supplies of various voltage and current ratings are required. We have listed power supply types and there important parameters in Table 1.

Table 1: BTMPS Parameters.

Magnetic element	Qty	Current rating	Voltage rating	Required Stability
Corrector Magnets	15	7 A	+/- 15V	100PPM
Dipole Bending Magnets	5	20 A	30V	100PPM
Quadurpole Magnets	15	13 A	+/- 15V	100PPM
Steering Magnets	15	10 A	+/- 10V	100PPM

Presently five power supply units are mounted in a 6U card frame. Hence there are ten such card frames, housing 50 supplies. Each power supply can be operated in local mode from front panel and in remote mode from PC. There is a 25-pin sub-D connector provided on the power supply which has digital & analog I/O signals for power supply remote operation. Each power supply has one analog input signal in the range of 0-10V to set the current output, 0-10V analog output signal which is proportional to set current. There are four digital inputs to power supply. These inputs are used to switch on power supply, to switch off power supply, one input to reset power supply faults and one for changing polarity of power supply current output. Power supply provides three digital status lines with galvanic isolation. They are ON/OFF indicator, Fault indicator, and Remote/Local indicator.

MAGNET POWER SUPPLY CONTROLLER (MPSC)

Keeping power supply configuration in mind we have designed MPSC in standard 3U card frame. Each MPSC card frame has five independent power supply controllers (slave controllers) and one master controller as shown in Figure 1. Master controller accepts command from PC located at control room via RS-485 port and then sends these commands to appropriate power supply controller over a local isolated SPI bus. We have planned to place these MPSC in 19" rack of 32U height. This implies 15

[#]lalita@rrcat.gov.in

power supplies and their controllers will be located in one 19" rack of 32U height.



Figure 1: Block Diagram of proposed Magnet Power Supply Controller (MPSC).

SLAVE CONTROLLER

A slave controller as shown in Figure 2 provides required control and interface signal to operate magnet power supplies remotely.



Figure 2: Block Diagram of Slave Controller.

It receives command from master controller and takes necessary actions. This card has a micro-controller [1] with 64K E²PROM, 2K SRAM, 18-bit DAC [2], 15 bit ADC [3], 4 digital relay outputs and four digital inputs. It provides reference signal to set current output of power supply. Hence power supply ground and reference ground is common. Similarly current read back signal is sharing same ground as power supply ground. Since we are having five slave controllers in single chassis sharing a SPI communication bus, to avoid ground loops we have used an isolated SPI bus for communication. Each slave controller has local DC to DC converter which isolates ground from backplane power bus. This slave controller also has a RS-232 interface which was used while developing the software to download code.

Since current stability needed for power supply is of the order of 100 PPM we have used DAC with 18-bit resolution with analogue output stability of 2 PPM/°C. The output of DAC is buffered with high accuracy low drift operational amplifier to maintain overall reference stability better than 50 PPM.

RESULTS

For testing the stability of power supply reference, DAC output was set to 2V and observed for whole day, it is within 50 PPM as shown in Figure 3.



Figure 3: Voltage Drift vs. Time for enclosed systems.

CONCLUSION

The prototype of the MPSC has been designed and built. It has been tested for its reference stability performance for more than a week with 7-1/2 digit Keithley 2001 multi meter.

ACKNOWLEDGEMENT

The authors wish to thank Shri K.K. Pant and his team, MAASD, RRCAT for their active support.

- [1] 80C51 Family architecture, www.nxp.com
- [2] http://www.analog.com/static/imported-files/ data_sheets/AD1139.pdf
- [3] http://datasheets.maximic.com/en/ds/MAX135.pdf

A DISTRIBUTED CAN BUS BASED EMBEDDED CONTROL SYSTEM FOR 750 keV DC ACCELERATOR

A. Kasliwal, T.G. Pandit, RRCAT, Indore, India

Abstract

This paper describes a distributed embedded system that uses a high performance mixed signal controller C8051F040 for its DAQ nodes and is based on CAN bus protocol for remote monitoring and controlling of various subsystems of 750 keV DC accelerator based irradiation facility at RRCAT, Indore. A PC with integrated PCI CAN card communicates with intelligent DAQ nodes over CAN bus and each node is interfaced with a subsystem. An opto isolated SN65HVD230 CAN driver is interfaced between each node and physical bus. Remote frames and message prioritising are used for efficient control. The PC application is developed using LabVIEW 8.6. The proposed system is more reliable and noise immune as compared to previously used systems that initially used a centralized system based on C8051 controller. This was then upgraded to a distributed system that used microcontroller AduC812 and communicated over RS485 link. The new system has been integrated and tested satisfactorily for its designed performance with test jigs that simulated the actual subsystems with a bus length of 75 meters. First the complete scheme of the system is presented, and then the hardware and software designs are discussed.

INTRODUCTION

A 750 keV 5 mA electron beam accelerator has been developed and commissioned for industrial processing application at RRCAT, Indore. In this accelerator a directly heated diode gun is used as an electron emitter, which is floating at minus 750 keV DC with respect to the earth. The accelerating voltage for the accelerator is generated using a 12-stage symmetrical voltage multiplier stack. The input to the multiplier stack is derived from a high frequency resonant inverter through ferrite core high voltage transformer. The inverter output and hence the high voltage is varied by controlling the DC bus voltage of a three phase fully controlled converter which feeds the inverter. The filament power supply is derived from the top of the multiplier stack across its isolation columns, which has an inherent isolation of 750 kV with respect to the earth. The accelerated electron beam is focused by two focusing coils and is made to scan the material under process with the help of a scanning magnet field. The high voltage stack, the accelerating column and the filament power supply are closed in a pressure vessel, which is filled with SF6 gas at a pressure of 5 atmospheres. The accelerator, which is under operation at RRCAT, Indore since 2002 is being used for research and development in the field of irradiation processing.

The PC based control and monitoring system of this DC Accelerator is being upgraded for easier operation and

better control and monitoring of different parameters of the accelerator and allied subsystems.

HARDWARE DESCRIPTION

The PC based system, controls and monitors different parameters of various subsystems of the DC accelerator which includes different power supplies, various control units, transducers, on/off controls, search and scram system etc. The analog parameters, which are being controlled and monitored, include accelerating voltage, emission current, filament current, scanning coil current among others.

In the proposed scheme that uses a distributed control system communicating over CAN(Controller Area Network) bus, each power supply and subsystem which is to be monitored and controlled through PC is interfaced with a data acquisition card (DAQ node). The DAQ node implements a microconverter C8051F040 from Silicon laboratories, which has an inbuilt high speed 8051-compatible CIP-51 core and a CAN controller. It also has a 12 bit ADC with 8-channel analog multiplexer and two 12-bit DACs. All analog inputs are taken in through differential amplifiers to reduce the effects of common mode noise. The DAQ node can also monitor eight digital inputs and control eight digital outputs.

All the DAQ nodes are connected on a common CAN bus line. The control PC is connected on the same CAN bus through an integrated PCI CAN card. CAN bus message based communication protocol is used for communication between PC and the DAQ nodes.

Each DAQ node will be allocated messages with specific addresses and will act as a slave. The PC is the master and communicates with the power supplies and other subsystems through this known message addresses using normal and remote frames. The data will be embedded into the well-framed message packets. As a result of the content-oriented addressing scheme, a high degree of system and configuration flexibility is achieved.

CAN is a distributed serial bus system widely used in industrial control applications, where robust environment exists. It provides high level of data integrity and automatic bus arbitration. It supports a multi-master hierarchy, which allows building intelligent and redundant systems with broadcast communication. It provides sophisticated error detecting mechanisms and retransmission of faulty messages. For error detection the CAN protocol implements three mechanisms at the message level: Cyclic Redundancy Check (CRC), Frame check, ACK errors. The CAN protocol also implements two mechanisms for error detection at the bit level: Monitoring: Each station that transmits also observes the bus level and thus detects difference between the bit sent and the bit received. Bit stuffing: The bit representation used by CAN is "Non Return to Zero (NRZ)" coding. The synchronization edges are generated by means of bit stuffing. This stuff bit has a complementary value, which is removed by the receivers.

It is easy to add stations to an existing CAN network without making any hardware or software modifications to the present stations as long as the new stations are purely receivers. This allows for a modular concept and also permits the reception of multiple data and the synchronization of distributed processes. Data transmission is not based on the availability of specific types of stations allowing simple servicing and upgrading of the network.

CAN bus supports lowest two layers of OSI reference model namely physical layer and data link layer. Physical layer is implemented by using opto isolated SN65HVD230 CAN transceiver from Texas Instruments and data link layer is implemented by inbuilt CAN controller as shown in Figure 1.



Figure 1 : The Layered ISO 11898 Standard Architecture.

The CAN transceiver employs the CAN serial communication physical layer in accordance with the ISO 11898 standard and provides differential transmit capability to the bus and differential receive capability to a CAN controller at speeds up to 1 Mbps as illustrated in Figure 2.



Figure 2: Details of typical CAN DAQ node.

ISO 11898 is the international standard for high-speed serial communication using the CAN bus protocol. It

supports multimaster operation, real-time control, programmable data rates and powerful redundant error checking procedures that provide reliable data transmission. It is suited for networking intelligent devices as well as sensors and actuators within the rugged electrical environment of a machine chassis or factory floor.

A DB-9 connector is provided on the card to facilitate serial connections to the CAN interface on the C8051F040. The DAQ card has additional communication interface for opto isolated RS485 based communication. The communication protocol can be selected as either CAN bus or RS485 by switching the jumper settings. The opto isolation reduces the problem of noise interference and ground lifting due to ground loops. The DAQ card also has a JTAG connector that provides access to the JTAG pins of the C8051F040. It is used to connect the USB Debug Adapter to the DAQ board for in-circuit non-intrusive debugging and Flash programming.

The control software is written into 64k bytes of insystem programmable FLASH memory. Figure 3 shows the schematic of the proposed control system.



Figure 3: Schematic of the control system.

SOFTWARE DESCRIPTION

Two distinct application softwares are being developed, the microcontroller firmware application and the host PC application. The microcontroller firmware is written in 'C' using KEIL 'C' cross compiler. The program is structured into various modules for carrying out different tasks like CAN intialization, setting up of ADC/DAC, handling digital I/O, interrupts, transmission, reception and decoding of CAN frames.

The PC application and its GUI interface is being developed using LabVIEW 8.6 from National Instruments, for multiple reasons like easy development, debugging and maintenance, excellent windows based graphical user interface and strong support for CAN bus based serial communication.

The PC is used to control the on-off operations or set references of different power supplies and other subsystems and display different parametric values and the status of the machine. The PC starts the system in the search mode and doesn't enter into operation mode till all the pre specified safety and operational conditions are met.

EARLIER CONTROL SCHEMES

Initial scheme [1] employed a single centralized microcontroller crate and used a number of Input, Output, ADC and DAC cards for communication with various subsystems. The micro-controller system then communicated with PC through RS232 serial link. The software including its GUI for the PC operation was developed using C++ programming language, which imposed several programming limitations.

It was then upgraded to a distributed control system [2] based on RS485 where each subsystem to be controlled through PC was equipped with a AduC812 based microcontroller card and was interfaced to PC through a common RS485 line. Each interface card had a unique predefined address. A RS485 to RS232 and RS232 to RS485 converter was used at PC end. The Control Software for PC was written in LabVIEW 7.1. PC was master and controllers operated as slaves

This system suffered from several shortcomings. Data frame creation and decoding was implemented at software level instead of hardware level both in microcontroller and LabVIEW code due to missing data link layer. To achieve reliable data transmission additional coding was required to implement data checking and retransmission capability. Analog inputs on interface card required additional filtering to minimize common mode noises. The need for better debugging capabilities for microconverter coding was felt.

With low data handling requirements and above mentioned required features, a CAN bus based system suited the application. Hence microconverter "C8051F040" was chosen as it has 8051 core with inbuilt CAN controller, 12 bit ADCs & DACs, sufficient IO ports and JTAG interface.

TESTS AND RESULTS

CAN bus based data acquisition cards have been designed and developed. The embedded system was tested thoroughly and performance at component and module level was observed. Each module was tested individually for its performance with simulated inputs. A test bench was developed to test the system. Initial test codes for controller card and software application for PC operations have been developed. The system is being tested and it is possible to both monitor and control the analog and digital controls of the test bench through the LabVIEW based GUI on the PC. Data is being transferred between the PC and the CAN card using CAN bus protocol at 1 Mbps. Data transfer over a cable length of 75 meters has been tested successfully at 500 kbps. The new design has given us the capability of adding more subsystems (like any additional power supply etc.) as and when required. It requires addition of a new DAO card with allocation of new message addresses and hooking it on the existing CAN bus line.

ACKNOWLEDGMENT

Authors acknowledge the contributions of Shri Rajesh Nagdeve and late Shri Amit Upadhyay towards development of the system and are thankful to Shri A.C. Thakurta and Shri R. Banwari for their valuable guidance.

- Kasliwal A. et al, "PC based control system for 750 keV DC accelerator", InPAC-2003, CAT, Indore.
- [2] Kasliwal A. et al, "Upgradation of PC based control system for 750 keV DC accelerator," InPAC-2005, VECC, Kolkata.

HIGH VOLTAGE CONTROLLER SYSTEM FOR SPECTROSCOPY **DIAGNOSTICS OF SST-1**

Hitesh Mandaliya[#], Minsha Shah, E.V. Praveenlal, Rachana Rajpal, R. Jha Institute for Plasma Research, Bhat, Gandhinagar, India.

Abstract

We have developed special instrumentation for spectroscopy diagnostics of the SST-1 Tokamak. Light output in the visible spectrum is guided through fiber optics from the Tokamak ports to the diagnostics Hall, where photo multipliers tubes and other instrumentation electronics are kept. High Voltage (0 - 1500 V) bias generation electronics is required to bias these PMTs. Total 14 PMTs to be biased for overall requirements of the diagnostics. We have developed modular electronics for HV bias generation, which consist of one controller and seven HV modules. We have designed and developed FPGA based controller card which controls seven HV modules. The Slot-0 card is having Spartan 3E FPGA and Standalone Controller Area Networking (CAN) controller. 32-bit RISC processor Microblaze has been deployed into the FPGA. We have used Hitek make HV supply modules which is programmable. In the HV modules, Analog Device Inc. make iCoupler, digital isolators are used to break the ground loops and to avoid ground-lifting problem. Various features like Manual mode/Remote mode operation, HV ON/OFF, HV Value setting through remote GUI have been developed on LabVIEW software.

SPECTROSCOPY DIAGNOSTICS ON SST

Spectroscopic diagnostic is an important tool to know about impurity content in plasma and other plasma parameter like electron and ion temperature, electron density, Z-effective. Light from plasma is been transported from a glass window through optical fiber and lens arrangement up to the detector, (photo-multiplier tube). In order to detect a range of signal we need to keep options of variable gain and filter bandwidth, programmable biasing of PMT to detect weak signals. This helps to study some of the perturbation at faster time scale.



Figure: 1 Spectroscopy Diagnostics scheme of SST-1.

mhitesh@ipr.res.in

So we need to design electronics system with which we can play to study and monitor this phenomenon in plasma and characterize it during experimental campaigns. Total seven optical ports will be used for spectroscopy diagnostics of Phase-1 of SST-1 development. Figure 1 shows the diagnostics general block diagram.

HV ELECTRONICS DESIGN

The SST-1 spectroscopy diagnostics requires two types of electronics, one is analog signal conditioning, which is responsible for amplifying PMT current output and converting into voltage. The second one is high voltage bias generation. To meet the physic experimental requirements it was decided to keep the high voltage remotely programmable. To achieve the above mentioned goal modular electronics development was chosen. The 4U/10T size modular chassis is housing of the electronics pcbs. This chassis has total eight 4U slots, The Slot-0 is Microprocessor based card. Remaining seven slots (Slot-1 to Slot-7) are high voltage generation modules. The Slot-0 controller is intelligent device and is responsible for overall control of all seven HV modules and also communicates with remote computer with Controller Area Networking (CAN) protocol. We have chosen 32 bit RISC architecture based MicroBlaze soft processor on FPGA as main control unit. Because it is soft processor we could add many soft peripherals to it according to requirements of controlling seven HV modules.



Figure: 2 Modular structure of HV Electronics.

At the time of designing electronics we also considered grounding issues, PMTs are connected to not only HV modules but also seven different transconductance amplifier. These seven amplifiers use common reference ground and are housed in separate 10T/6U chassis. If we develop seven HV modules having common reference then it could lead to ground loop formation. Also because of safety we decided to provide isolation to HV modules from tokomak signal ground. To meet above two requirements we decided to generate floating HV i.e. each module should have its own reference point and there is no common reference for all seven modules. To achieve this we isolated all the digital control lines to and from microprocessor with Analog Devices Inc make iCoupler devices, iCoupler is digital isolator with chip scale transformer inside. This provides 5kV isolation to signal ground and also helped in breaking the ground loops. The secondary side of iCoplier is HV module interface, this HV module interface is powered with isolated AC/DC converter from MORNSUN power, China. Separate AC DC converter for each module is used for HV bias to generate floating supply for each module.

Slot-0 Controller Hardware

As mentioned above this is main control unit for the entire system. We have used MicroBlaze soft processor. The MicroBlaze Micro Controller System (MCS) is highly integrated standalone processor system intended for controller applications [1]. Data and program is stored in a local memory, debug is facilitated by the MicroBlaze Debug Module, MDM. peripheral IP was not available with us. We decided to use external hard block. The processor receives various commands and HV value, HV ON/OFF information from the GUI computer over CAN network. Upon receiving the CAN frames it decodes them and then performs actions of turning ON/OFF HV module or HV amplitude setting. To turn ON/OFF one module one bit Output port of GPIO IP is used. Total such 7 lines are used for turning ON/OFF HV modules. HV amplitude is set by 0 to 10V control voltage generated by Digital to Analog Converter inside the HV module. DAC update IP is developed for accessing the DAC inside the HV Modules via backplane from Slot-0.

High Voltage Module Hardware

The Hitek make HV supply modules are used inside the HV Module which needs 0-10V control voltage to set the HV output from 0 to 1500V. Each module receives isolated 24V supply from backplane via 96pin euro connector. Each module also received digital lines from Slot-0, which are HV ON/OFF, DAC Data lines (12bit), DAC Write strobe signal, Module identification 4 lines. The backplane is designed in such a way that each HV modules can be inserted in any slots from. Slot-1 through Slot-7 of the 4U chassis. DAC write strobe and HV ON/OFF signals are uniquely routed for each module in the backplane. The hardware also supports manual operation of the individual HV Module. DPDT switch is provided on facial plate of each HV Module to operate module either in Remote Mode or Manual Mode. From Baclplane via 96



Figure: 3 Slot-0 Controller Blocks.

Xilinx Platform Studio (XPS) is used to configure and set up the hardware platform of the soft processor. We used external standalone CAN controller SJA1000 as CAN



Figure:4 HV Module Block diagram.

In Manual Mode the Hitek power supply receives control voltage from voltage divider and its voltage is controlled by binary switch available on facial plate. In Remote mode the Hitek module receives control voltage from 12 bit Digital to Analog Converter. 12 bits provides enough resolution to vary the final HV output in step of 1V.

This DAC is controlled by the digital control signals from the Slot-0 Controller. We decided to route only digital signals over backplane of the chassis. Analog Signals suffer from noise because of long traces and two euro connectors in their signal path. This helped in achieving better accuracy and precision in HV setting.

SOFWARE PART

Embedded Program for MicroBlaze

Xilinx Plateform Studio (XPS) is used to develop the C code for MicroBlaze soft processor. XPS provides various APIs to interact with various IPs. It prepares various CAN frames and also decodes received CAN frames and executes subroutines of setting DAC value, HV ON/OFF bit setting. Total four such chassis will be used. We use CAN ID field of CAN frame to address the particular Chassis. We have introduced another fields like Code ID, Module ID, Code Data which are part of the CAN Data fields and used to address the particular module within chassis and performing various actions. Embedded software flowchart is shown in following Figure 5.



Figure: 5 Embedded Program Flowchart.

GUI using LabVIEW

То access the hardware remotely virtual instrumentation software LabVIEW is used to develop the Graphical User Interface. With this GUI and CAN connectivity the complete hardware system is visualized virtually on the computer screen. GUI is common interface for Analog Signal Conditioning hardware and HV Electronics. We have used National Instruments make USB-CAN Module NI8473 to connect CAN bus to computer through USB Port. This software also prepares log file for the HV setting and stores in remote computer. LabVIEW GUI is also interlocked with Master Control System (MCS) of SST-1. This GUI and MCS communicates with TCP/IP protocol and exchanges various plasma shot related information, like shot No., HV electronics READY status, Start of New Shot, End of Shot, Shot Abort Request/Grant. Figure 6 shows "Virtual Hardware", i.e. LabVIEW GUI screenshot.



Figure: 6 Spectroscopy GUI (HV Chassis View).

RESULTS

Usage of digital control for setting and monitoring of High Voltage results in excellent accuracy and precision. We took measurement using Keithly make digital multimeter to read the high voltage output of HV Module. The variation between set voltage and actual measured voltage is less then 1.2V for entire range from 0 to1500V. We also conducted experimental campaign of SST and tested this HV system with PMTs, Analog Signal Conditioning Electronics. We could get good results of whole spectroscopy System and the analog signals were acquired by Data Acquisition System of SST-1. PMTs were tested with external light source as well as for first Plasma Trial shot of SST-1.

REFERENCES

[1] www.xilinx.con

PROGRESS OF THE JINR e-LINAC ACCELERATOR TEST-BENCH CONTROL SYSTEMS

M.A. Nozdrin[#], N.I. Balalykin, V.F. Minashkin, V.Yu. Schegolev, G.D. Shirkov, G.V. Trubnikov, JINR, Dubna, Russia

Abstract

Due to Joint Institute for Nuclear Research participation in ILC collaboration, e-linac accelerator testbench is being created in the Laboratory of high energy physics of JINR. The bench is designed for several goals: structures and diagnostics accelerating testing, photoinjector prototype creation and investigation, radiation resistance studies of different materials etc. In addition, several proposals of FEL creation on the basis of the e-linac exist. Current setup, results of the test-bench control systems evolution since 2009 and future plans are presented. The most important updates include radiation control system calibration, verification and installation and an upgrade of the video control system.

INTRODUCTION

Linear accelerator test-bench in the Joint Institute for Nuclear Research is based on the so called MEA accelerator [1] – part of the accelerator complex which was transferred to the possession of JINR by the National Institute for Subatomic Physics (NIKHEF, Amsterdam).

In 2009 successful commissioning of the test-bench injector with beam energy of 400 keV was performed [2]. Since then a number of upgrades was done in bench control systems. Video and analog signals control system was substantially extended. Another major upgrade was calibration and verification of the Automatic system of radiation safety control gamma-detectors in JINR Radiation Safety Department and start of this system operational testing. The short description of non-upgraded since 2009 Electron gun control system is also given.

At the present time the injector (composed of electron gun, chopper, prebuncher and buncher) and the first accelerating section are assembled and put into operation. Estimated beam energy is about 25 MeV. In the short term it is planned to use the bench as a driver for the infrared FEL. The undulator of the IR range (transferred to JINR by NPO of automatic systems, Samara; E = 25 MeV, $\lambda = 18.7 \mu$ m) is going to be installed.

VIDEO AND ANALOG SIGNALS CONTROL SYSTEM

While only electron gun was in operation, the video system was composed of two IP-cameras (both are still working at the same places):

- Aviosys 9060 A is used for bench room general surveillance;
- Aviosys 9000 is installed next to the prebuncher and used for observing initial beam profile. The screen position in the beamline can be controlled remotely from the control room.

[#]nozdrin@jinr.ru

Both cameras are connected to the video control system computer by LAN. Aviosys Surf16Ch software is used for imaging.



Figure 1: Analog cameras arrangement (top view).

After installation of the two bending magnets in 2012 two more cameras were installed:

- Sun Kwang SK-2005 PH analog camera is installed next to the first bending magnet;
- Q-cam QM-68PAT analog camera can be installed in two places (manually) which are shown in Fig. 1.



Figure 2: Scheme of the VASCS.

Until October 2012 both analog cameras were connected to the video control system computer using BeholdTV 409 FM TV-tuner (with manual switching between cameras). Now the video server Aviosys IP Video 9100 is being used. This server allows simultaneous watching of up to 4 analog cameras via LAN. Currently the built-in server web-interface is being used for imaging.

The video and analog signals control system also includes several current monitors, faraday cup and two oscilloscopes (200 MHz Tektronix and PC-scope Velleman PCS500) for their signals representation.

AUTOMATIC SYSTEM OF RADIATION SAFETY CONTROL

The radiation control system was assembled at the test stand in 2009. At that moment it was composed of:

- BDS-1M-63x63 gamma radiation detection units (with energy registration range of 0.16 to 5 MeV), seven pieces:
- UDBN-01-01 neutron radiation detectors (with minimum effective dose measurement range of $0.1 - 10^4 \mu Sv/h$ and energy registration range of 0.025 - 14 MeV), two pieces;
- BPK-02 power supply and commutation unit (PCU);
- Computer with an uninterruptible power supply and RadCtrl software installed.

In summer 2010 calibration and verification of the above-mentioned gamma-detectors were conducted. The calibration was done in the Radiation Safety Department of JINR in the following way:

- Each detector was irradiated in the reference points with known dose rate. The count (amount of registered particles) was measured. The detector exposure time was set to 1 second.
- On the measurement results the piecewise-linear approximation of the dose rate to the count ratio function was done. The plot of the detectorsaveraged function is presented in Fig. 3.



• Obtained coefficients for each of the linear parts of

the function were put into the RadCtrl software.

Figure 3: Detector-averaged dependence of the dose rate (y-axis) on the count (x-axis).

Verification of the correspondence between reference dose rate values and RadCtrl readout was performed after the calibration. The difference between measured and reference values has not exceed the admissible error.

Owing to the pulse character of the accelerator operation (a relative pulse duration of $1 \times 10^4 - 3 \times 10^5$ at pulse duration of $1 - 10 \mu s$) it is necessary to correct the coefficients for the gamma-detectors. With this purpose system was complemented by the DKS-AT1123 portable X-ray and gamma-radiation dosimeter which is capable of adequately measuring the dose rate of pulse radiation with minimal pulse duration of 10 ns.

In 2011 ASCRC has been mounted at the accelerator bench and relating areas [4]. Currently the system is working in operational testing mode.



Figure 4: Scheme of the radiation control system.

RadCtrl Software

This software (Fig. 5) is programmed in Object Pascal using Turbo Delphi software, runs on Windows XP operating system and performs the following functions:

- Displaying the current readout of all radiation detectors in both numeric and graphic representation.
- Signalization when one of the two threshold values ("yellow" and "red") is crossed.
- Signalisation when a detector is disconnected.
- Displaying the object's scheme as appropriate.
- Displaying the measurements archive for a chosen detector as appropriate (archive date can be saved in MS Excel format).
- Changing of the threshold values and the chart representation (linear/logarithmic).





ELECTRON GUN CONTROL SYSTEM

Triode type DC electron gun [2, 3] with an impregnated thermionic cathode (W with 20% Ba, Ca and Al oxides) is being used at the test-bench. This is a triode-type gun with a constant high voltage at the cathode, a grounded anode, and a control electrode. Gun focusing system consists of extractor electrode and 15 anodes with forced resistive (R = 200 M Ω) potential distribution (about 30 kV per interval). The voltage of the first focusing electrode can vary from 12 to 30 kV.

Cathode Electronics

Cathode electronics is the key component of the electron gun control system. It consists of:

- Gun controller board designed for connection with control computer with GunCtrl software installed on it, reference voltages assignment and main parameters of the gun control.
- 50 kHz supply board converts input AC voltage of 187 V (50 Hz) to AC voltage of 2x65 V (50 kHz). This board provides required supply voltages to all cathode electronics elements.

- Filament supply board controls cathode filament heating current.
- Extractor pulser module provides required gun unlocking pulse (from -400 V to +5 kV).
- Focusing electrodes board controls gun first focusing electrode voltage.

GunCtrl Software

GunCtrl (Fig. 6) is the software developed to provide a user with friendly interface between the gun controller and system operator. It is programmed in Object Pascal using Turbo Delphi software and runs on Windows XP operating system.

🗊 Gun Control			_ 🗆 🗙
File Windows			
	P		
V Refs		V Real	Current
Filament Supply	1000 🚖 mV	0,144 V	0,144 A
Focusing Electrodes	0 🚖 mV	0,000 kV	
Extractor Pulser	0 🔹 mV	Errors	
Miscellanous			A
Flow Control	ON ON		
Pressure	0,000 Bar		
Temperature	0,000 C*		
Board Voltage	0,000 V		v
Stop	Program working	16	%

Figure 6: Main screen of the GunCtrl software

This software allows to set the reference voltages for filament heating, first focusing electrode and extractor.

Operator can also watch real voltage and current of filament heating, first focusing electrode voltage, pressure and temperature of the SF_6 insulating gas, cathode electronics board voltage and cooling system status.

CONCLUSION AND OUTLOOK

After modernization described control systems became more mature and at present fully cover the accelerator test-bench requirements (in particular, working in the FEL generation mode after undulator installation).

At the end of 2011 test launch of the accelerator was performed. The pulse current was 3 mA (with the pulse length of 1 μ s) with the estimated energy of 23-25 MeV. All control systems worked in normal mode.

- [1] Kroes F.B., Electron linac MEA, Compendium of scientific linacs, LINAC'96, 1996, p. 151.
- [2] N.I. Balalykin et al. Physics of Particles and Nuclei Letters. Volume 7, Number 7 (2010), 525-528.
- [3] Kroes F.B. et al., Improvement of the 400 kV linac electron source of AmPS, Proc. of EPAC'92, p. 1032.
- [4] N.I. Balalykin et al. Physics of Particles and Nuclei Letters, Volume 9, Number 4-5 (2012), 452-455.

Qt BASED CONTROL SYSTEM SOFTWARE FOR LOW ENERGY ACCELERATOR FACILITY

A. Basu, S. Singh, S.B.V. Nagraju, S. Gupta, P. Singh, BARC, Mumbai, India

Abstract

Qt based control system software for Low Energy Accelerating Facility (LEAF) is operational at Bhabha Atomic Research Centre (BARC), Trombay, Mumbai. LEAF is a 50 keV negative ion electrostatic accelerator based on SNICS ion source. Control system uses Nokia Trolltech's QT 4.x API for control system software. Ni 6008 USB based multifunction cards has been used for control and read back field equipments such as power supplies, pumps, valves etc. Control system architecture is designed to be client server. Qt is chosen for its excellent GUI capability and platform independent nature. Control system follows client server architecture. Following paper will describe the control system.

INTRODUCTION

Low Energy Accelerating Facility (Figure 1) is a 50 keV DC electrostatic accelerator operational at Van de Graaff building in Bhabha Atomic Research Centre, Trombay, Mumbai [1]. LEAF can accelerate various beams across periodic table. A sputtered negative ion source generates negative ions which are accelerated using high voltage. For this purpose entire ion source floats at negative high voltage (typically -50 kV). Extracted ion beams are bent by a 90 degree magnet which acts as mass analyzer and selects desired mass to deliver at target. Ion currents delivered at target is micro amperes of particle current for prolific beams such as C⁻ or H⁻ and for difficult beams such as Li⁻, delivered current at target is few hundred nano amperes. This accelerator has already delivered negative ion beams of Hydrogen, Lithium, Carbon, Sulphur, Tellurium, Gold, and Silver etc. This accelerator is used for various experiments in material science and atomic physics studies using ion clusters [1].

Though LEAF is a small accelerator but smooth and efficient running of this accelerator requires robust control system. To deliver the various energies and masses of the ion beams at the target it is required to tune the beam by changing various power supply voltages and currents. Control System receives operator command and set the values on field devices. Data acquired from the field devices are displayed on the GUI and used for the interlocking system. LEAF control system consists of a data acquisition and operator command execution module and Graphic User Interface (GUI) based analog and digital control interface.

One of the aims of the LEAF control system was to make it automatic, so that a good set of parameter values can be automatically loaded by the control system in a gradual and safe way without the intervention of operator.

DESCRIPTION

Analog and digital field data is acquired from the field via National Instrument's micro-controller based (NI 6008) system. It is a low cost, USB powered multifunction DAQ. It has 8 analog input channels which gives 12 bit resolution for differential connection and 11 bit resolution for single ended connection. It has a sampling rate of 10kS/s which is sufficient for the control requirement of LEAF. NI 6008 has 12 digital channels which can be configured either digital in or digital out. Digital channels, in digital out mode are used to turn on and off devices. Digital input modes reads the actual on off status of the device. Single ended analog input channels are used to read the analog field signals [2].

LEAF control and instrumentation is physically divided into two parts because Ion source of LEAF and all power supplies are floating at negative 50 kV. Non ion Source parameters are at ground potential.

LEAF control software has the client server architecture (Figure 2).To Ensure proper galvanic isolation, a dedicated server is written for the ion source side. Another server deals all the other parameters at ground potential.



Figure 1: Low Energy Accelerator Facility.


Figure 2: Control System Architecture.

Each server is having its own scan cycle. In a typical scan cycle, server first acquires analog and digital field data, code them and send it to client via LAN using connection oriented TCP/IP protocol. Next server reads the client command, if any, and decoding the command apply analog or digital signal in the USB-6008 microcontroller. In LEAF control system each server's scan cycle period is 250 milli second.



Figure 3: Server Scan Cycle.

Operator interface of the control system also acts as client. Client receives the field data from both the servers, decode them and display the field parameters on GUI. Operator can change different analog and digital values as per operational requirements and change is communicated to the server as the client command. Client side also has an interlock module. In the client cycle various interlock conditions are evaluated and either alarm is raised for operator or appropriate action according to defined rule is executed automatically.



Figure 4: Client Scan Cycle.

Software Description

Entire software is written in Qt 4.0 [3]. Server side is a timer driven console program. Qt's network module is used to communicate using connection oriented TCP/IP protocol. National Instrument's USB-6008 interaction library is imported to interact with it. Effort is taken to use

only those calls, which can also be used in LINUX operating system. Presently server side program runs in Windows 7 Home version in the ion Source server and under windows XP professional in beam line server. Servers are operating satisfactorily under both operating systems.

Client cum operator interface used extensively Qt's excellent graphical capabilities. A screenshot of the GUI is present in Figure 5.



Figure 5: Operator Interface GUI.

All relevant details of the analog and digital channels and description of all field devices are kept in the configuration files. Servers and client read relevant configuration files and sets up control program by taking data from configuration files. After initial configuration, repetitive server and client cycles described in the Figure 3 and Figure 4, keep on running till the programs are alive.

In every scan cycle collected data is logged in the file, which helps to understand the dynamics of machine and to troubleshoot.

PERFORMANCE

The control system is operational for more than six months and working as per specification and design. Automatic loading of the parameters, automatic shut down of the machine is working satisfactorily.

FUTURE SCOPE

The LEAF control system is designed to scale up as and when required. Presently operator and user feedbacks are being collected and most of them will be incorporated in the future.

ACKNOWLEDGEMENT

We acknowledge the encouragement and support received from all the users of LEAF and Dr. S. Kailas, Director, Physics Group.

- A. Basu, et al. "Operation and upgradation of Low Energy Accelerator Facility (LEAF) at BARC", InPAC 2011.
- [2] http://www.ni.com/pdf/manuals/3713031.pdf
- [3] http://doc.qt.nokia.com/4.7-snapshot/

MODELLING AND SIMULATION OF INDUS-2 RF FEEDBACK CONTROL SYSTEM

D. Sharma*, P. S. Bagduwal, N. Tiwari, M. Lad, P. R. Hannurkar RRCAT, Indore-452003, India

Abstract

Indus-2 synchrotron radiation source has four RF stations along with their feedback control systems. For higher beam energy and current operation amplitude and phase feedback control systems of Indus-2 are being upgraded. To understand the behaviour of amplitude and phase control loop under different operating conditions, modelling and simulation of RF feedback control system is done. RF cavity baseband I/Q model has been created due to its close correspondence with actual implementation and better computational efficiency which makes the simulation faster. Correspondence between cavity baseband and RF model is confirmed by comparing their simulation results. Low Level RF (LLRF) feedback control system simulation is done using the same cavity baseband I/Q model. Error signals are intentionally generated and response of the closed loop system is observed. Simulation will help us in optimizing parameters of upgraded LLRF system for higher beam energy and current operation.

INTRODUCTION

Indus-2 is a synchrotron radiation source at Raja Ramanna Centre for Advanced Technology, Indore, India. Presently it is being regularly operated in round the clock mode with stored current of 100 mA at 2.5 GeV. The role of RF system at this facility is to boost the electron energy from 550 MeV to 2.5 GeV and compensate for synchrotron radiation losses. The RF system employs four numbers of elliptical cavities to generate sufficient accelerating RF voltage, each excited by individual RF station. Each station consists of a high power RF amplifier, solid state driver amplifier & LLRF feedback loops to keep the cavity RF field stable [1].

For higher beam current and better performance of LLRF feedback control loop it is being upgraded to digital feedback system. Inphase and Quadrature (I/Q) modulation technique is being implemented with FPGA based digital controller. To understand the behaviour of I/Q based feedback LLRF control system under different operating conditions, modelling and simulation is done. Modelling and simulation also helps in algorithm study, controller design and optimization. Baseband simulation has been performed as it makes the simulation faster due to better computational efficiency and for its close correspondence with the actual implementation. A description of the models and corresponding results has been discussed.

*dheerajsharma@rrcat.gov.in

RF CONTROL SYSTEM MODELLING

RF Cavity Model

Resonant modes of cavity can be described by means of resonant LCR circuits. In equivalent LCR model it is assumed that voltage across the resistor is the cavity gap voltage. A parallel LCR circuit with an input current source is shown in Fig. 1.



Figure 1: Parallel LCR resonant circuit.

The power source feeding power to RF cavity can be modeled as an LCR circuit driven by a current source I_g through transformer as a coupler [2]. An RF cavity resonator coupled by means of a *1*:*N* transformer [3] to an input RF power source with internal impedance Z_o is shown in Fig. 2 and the same is simplified in Fig. 3.



Figure 2: RF cavity coupled to an RF power source.

I		c T	$R_{eq} $ $\stackrel{+}{\underset{-}{\overset{+}{\overset{+}}}} V$
---	--	--------	--

Figure 3: Equivalent circuit of Figure 2.

Where R_{eq} represents the equivalent impedance of the cavity under loaded condition. Cavity gap voltage is given by Eq. (1).

$$I = \frac{1}{L} \int V dt + C \frac{dV}{dt} + \frac{V}{R_{eq}}$$
(1)

Eq. (1) can also be rewritten in terms of bandwidth $\Delta \omega$ and resonance frequency ω_o as Eq. (2):

$$\frac{d^2 V}{dt^2} + \Delta \omega \frac{dV}{dt} + \omega_0^2 V = \frac{1}{C} \frac{dI}{dt}$$
(2)

Laplace transform of Eq. (2) is given by

$$\frac{V(s)}{I(s)} = \frac{\frac{S}{C}}{s^2 + \Delta\omega s + \omega_0^2}$$
(3)

Eq. (3) can be used for finding the transfer function of the normal conducting RF cavity of Indus-2. Different parameters of Indus-2 RF cavity are given in Table 1.

Parameter	Symbol	Value
Unloaded Quality Factor	Q	40000
Loaded Quality Factor	Q_L	10000
Resonant Frequency	f_0	505.812 MHz
Shunt Impedance	R	6.6 MΩ

Table 1: Indus-2 RF Cavity Parameters.

After using the parameters given in Table 1 the transfer function of Indus-2 RF cavity is given by Eq. (4)

$$\frac{V(s)}{I(s)} = \frac{0.5235464 \times 10^{12} s}{s^2 + 0.3173008 \times 10^6 s + 0.10067984 \times 10^{20}}$$
(4)

RF Cavity Baseband I/Q Model

As the new control system will be based on I/Q modulation technique Inphase and Quadrature baseband response of the cavity has been modeled. This facilitates the simulation and designing in following ways:

- Controller is to be designed for I and Q signals.
- Only amplitude and phase of RF signal is of concern.
- Simulation gets faster as high frequency carrier information is removed.

Eq. (2) using can be represented in terms of half bandwidth of cavity as Eq. (5):

$$\frac{d^2 V}{dt^2} + 2\Delta\omega_{\frac{1}{2}}\frac{dV}{dt} + \omega_0^2 V = 2\Delta\omega_{\frac{1}{2}}R_{eq}\frac{dI}{dt}$$
(5)

Further input RF current *I*, and cavity output voltage *V*, in terms of vector notations can be represented as follows:

$$I = Ie^{j\omega t} = (I_r + jI_i)e^{j\omega t}$$
(6)

$$V = \vec{V}e^{j\omega t} = (V_r + jV_i)e^{j\omega t}$$
⁽⁷⁾

As the envelope of the cavity voltage V, is a slowly varying function of time compared to the time period of the RF oscillations, the following assumptions can be made

$$\frac{d\vec{V}}{dt} << \omega \vec{V} \quad \& \quad \frac{d^2\vec{V}}{dt^2} << \omega \frac{d\vec{V}}{dt}$$

For operation of cavity near its resonance frequency i.e. $\omega \approx \omega_0$ with the above assumptions Eq. (5) can be written as Eq. (8):

$$\frac{d\vec{V}}{dt} + \left(\Delta\omega_{\frac{1}{2}} - j\,d\omega\right)\vec{V} = \Delta\omega_{\frac{1}{2}}R_{eq}\vec{I}$$
(8)

Where $d\omega (= \omega_0 - \omega)$ is called "detuning parameter".

After separating real and imaginary part one set of coupled differential equations are found which represent the baseband I/Q model of RF cavity as follows:

$$\frac{dV_r}{dt} + \Delta\omega_{\frac{1}{2}}V_r + d\omega V_i = \Delta\omega_{\frac{1}{2}}R_{eq}I_r$$
(9)

$$\frac{dV_i}{dt} + \Delta \omega_{\frac{1}{2}} V_i - d\omega V_r = \Delta \omega_{\frac{1}{2}} R_{eq} I_i$$
(10)

As the high frequency information is not present in the baseband I/Q model of the cavity the simulation becomes faster without losing the precision.

For Indus-2 RF cavity the above equations are as follows:

$$\frac{dV_r}{dt} + 158.65043 \times 10^3 V_r + d\omega V_i = 26.177321 \times 10^{10} I_r \quad (11)$$
$$\frac{dV_i}{dt} + 158.65043 \times 10^3 V_i - d\omega V_r = 26.177321 \times 10^{10} I_i \quad (12)$$

Comparison of Cavity RF Model and Baseband I/Q Model

A model for comparing the response of the cavity RF model and baseband I/Q model at resonant frequency is shown in Fig. 4.



Figure 4: Comparison of cavity baseband and RF model.

In Fig. 4 an RF current source of amplitude 1A, phase 45° (I_r = 0.707A & I_i = 0.707A) and frequency 505.8 MHz is applied as input to the cavity RF model and baseband I/Q model simultaneously. Output response from both the models shown in Fig. 5, clearly indicates the close correspondence between them. The response from baseband model is envelope of the response from RF model.

The cavity peak voltage obtained from the simulation at steady state is 1.648 MV which is very close to the expected value of 1.65MV (R_{eq} *I). The cavity filling time t_{fill} is found to be 6.295 µs.



Figure 5: Comparison of simulation waveforms.

Correspondence of cavity RF model and baseband model for amplitude and phase response is also verified by the bode plots as shown in Fig. 6 and Fig. 7.

Status Report/Overview of Control System







Figure 7: Phase plot of RF and baseband model.

LLRF Feedback System Modelling

Finally the baseband I/Q model of RF cavity is used for complete LLRF feedback control system modelling. From Eq. (9) the baseband transfer function for real part with zero detuning is given by Eq. (13) which is also same for imaginary part.

$$\frac{V_r(s)}{I_r(s)} = \frac{R_{eq}}{(\frac{1}{\Delta \omega_{\gamma_2}})s+1}$$
(13)

A PI controller has been designed for in phase response of cavity and same is used for quadrature response due to the symmetry at resonance. Fig. 8 shows a unity feedback system with PI controller for the transfer function of Eq. (13) with 15 dB step error.



Figure 8: Unity feedback system with PI controller for inphase response of cavity.

For the applications like Indus-2 RF cavity operation, if we restrict the maximum overshoot up to 5% of the final field amplitude (i.e. 420 kV) then the value of proportional and integral constants can be calculated as k_p = 0.00001 and k_i = 0.8 respectively. Different operating conditions like beam loading, temperature variations etc can easily be modelled by changing the detuning parameter $d\omega$ in the baseband I/Q model of the cavity.

For cavity operation at resonance the LLRF feedback model is shown in Fig. 9. The derived cavity baseband

I/Q model has been put as a subsystem block. Setting input amplitude and phase as 400 kV and 45° respectively after reaching steady state a step error (15 dB) has been introduced. The response of LLRF feedback system model shown in Fig. 10 closely matches to the expected behavior with PI controller. The same baseband LLRF model can be used for designing and optimization of the system for various operating conditions.



Figure 9: Baseband LLRF feedback control system.



Figure 10: Cavity voltage amplitude (step error at 80 µs).

CONCLUSION

This simulation is a helpful tool to model, design and optimize the closed loop LLRF control system under different operating conditions of RF cavity. Similar model will also help in studying the behaviour of I/Q based digital LLRF system for various controlling algorithms.

ACKNOWLEDGMENT

We acknowledge valuable discussions with Dr. Mangesh B. Borage, RRCAT. Authors are also thankful to Shri Ashif Reza, Shri Joydeep Jana and Shri S. J. Buhari for their help.

- Mahendra Lad, Nitesh Tiwari, Pritam S. Bagduwal and P. R. Hannurkar "Commissioning & Optimization of Indus-2 RF System for 2GeV/100mA" InPAC-11, IUAC New Delhi, Feb15-18, 2011.
- [2] Tomasz Czarski, Krzysztof T.Pozniak, Ryszard S.Romaniuk, Stefan Simrock, "TESLA cavity modelling and digital implementation in FPGA technology for control system development," TESLA-FEL Report 2006.
- [3] A. Neumann, J. Knobloc, "Cavity and Linac RF and Detuning Control Simulations" Proceedings of SRF 2007, Peking Univ., Beijing, China.

AN EMBEDDED SYSTEM BASED COMPUTER CONTROLLED PROCESS AUTOMATION FOR RECOVERY AND PURIFICATION OF 99m Tc FROM $(n, \gamma)^{99}$ Mo

Anirban De^{*}, S.S. Pal, P. Bhaskar, S. Kumari, V.K. Khare, A. Duttaroy, M. Garai, S.K. Thakur, S. Saha, VECC, Kolkata, India

Sankha Chattopadhyay, Luna Barua, Sujata Saha Das, U. Kumar, M.K. Das, BRIT, Kolkata, India

Abstract

⁹⁹Mo produced ^{99m}Tc ($t_{1/2}$ =6hr, 140keV γ -ray) is the most useful radioisotope for nuclear diagnostics. High specific activity ⁹⁹Mo is supplied globally mainly by five old reactors whose routine or unscheduled maintenance shutdown causes supply irregularities that adversely affects patient management in nuclear medicine centres. 99m Tc may also be produced via ${}^{98}Mo(n,\gamma)$ in a natural MoO₃ target in reactor or by ${}^{100}Mo(n,2n){}^{99}Mo$ or ${}^{100}Mo(p,2n){}^{99m}Tc$ reaction in cyclotron. To meet the crisis proposals are there to produce ${}^{99}Mo$ by ${}^{100}Mo(n,2n){}^{99}Mo$ or ${}^{99m}Tc$ directly by ${}^{100}Mo(p,2n){}^{99m}Tc$ in a cyclotron. Of the several separation methods of ^{99m}Tc from molybdenum, the most common are adsorption column chromatography, sublimation and liquid-liquid solvent extraction. The conventional methods besides being cumbersome are often hazardous, polluting, require skilled manpower and facilities like fume hood and so are not always practically feasible for hospitals. To address these, VECC and BRIT, Kolkata have collaborated to develop an embedded system based automated ⁹⁹Mo/^{99m}Tc generator from low specific activity ⁹⁹Mo using solvent extraction technique, supervised by a PC based GUI.

INTRODUCTION

Technetium-99m (t_{1/2}= 6.02h; 140.51 keV (89%), principle γ -emission energy) is known to be the most useful radioisotope in diagnostic nuclear medicine. More than 80% of all diagnostic procedures done worldwide in nuclear medicine centre are performed with 99m Tc. There are several methods [1, 2, 3, 4, 5] for routine separation of ^{99m}Tc from ⁹⁹Mo, namely, alumina column chromatography technique based on the elution of ^{99m}Tc from high specific activity ⁹⁹Mo obtained by thermal fission of ²³⁵U, the zirconium molybdate gel column chromatography technique based on the elution of 99m Tc from the medium specific activity $(n, \gamma)^{99}$ Mo and solvent extraction technique based on extraction of 99m Tc with methyl ethyl ketone (MEK) followed by removal of MEK from low-medium specific activity (n, γ) ⁹⁹Mo. Currently the solvent extraction generators for the separation of ^{99m}Tc from ⁹⁹Mo produced by neutron activation of ⁹⁸Mo is achieved by manual solvent extraction with MEK in most of the hospitals in India and other developing countries. Though it is being practiced, but the method is not desirable in hospital set up as the MEK being inflammable solvent, the evaporation of MEK by heating is hazardous and environmentally polluting. Not only is there the requirement of additional facilities such as fume hood and trained and skilled technicians, which may not always be practically feasible for hospitals but the process is cumbersome as well. There is also the possibility of bacterial contamination in the product and radiation exposure to the operating technicians.

To solve the abovementioned problems, an automated closed cyclic module for separation and recovery of various isotopes, radioactive or non-radioactive, using solvent extraction technique, and in particular, for separation and recovery of 99m Tc from low-medium specific activity 99 Mo obtained in research reactor has been indigenously developed jointly by VECC and BRIT, Regional Centre, Kolkata. The module may also be used for separation of 99m Tc produced in cyclotron. Its principle of working, scope and advantages are explained in the following sections with a discussion of the process chemistry as well.

PROCESS CHEMISTRY

The generator system is based on the selective extraction of pertechnetate ($^{99m}\text{TcO}_4^-$) in MEK from aqueous alkaline (n,γ)Na $_2^{99}$ MoO₄ solution, subsequent purification of the organic phase by passing through an alumina column to remove traces of Mo, alkali etc. and careful evaporation of the organic phase. The residue obtained is reconstituted in physiological saline (10 ml), purified through an on-line 0.22µm membrane filter to obtain pharmaceutical grade 99m Tc and collected in a vacuum vial.

AUTOMATION OVERVIEW

Scope

The process constitutes of sequential mixing, seperation, evaporation and collection of various chemicals and their by-products. The mixing and directing the flow of chemicals to various parts of the system was achieved by a set of solenoid valves and a vacuum pump. By exploiting the difference in density and electrical conductivity of the organic and aqueous constituents, the phase separation was carried out online. The evaporation of the trace organic amount before the product collection was done by temperature controlled heater system set at the boiling point of the organic chemical. A clock system was designed to set the timings of each sequence of operation starting from the initial chemicals to the final product. the integrated design employing a microcontroller based embedded system with a compatible PC based GUI enabling the operator to programme and intervene the process remotely.

^{*} ade@vecc.gov.in

Controller Electronics

A 16-bit microcontroller (Texas Instruments make MSP430FG4619) based control architecture, a customized version of [6], constitutes the heart of the system. On the one hand it communicates with the PC based GUI over the serial link and on the other hand sets in motion the actuator circuitry. From the GUI it receives the configuration words and the process instructions. Based on these it configures a timer and also selects the actuators that are to be operated upon for the corresponding step of the process. Its digital input bank reads the actuator status and sends the update to the front panel LEDs and the GUI. Priorities are programmed in the firmware so as to make the system failsafe, e.g. command to terminate the process has been given the highest priority that will overrule any operating conditions and will shut down the process, closing all the valves, pump and heater, at any instant the particular command is received. Also on every reset or at the instant the circtuitry is powered, it drives all the actuators to their OFF state.

Sensors and Actuator Hardware

A relay based driver circuitry is designed for actuation of the solenoid valves and the pump. This section is driven by the controller whose first stage selects the actuator channel that is being operated upon and also isolates the digital circuitry from the analog one. A driver circuitry, of the corresponding channel, then triggers the associated relay that serves the dual purpose of activating/deactivating the particular actuator and offers yet another level of isolation between the driver power and the electrical power. An isolated feedback path is designed to read the status of the actuating relays and sends the data, over a single channel, to the microcontroller via a digital multiplexing arrangement.

The conductivity detector system is set at the threshold standardized by the process chemistry. It senses the electrical conductivity of the process chemicals and sends the status to the controller via a comparator block that finally dictates the operation of the valves associated with that section of the process.

A temperature controlled heater system is designed to warm up a water bath at a temperature set by the DAC of the controller. AD590 based instrumentation senses the temperature and is utilized as feedback to regulate the temperature of the heater system. The kanthal wire based 180W heater coil was developed for the purpose and is powered by a 30V DC supply. The water bath sets the upper limit of the heater temperature, which is normally operated at temperature higher thatn the boiling point of MEK. The temperature setting is done by DAC, set by the controller and amplified according to the electronics requirement and the readout is buffered from the AD590 feedback signal and conditioned before feeding to the ADC.

GUI

ď

2012

0

A Visual Basic based GUI was designed (Figure 1) on defined RS-232 based serial communication protocol between the embedded controller and the PC. It gives the user access to change the timings of the sequence of the process and start, halt/resume or terminate it as well. The process is also supervised via animated objects that informs the conductivity status of the process chemical as read by the detector, the status of the actuators and the temperature of the water bath. Manual checks and unavoidable operator interventions during the process are reminded by the GUI in the form of flashing comments as and when required. After each cycle completes, the GUI enforces the operator to refresh the heater setting and the process timings.



Figure 1: Operator Interface of the PC based GUI.

The important benefits that the system offers are

- As the greater part of the process involves computer control, there is no or little chance of exposing the operator to radiation dose.
- · The overall assembly including the solenoid valves can be conveniently used in hospital radiopharmacy.
- · The used MEK after extraction may be reused for next extraction, excluding the disposal problem.
- As the organic and aqueous phase separation is controlled by a conductivity detector there is no or very less chance of MEK remaining in the aqueous layer.
- As the aqueous layer containing ⁹⁹Mo after ^{99m}Tc extraction is unloaded from the solvent extractor a fresh loading of ⁹⁹Mo/^{99m}Tc may be done for another extraction and purification of 99m Tc at the same day. The absence of volume restriction improves the versatility of the system.
- The process of air bubbling through the product at elevated temperature to remove traces of MEK from the final ^{99m}Tc product proved to be more effective than the conventional way of air flow at room temperature.
- The controller has been configured to include 16 actuators but the design modularity allows a scope for extension without any major hardware modifications.
- The software provides online display of communication status and a time-stamped background log of the actuator information that helps in easier maintenance.



Figure 2: Hardware assembly consisting of (clockwise from top left) Process Assembly in fumoid, Solenoid Valve Assembly, Electronics Assembly inside the 19" Controller rack, Front panel of Controller Unit.

BENEFITS, FUTURE SCOPE AND CONCLUSION

In summary the system greatly simplifies the operation by minimizing manual intervertion while enforcing strict adherence to embedded protocol, eliminating chance of wrong procedure. This, supplemented with a fail safe approach at several levels of its design and the system economics, advocates high potentiality for its application in nuclear medicine centre. The new generator system, developed primarilyfor separation of 99m Tc from reactor produced low-medium specific activity 99 Mo, can also be used for separation of ^{99m}Tc directly produced from ¹⁰⁰Mo by (p, 2n) reaction in a cyclotron. The automated radiochemistry used is a remarkable development and it can be adapted in any radiochemical separation in general and production of new PET pharmaceuticals in particular. Thus this will be extremely useful in routine production of new radiopharmaceuticals in the upcoming medical cyclotron project in Kolkata.

It has been succesfully tested and demonstrated at RPL, BRIT, VECC, Kolkata and BRIT, Mumbai and is currently being evaluated at RMC, BARC, Mumbai.

ACKNOWLEDGEMENT

The authors acknowledge the support of Dr. D.K. Srivastava, Director, VECC, Dr. A.K. Kohli, CE, BRIT and Dr. M.G.R. Rajan, Head, RMC, BARC in realizing the system. Acknowledgement is also due to Mr. Yashwant Kumar for the photographs of the various sub-systems.

- [1] S. Chattopadhyay, M.K. Das, B.R. Sarkar, N. Ramamoorthy, "Separation of pertechnetate from molybdate by anion-exchange chromatography: recovery of ^{99m}Tc from $(n,\gamma)^{99}$ Mo and suitability for use in central radiopharmacy (CRPh)", Radiochim Acta 2002, 90, 41721.
- [2] S. Chattopadhyay, S. Saha Das, M.K. Das, N.C. Goomer, "Recovery of 99m Tc from Na2[99Mo]MoO4 solution obtained from reactor-produced $(n,\gamma)^{99}$ Mo using a tiny Dowex1 column in tandem with a small alumina column", Appl Radiat Isot 2008, 66, 1814-17.
- [3] S. Chattopadhyay, S. Saha Das, M.K. Das, "A novel technique for the effective concentration of 99m Tc from a large alumina column loaded with low-specific-activity (n, γ) ⁹⁹Mo", Appl Radiat Isot 2008, 66, 1295-99.
- [4] S. Chattopadhyay, S. Saha Das, L. Barua, "A simple and rapid technique for recovery of ^{99m}Tc from low specific activity $(n,\gamma)^{99}$ Mo based on solid-liquid extraction and column chromatography methodologies", Nucl Med Biol 2010, 37 (1), 17-20.
- [5] S. Chattopadhyay, S. Saha Das, L. Barua, "A simple and rapid technique for recovery of ^{99m}Tc from low specific activity $(n,\gamma)^{99}$ Mo based on solvent extraction and column chromatography", Appl Radiat Isot, 2010, 68(1), 1-4.
- [6] Anirban De, "Design of a Generalized and Modular Architecture for Embedded Controller for Power Supplies", Seminar on Applications of Computer & Embedded Technology, VECC, Calcutta, India, Oct 28-29, 2009, pp. 135-137.

INSTRUMENTATION ARCHITECTURE FOR ITER DIAGNOSTIC NEUTRAL BEAM POWER SUPPLY SYSTEM

A. Thakar, H. Dhola, R. Dave, N.P. Singh, B. Raval, D. Parmar, A. Patel, S. Gajjar, V. Gupta, U. Baruah, ITER-India, IPR, Gandhinagar, India L. Svensson, J.Y. Journeaux, D. Lathi, B. Schunke ITER Organisation, St Paul Lez Durance, France

Abstract

A Neutral Beam (NB) Injection system is used for heating or diagnostics of the plasma in a Tokamak. The Diagnostics Neutral Beam (DNB) system for ITER (International Thermonuclear Experimental Reactor) based on acceleration of negative ions; injects a neutral (H_a) beam at 100keV with specified modulation into the plasma for charge exchange recombination spectroscopy. DNB Power Supply (DNBPS) system consists of various high voltage power supplies, high current power supplies and RF Generators. The system operates in a given operating sequence; very high electromagnetic transients are intrinsically generated during frequent short circuit at the accelerator grid (breakdowns) and sudden loss of load (Beam off).

Instrumentation is to be provided to operate the DNBPS system remotely with required control and protection in synchronisation with ITER operation as directed by CODAC (COntrol Data Access and Communication); the central control system for ITER. Instrumentation functionality includes 1.Operation and control of DNBPS subsystems and associated auxiliaries 2.Protection of DNB components and power supplies using interlock system, 3.To ensure safe operation of high voltage hazardous systems 4. Acquisition of injector performance parameters and 5.To facilitate test and maintenance of individual subsystem.

This paper discusses about proposed DNBPS instrumentation architecture. The design generally follows the protocols from the ITER- Plant Control Design Handbook (PCDH).

INTRODUCTION

The DNB Power Supply system [1,2] feeds the required controllable electrical power to the DNB beam source (BS), the Residual Ion Dump (RID) and the Active Correction and Compensation coils (ACCC). The system comprises of various high voltage, High Current power supplies and RF generators for plasma generation in the ion source, with integrated controllers.

Sub Systems	Controlling parameter	Measurements	Protection events
Acceleration Grid PS (AGPS)	Voltage, Modulation	Voltage, Current, Cooling water parameters	Breakdown, Beam off, Failure in RID voltage, Short circuit at power supply end
96kV, 75A			
Extractor Grid Power Supply (EGPS)	Voltage, Modulation	Voltage, Current, Cooling water parameters	Breakdown, Beam off, Short circuit at power supply end
12kV, 140A			
Residual Ion Dump Power supply (RIDPS) 8kV, 60A	Voltage, Operation time	Voltage, Current, Cooling water parameters	Short circuit
RF Generators 4 X 200kW	Frequency, RF power, Modulation	Frequency, Phase, Power, Cooling water parameters	Break down, Beam off (Reduce RF power to notch level)
Active Correction Coil Power Supply (ACCPS) 1.4kV, 440A	Current	Current, Magnetic field around DNB cell, Cooling water parameters	Short circuit protection
Plasma Grid Bias Power Supply ; 30V, 600A	Current, Voltage	Current, Voltage, Cooling water parameters	Short circuit protection
Plasma Grid Filter Power Supply;15V, 4kA	Current, Voltage	Current, Voltage, Cooling water parameters	Short circuit protection
Cs oven	Shutter control, Temperature	Temperature	-

Table 1: DNBPS Main Subsystems, their control, monitoring and protection parameters

The DNB Injector is installed in the main Tokamak building whereas the DNBPS sub systems are installed in adjacent buildings approximately 130 metres from the injector itself. The connection between the DNB Beam source and their PSs will be done through a HV transmission line.

Table 1 gives different subsystems of DNBPS with major specifications and required measurements, controls and protection against possible damaging events.

BASIC DESIGN REQUIREMENTS

DNBPS instrumentation architecture is mainly devised in compliance with ITER PCDH. Plant systems should interface with CODAC using Plant operation, Timing and Data network. These networks are managed by CODAC including assignment of IP addresses. DNBPS Controller acts as a master controller for different DNBPS subsystems and provides single point interface with the CODAC.



Figure 1: DNBPS Operating Sequence.

Major steps for DNB operation, to be taken care by instrumentation are:

Preoperational preparation of different sub systems

and auxiliaries.

- To start beam operation, start different sub systems in a predefined sequence.
- During operation modulate the beam using AGPS and EGPS, control beam energy and monitor performance.
- Stop/Restart the beam.

All power supplies need synchronized operation; the relative time sequence for major power supplies is as shown in Figure 1.

Few power supplies require local controller with real time operating system and very deterministic signal generations by Fast controller with control loop rate faster than 100 Hz. High voltage auxiliaries like switchgears and transformers need to be monitored as well, which can be done by slow controller (like PLCs) working at a rate of less than 100Hz. Hence, DNB instrumentation design consists of

- 1. Fast controller for control and data acquisition
- 2. Slow controller for control and data acquisition
- 3. Sensors, actuators and signal conditioning modules

In addition to above, the instrumentation has to take care of investment protection and personnel safety in coordination with central interlock and central safety system respectively.

I&C FUNCTIONAL AND PHYSICAL LAYOUT

Figure 2 shows conceptual functional and physical layout of DNBPS instrumentation for operation, control and data acquisition. (Interlock and safety part of architecture is excluded in this Figure).

DNBPS controller in Figure 2 is a master controller with single point interface with CODAC. ISEPS



Figure 2 : DNB I&C functional and physical layout.

controller is a slave controller working at ground potential of 100kV; takes care of ISEPS and a group of other power supplies operation. Ion Source related sensor signals routed through transmission line are acquired here. ISEPS controller communicates to DNBPS controller through optical Ethernet.

EGPS, AGPS and RF generators need synchronous operation with tolerable delay of 100μ S and also synchronized switch off in case of break down events.

The instrumentation shall be designed to facilitate DNB operation in two different modes.

- 1. Local operation: for testing of individual system (commissioning mode) and maintenance
- 2. Remote operation: operation by central control system or CODAC in two different modes.
 - Beam interception on calorimeter (conditioning mode)
 - Beam injection into the Tokamak (injection mode)

EMI ISSUES

The ISEPS controller instrumentation is to work on 100kV ground reference which is supposed to vary during power supply modulation. Moreover it should be immune enough to work in presence of RF radiations likely to be generated from RF generators.

During conditioning and injection operation of the DNB injector, breakdowns can occur frequently and unpredictably. This event demands the fast switching off of AGPS and EGPS to prevent possible damage to the grids. Switching transients is the major cause of electromagnetic interference in addition to stray magnetic field from the tokomak.

Hence the instrumentation design needs to be of stringent quality class. The design shall follow applicable IEC standards viz 61000, 61158, 61508, 61511, 61069, 60709 and IEEE 802.3.

Neutral beam cell environment at ITER site is likely to be with nuclear radiations; the injector related sensor signals need to be transmitted up to the safe distant location for acquisition.

INSTRUMENTATION INTERFACE WITH CODAC

Plant system (different subsystems of ITER) instrumentation is needed to be connected to CODAC. ITER CODAC is the integrated control, data access and communication system; responsible for coordinating all the plant systems. Hence the plant system instrumentation should be compatible with CODAC.

DNBPS Instrumentation Architecture is divided in three layers; Presentation, Control and Equipment as shown in Figure 3. Presentation layer is central control system like CODAC, central interlock system and central safety system. Control layer contains PCDH compliant instrumentation like Plant System Host (PSH), Fast controller and Slow Controllers. Equipment layer contains local controller of different Power supplies. Most of the controllers in control layer run CODAC Core System based on EPICS with Linux Platform like PSH and Fast controller. Slow controllers are PLCs.

Instructions required to drive the DNBPS controller (like Start, stop, state change) shall be sent by CODAC using EPICS.; DNBPS controller may communicate to PSH for error signal, protection signal and other information exchange.



Figure 3: DNBPS I&C Interface with CODAC.

Fast controller can be connected to CODAC directly or through PSH. Slow controller and PSH/DNBPS controller may communicate to each other through standard protocols. PLCs communicate with Remote I/Os with standard TCP/IP protocols.

For local operation HMIs may be needed with each sub system controller. PSH/DNBPS Controller can observe total operation of Slow Controller and log the data. PSH is having EPICS IOC to operate slow controller. PSH has device driver support for slow controllers & fast controllers.

AGPS and RIDPS are directly under DNBPS controller. ISEPS Controller is a slave of DNBPS controller. DNBPS controller transfers CODAC commands to ISEPS controller. ISEPS controller then instructs EGPS, RFPS and Other PS system accordingly. Thermocouples data acquired by ISEPS controller can be transferred to CODAC through DNBPS controller. Some optical Hardwired synchronization shall be provided to AGPS, EGPS & RFPS for synchronous operation during breakdown and beam off events.

Status of DNB subsystems shall be monitored by CODAC through single point contact; i.e. DNBPS controller.

- [1] R. Hemsworth et. al, Status of the ITER heating neutral beam system, Nucl. Fusion 49 (2009).
- [2] Lennart Svensson et.al, Instrumentation and diagnostics for the ITER Neutral Beam System, Fusion Engineering and Design 86 (2011).

ELECTRON CYCLOTRON RESONANCE ION SOURCE CONTROL SYSTEM

H. M. Kewlani^{*}, P. Roychowdhury, D.P. Chakravarthy, L. Mishra, S.H. Gharat, K.C. Mittal Bhabha Atomic Research Centre, BARC, Mumbai, India

Abstract

The ECR Ion source control system is a computer based control system. Main components of the ECR ion source are microwave generation, plasma chamber, solenoid magnets and power supplies, extraction electrodes and power supplies, beam measuring device and vacuum system (see Figure 1). All electronics devices have their built in microprocessor base electronic interface, which can be remotely accessed by serial or Ethernet link. Two Ethernet to four port serial converter are used to extend the serial port of the computer. Serial interface of all the devices are connected to the extended serial ports of the computer. A serial link of high voltage power supplies have provided optical isolation using serial to optical converter to overcome EMI and EMC problems. The software has been developed in house for remote operation of the ECR ion source.

INTRODUCTION

An ECR proton ion source has been developed for the Low Energy High Intensity Proton Accelerator (LEHIPA) [1]. The ion beam current of 42 mA (unanalyzed) has been extracted at 40 keV of beam energy. The three electrode extraction geometry has been used for ion extraction. For reliable, stable and longtime operations of the ion source it is mandatory to monitor forward and reflected microwave power, gas pressure, magnetic field and the beam parameters. The ECR ion source is sub divided in to microwave section, solenoid coil and its power supplies, high voltage power supplies, vacuum system and beam measuring device (Faraday cup). Remote operation of all section has been done using computer. Detailed control system is discussed in control system design.

CONTROL SYSTEM ARCHITECTURE

The control system of ECR ion source is computer based control system. Control system architecture is shown in Figure 2. Industrial Ethernet switch MOXA EDS208 [2] is connected to the computer. Two MOXA 5450I ethernet to 4 port serial converter is used to extend the serial port of the computer via Ethernet switch. All the five sections of ECR ion source are

Microwave Section

A 0-2 kilowatt microwave generator (magnetron) is used to feed microwave power to the plasma chamber. Microwave generator is remotely accessed via a serial RS232 link. For monitoring of the microwave power a dual channel RF power meter is used which is also remotely accessed via serial RS232 link. Both microwave generator and RF power meter assigned their dedicated serial port which is shown in Fig. 2.





Solenoid Magnet Power Supplies

For generating electron cyclotron resonance condition two solenoid coils are used to produce axial magnetic field of 832 Gauss on plasma chamber. High current power supplies of 800A, 10 V are used to supply high current in solenoid coils. Both high current power supplies are remotely accessed and have their dedicated comport.

High Voltage Power Supplies

Two high voltage power supplies rated 50 kV, 100 mA and -2kV, 100 mA are used for ion beam extraction. Both high voltage power supplies have given input power via isolation transformer to overcome ground loop problem. High voltage power supplies have remoter operation feature of 0-5 V analog input voltage type. Like 0-5 V analog input correspond to 0 - 50kV or 0 - (-2kV) V_{out}. Remote operation of HVPS is done using RS485 serial interface. Serial to optical conversion using MOXA TCF 142 has been done to provide isolation of HVPS and automation computer. An ADAM 4017+ [3] is used to provide analog output 0-5 V to HVPS. For voltage read back of 0-5V ADAM 4024 analog input module is used. Switching between remote and local mode is done using an ADAM 4068 relay module. All these ADAM modules are controlled from a single RS485 serial interface link.

^{*}kewlani@barc.gov.in



Figure.2: Control system architecture

Beam Current and Vacuum Monitor

The faraday cup is used for continues ion beam current measurement. A current shunt of 100 ohms is used and the voltage across it is measured using an ADAM 4024 analog input module. Vacuum gauges are used to monitor vacuum level and are connected to a vacuum gauge controller. A vacuum gauge controller is remotely accessed via a serial RS232 link.

Control System Software

The ECR control system is a computer based control system. Graphic user interface can be developed in any platform like QT, JAVA.

Conclusion

The ECR ion source control system up to the proton beam extraction has been designed and developed. Future part of the system is the development of the control system for the ECR ion source beam line.

ACKNOWLEDGMENT

The author wishes to thank Dr. L.M. Gantayat, Director, BTDG, BARC for his keen interest and support for this work. The author also wishes to thank Shri R.B. Chavan (SO/C, APPD) and Shri Sudhir Singh (SO/F, IADD) for their guidance during the work.

- P. Roychowdhury and D. P. Chakravarthy, "High intensity electron cyclotron resonance proton source for low energy high intensity proton accelerator", Rev. Sci. Instrum. 80, 123305 (2009).
- [2] www.moxa.com
- [3] www.advantech.com

THE CS FRAMEWORK AS A CONTROL SYSTEM FOR THE HITRAP FACILITY AT GSI

D. Neidherr^{*}, D. Beck, H. Brand, F. Herfurth, GSI Helmholtzzentrum für Schwerionenforschung GmbH, 64291 Darmstadt, Germany

Abstract

At GSI Darmstadt the linear decelerator HITRAP is currently under commissioning. The aim is to decelerate heavy, highly charged ions up to U⁹²⁺-, which come from the experimental storage ring (ESR) at an energy of 4 MeV/u, in several distinct steps down to less than one meV. In the end, bunches of up to 10^5 ions are captured and stored in a Penning trap from where they can be delivered to different experiments, which perform for example studies for ion-gas and ion-surface interactions, precision mass measurements, laser spectroscopy, or gfactor measurements. The low-energy part of HITRAP is controlled by an adapted version of the CS framework developed at GSI which is an object-oriented, eventdriven and multithreaded framework, based on National Instruments LabVIEW 2009 and designed for small and medium size experiments. The object-oriented approach of the CS framework makes the system very flexible with respect to changes in the configuration. The DIM protocol used as communication layer makes it possible to distribute it across a large network. For HITRAP new classes were implemented in order to read out a new type of detector and to fulfil the specific needs of this facility. This paper will give a short overview about those developments.

THE HITRAP FACILITY

The HITRAP [1] project at the GSI Helmholtzzentrum für Schwerionenforschung in Darmstadt/Germany is dedicated to make heavy, highly-charged ions accessible to precision experiments. The bare U^{92+} core represents the strongest electric field which can be produced in the laboratory. Therefore experiments with such ions at rest can test a number of physical theories, like the QED with an enormous precision [2].

HITRAP is installed behind the experimental storage ring ESR and is composed of three main parts: (i) a linear deceleration stage consisting of an IH-structure and a 4rod RFQ which decelerates the beam coming from the ESR at 4 MeV/u down to 6 keV/u, (ii) a cylindrical Penning trap to accumulate and cool down the ions via electron and resistive cooling to 4 K, and (iii) a section which transfers the cooled ions to the different experiments. A sketch of the setup is shown in Fig. 1. Parts (ii) and (iii) are controlled with a control system based on the CS framework.

The ions from the ESR reach the HITRAP facility with a repetition rate of not larger than one shot every 30 seconds. Once decelerated in the linear section they are

*D.Neidherr@gsi.de

injected in the Penning trap where different cooling techniques are applied [3]. For beam monitoring at HITRAP in general a micro-channel plate (MCP) detector together with a phosphor screen and a CCD camera is used. These detector assemblies are mounted on a moveable feedthrough together with a faraday cup for an absolute current measurement. Because all with the setup connected devices are controllable within the control system, automated parameter scans can be performed, which improves the optimization procedure a lot.

THE CS FRAMEWORK

The CS framework [4,5] was developed at GSI/Darmstadt in order to provide a common platform for different experiments. Especially setups with a lot of ion optical elements and detectors can very easily benefit from the work already put in the CS. It is based on native LabVIEW from NI and uses object-oriented techniques in order to archive the needed flexibility for a general framework. The different objects use an event-driven mechanism to communicate with each other making it even possible to reconfigure the system on the fly. A class can for example represent a certain type of device and is used as a pattern to create objects, which are initialized with the help of a configuration database. The Data Logging and Supervisory Control (DSC) module of LabVIEW adds also an intrinsic possibility for alarming and historical trending.

Today, mainly trap experiments like SHIPTRAP [6] at GSI, ISOLTRAP [7] and WITCH [8] at CERN, TRIGATRAP [9] at the University of Mainz, LEBIT [10] at NSCL/MSU, PENTATRAP [11] at MPIK/Heidelberg, but also experiments like PHELIX [12] at GSI or Polaris [13] in Jena trigger further developments of the framework. In general, the similarities between these experiments make it easy to reuse code from one experiment at another one. This means that adding a new device to a setup is very often not linked with additional programmers work, an advantage since at many of these experiments PhD students have to maintain the control system additionally to their scientific work. Already available are classes for many different power supplies, frequency generators but also other devices such as a pulsed pattern generator on an FPGA card [14], capable of producing pulses with a precision of several ns, or a sequencer class to control the behaviour of an automated measurement.

Comparable to the situations at the other experiments, several specific add-ons had to be implemented for HITRAP.



HITRAP SPECIFC ADD-ONS FOR THE CS FRAMEWORK

A lot of smaller improvements and extensions were made for the control system of the HITRAP setup. For example, the 90 degree bending magnet connecting the lower horizontal beam line with the vertical beam line works now with a P-regulation so that fluctuations in temperature and humidity should not affect anymore the transmission efficiency and resolution.

Each diagnostic chamber consists of a faraday cup and an MCP detector combined with a phosphorous screen and a CCD camera. The detectors are mounted on movable feedthroughs connected to step motors, which can be controlled within the control system.

The two main developments for the HITRAP control system were a diagnostic program to read out the camera information and link them to the control of beam line elements and a new, flexible graphical user interface (GUI). Both improve the usability of the control system and can also be used at other experiments if needed.

IMAQ Cameras Within the CS framework

With NI-IMAQdx National Instruments provides already a driver library for image acquisition. Supported are for example IEEE 1394, Gigabit Ethernet, or network IP cameras (Firewire as well as GigE cameras are used at HITRAP). This library has been already used at GSI in a feasibility study [15] to explore the possibilities of native LabVIEW OOP. For this, the so-called HGF base classes were developed. The main application for this study were small standalone tasks, but since it is object oriented it is very easy to incorporate the HGF IMAQ classes into the CS. Only a few adjustments have to be made. One of the major differences between LVOOP and the object oriented concept used in the CS framework is that LVOOP does not treat objects as entities, whereas this is necessary in the CS since there, an object is in general directly connected with a hardware device. To use passive LVOOP classes in the CS one just has to create a wrapper CS class where the desired LVOOP base class object needs to be copied into the CS attribute data. This means that this one CS class can in principle call all functions of the LVOOP base class including their derived classes by calling the dynamic dispatch VI's (C++/Java: virtual function) of the base class. As a configuration parameter the CS class just needs the name of the LVOOP class (which is in case of IMAQ classes an identifier for the used camera type) in order execute the correct overwrite VI of the derived class.

The GUI developed together with the camera object is shown in Fig. 2. It also represents an independent object in the CS and communicates with the different camera objects over DIM [16] which means that the device and the operation layer can be separated onto two different PCs.

The GUI provides all the necessary features to support the optimization procedure at HITRAP. The aim is to transport the ions available from the ESR through the different preparation steps to the experiments with high efficiency. Especially because of the high charge state this is a very difficult task, but at least the losses due to the

220



Figure 2: Layout of the Camera readout program. On the right side the current camera picture is shown, whereas the configuration parameters can be entered on the left side. On the upper part of the GUI the horizontal and vertical projections are shown.

transport should be minimized. For this, the cameras are calibrated in order to know when the beam is not only well focussed but also centred at specific focal points in the beamline.

The grid lines visible on the image are drawn with a distance of 5 mm between each other. It is also possible to mark different regions of interest where for example the beam is expected to occur. Parameters, like voltages or timings can be scanned and the resulting intensities can be plotted against the parameter values. Also two-dimensional scans are so possible which should make the optimization procedure much more convenient.

A General GUI for HITRAP

Figure 3 shows the general GUI for one part of the HITRAP beamline. Up to now, the *CS* framework has no built-in GUI. At all experiments either self-made GUIs or the so-called UserGOGs, a scripting method within the *CS* are used. The first solution poses a rather large work for the programmer, whereas the second one is very easy to use but has some severe disadvantages. In fact, the UserGOGs were implemented mainly for debugging purposes so they lack features which are mandatory for SCADA GUIs nowadays, like a proper alarm handling already at the panel of the GUI.

So the idea of the HITRAP GUI is not only to provide a proper GUI for the HITRAP facility but also to make a basis available for other experiments which are at least at the end of the commissioning stage. The responsible programmer just has to prepare a drawing of the setup, copy some controls to the front panel and configure the connection of the GUI with the device objects within the *CS*. In total, this should be much less work than starting from scratch.

SUMMARY

The *CS* framework is used at many different experiments worldwide. All these experiments create addons necessary to adopt the core of the *CS* to their needs. For the HITRAP facility at GSI, dedicated to produce heavy, highly-charged ions at very low energies, two new software components were developed: An image acquisition system and a GUI for the facility. Both can easily be integrated in the control system of other experiments so that the highly flexible basis of the *CS* framework is not affected.



Figure 3: A first version of the general GUI of HITRAP. Shown are controls mainly for power supplies and step motors. It is not the intention of this figure to show details.

REFERENCES

- [1] F. Herfurth et al., Hyp. Int. **173** (2006) 93.
- [2] H.-J. Kluge et al., Adv. Quant. Chem. 53 (2008), 83-98.
- [3] G. Maero et al., WEM2C06, Proceedings of COOL2007, Bad Kreuznach, Germany, 130-133.
- [4] D. Beck et al., Nucl. Instr. Meth. A 527 (2004), 567-579.
- [5] http://wiki.gsi.de/CSframework
- [6] M. Dworschak et al., Phys. Rev. C 83 (2010) 064312.
- [7] M. Mukherjee et al., Eur. Phys. J. A 35 (2008) 1-29.
- [8] M. Tandecki et al., Nucl. Instr. Meth. A 629 (2011) 396-405.
- [9] J. Ketelear et al., Nucl. Instr. Meth. A **594** (2008) 162-177.
- [10] S. Schwarz et al., Nucl. Instr. Meth. B **204** (2003) 507.
- [11] J. Repp et al., Appl. Phys. B **107** (2012) 983-996.
- [12] H. Brand et al., TPPA01, Proceedings of ICALEPCS07, Knoxville, USA, 163-165.
- [13] J. Hein et al., Laser and Nuclei, Lecture Notes in Physics 694, Springer, Berlin (2006).
- [14] F. Ziegler et al., Nucl. Instr. Meth. A 679 (2012) 1.
- [15] H. Brand and D. Beck, GSI Scientific report (2008) 260.
- [16] C. Gaspar and M. Dönszelmann, Proceedings of RT'93, 8th Real-Time Computer Applications in Nuclear, Particle and Plasma Physics, Vancouver, Canada (1993).

authors

OVERVIEW OF CONTROL SYSTEM FOR 30MEV RF SOURCE

R. B. Chavan[#], S. Chandan, A. R. Tillu, V. Yadav, H. Sarukte, K. P. Dixit, K. C. Mittal, BARC, Mumbai, India

Abstract

Control system for RF source of 30 MeV, 3 kW RF Linac for neutron generation is being developed. The system consists of two 15 MeV linac structures, each powered independently with klystron rated for 7.5 MW(pk)/7.5 kW(avg). Two klystron modulators of 160kV, 110A, 7µs and 250Hz feed pulsed power into the klystron, which produces RF power at 2856 MHz. The klystrons will be driven by low power RF driver amplifiers programmed for matching phase, frequency and power into the linac. Both the driver amplifiers are controlled through RS-232 Protocol. The HV pulsing and RF drive for the klystron has been interlocked with water flow, arc detector, SF6 gas pressure etc. The control system is designed using Real time embedded controller, where pulses for synchronization are being generated in FPGA. Most of the power supplies like electromagnet, HVDC, etc. are on RS-232 protocol. These power supplies are controlled via suitable RS-232 to Ethernet converter. State machine topology is being used to design the logic. The database for logging data is developed in SQL. This paper describes the details of the software implementation and hardware used to realize the control of the RF power source.

INTRODUCTION

A 30MeV, 3kW Linac is being developed. The thermionic diode gun of 85 kV, 1 A based on indirectly heated LaB₆ cathode of pierce geometry will be directly coupled to the linac system, which will inject a train of pulses into the linac at the rate of 250 PRF, each having a pulse width of \sim 7µs. The beam will be further accelerated to 30MeV by two 15MeV RF coupled cavity linac structures, each powered independently with klystron rated for 7.5MW (pk)/7.5kW (avg). The RF power at 2856MHz to the linac will be fed via a wave guide, Circulator, Dual Directional Coupler, H Bends and E Bends. Arc detector is located at the bend of the RF line to detect any arc in the RF Line. SF₆ gas at 28 psi pressure is maintained in the RF waveguide line to withstand high RF Power. A vacuum of $\sim 4 \times 10^{-7}$ torr will be maintained with the help of a turbo/sputter ion pump (TP/SIP) combination system.

SYSTEM REQUIREMENT

The major tasks to be accomplished in control system of RF Source include:

) Generation of pulses with synchronization

ii) Monitoring in real time mode of all the interlocks /

```
<sup>#</sup>ramchandra_c@yahoo.com
```

critical parameters like vacuum, Arc, water temperature etc

- iii) Control / monitoring of High voltage supplies, e-gun supplies, Electromagnet power supplies, etc.
- iv) Monitoring of RF Microwave parameters like forward power, reflected power, etc.
- v) Transmission of the trigger pulses through EMI prone area.
- vi) Protection of the controller from radiated and conducted noise.
- vii) RF phase matching between both the linacs
- viii) Data logging of the parameters

Figure 1 shows the schematic of 30MeV RF source.



Figure 1: Schematic of 30MeV RF Source.

DESIGN ASPECTS

Hardware

The system consists of Real time embedded controller with FPGA for pulse generation logic. The controller is connected to the input output modules with 8 slot chassis. As per the requirement provision for 1PGM (Pulse Generation Module) 16DI, 16DO, 8 AI, 4 AO and 4 Thermocouple channels has been given. Two 4-ports RS-232 to Ethernet converter is used for controlling RS -232 protocol devices like electromagnet power supplies, driver amplifier, RF power Meter etc.

Software

The design is based on State Machine Architecture. It gives an abstract description of the behavior of the system. This behavior is analyzed and represented in series of events that occur in one or more possible states.

222

authe

The state transition diagram for the control was prepared. It consists of 7 states viz Initialize, OFF, Warm up, Ready, HV ON, Trigger ON, Fault state and Shut down. Based on state diagram logic/flowchart was designed. Figure 2 shows the state machine diagram for implementation of the algorithm for the control system.



Figure 2: State Machine Diagram.

Initially all the ports are configured and variables are initialized. In the OFF state, it constantly monitors the user interface for the next action to take. It transits to Warm UP state upon response from the user for thyratron heating. Once the thyrtaron heating is completed, it automatically moves to Ready State. Further it moves to HV ON and Trigger ON state as per the response from the user. In any state, if fault occurs, it moves to fault state. On rectification of the fault, it moves to other state depending upon the status. The State machine algorithm was implemented using While loop and Case Structure. There are 3 modes of developing the software (i) FPGA,

(ii) Real Time and (iii) HMI (PC).

In FPGA, pulse generation logic is developed using flat sequence structure and case structure and all the I/O Nodes of the modules are accessed. The FPGA loop is timed to 40MHz to get a clock pulse of 25ns. The PRF, Pulse width, the delay and pulse Start/Stop values are received from HMI to RT program. In RT, the TON and TOFF period of the Pulse are calculated. These values are passed to FPGA. When the user clicks the Pulse Start button to generate the trigger pulses, in first cycle of pulse generation, the delay is generated in order to synchronize the output of e-gun modulator and the klystron modulator to get the accelerated beam, and then the TON and the TOFF is generated. In subsequent cycles only TON and TOFF is generated. When the PRF is changed, only TOFF is changed to achieve the desired PRF. When the command for stopping the pulse generation is issued, it completes the current cycle and stops generating the trigger pulses. The PRF can be changed in steps of 1 Hz. The delay between the two pulses can be adjusted in steps of 0.1μ s.

All the I/O Nodes of the modules are connected to the controller via chassis accessed through FPGA. The FPGA code is complied and converted to bit file to access the I/O nodes in real time module. The path of the complied bit file is given to "Open FPGA Reference" to generate the reference. To get the access to the I/O nodes in the RT, FPGA Read/Write Control commands are used. Finally to close the reference "Close FPGA Reference" is used. The variables like TON, TOFF and Delay is passed to the FPGA using FPGA Read/Write Control.

Timed loop is used in Real time module for deterministic action. In this loop, we can set the priority of the loops, execution time, etc. Code for monitoring all the interlocks is developed in timed loop. If any of the interlocks gets activated the pulses to the driver amplifier and the e-gun modulator are disabled. Hence, no RF and no high voltage output are available. The interlocks are identified and the logic is implemented in the timed critical loop of 10ms in real time embedded controller, where it ensures that the action will be taken within 10ms. 4-port RS-232 to Ethernet converter is used for controlling various RS-232 protocol devices. Each power supply is controlled in independent loop having independent queues for each device. These queues are used to store the events generated by the user on the GUI. If no event is generated, it reads the current parameters and updates it. Initially the port is open and configured with required parameter. The query / commands are sent to the power supplies in the loop for the control / monitor of the power supply. Query is sent to acquire and the command is sent to set the parameter of the power supply. Each loop is executed every 1s. This code has been developed in RT and only the parameter values of the power supplies are sent to the PC.

Finally to transmit and receive the data between realtime controller and the HMI (PC), TCP/IP Protocol is used. Here the controller is a server and the PC act as a client. TCP Listen command is used in the RT waiting for any in coming request. The PC sends the request to the controller with parameter (IP address of the controller and the port number). Once the request is accepted by the controller further communication can achieved using TCP Read and TCP Write commands. Touch screen PC is used for human interface. GUI for the human interface has been developed. All the values are transferred from RT to PC and are displayed on the GUI.

The database for storing the data like PRF, Vacuum, RF Gain, forward, Reflected etc with Date and Time is developed in SOL. The data to be stored in database is stored in local variable. These values are inserted in the database with proper SOL Ouery containing table name and column details. The time interval between the data storing is variable from 1s to 5mins. The numeric box is provided for selecting the time interval. The database has separate loop running at every 500ms. To display the stored data user selects the date and the data of that particular date is displayed in the grid. The data displayed in the gird can be exported in DOC / XL.

Analog – Digital Isolation

ADAM 3014 signal conditioning isolation module is used for protecting analog processed signal from ground loops, other electrical interferences. It provides channel channel optical isolation of 1kV DC.

Relay is used between digital channel and the field signal. It provides physical isolation between the signals.

Fiber Optic Pulse Transmission Circuit

For transmitting the TTL trigger pulse Fiber optic pulse transmission circuit was designed and developed. The TTL pulse generated by pulse generation module is fed to HFBR 1412 - Transmitter and the signal is transmitted over fiber optic to HFBR 2412 - Receiver kept near the trigger circuit of the individual system, where the fiber optic signal is converted back to electrical signal.

EMC Enclosure

As the controller will be placed in areas which have high radiated fields and conducted noise, proper EMC enclosure was selected. The rise time of 10µs pulse is ~500ns. Hence the maximum frequency component of the radiated noise is Fh = 0.35/500ns = 700 kHz. To protect the controller from the radiated noise Rittal compact Enclosure AE with 100dB attenuation from 700 kHz -1MHz was chosen. The enclosure size of 600mm W X 600mm H X 210mm D was selected in order to enclose the controller with its I/O protection circuits. Proper EMC glands are selected for taking out the cables out of the enclosure.

TESTS

Pulse generation logic was tested for synchronization between e-gun modulator, driver amplifier and klystron modulator. The pulses have adjustable pulse width from 0.1µs to 4µs and adjustable pulse delay with steps of 0.1us. Faults were simulated and the logic of the interlocks was tested successfully.

CONCLUSION

The development of the above described control system is complete. Control of few supplies is being done remotely through PC. The testing of the system with Klystron modulator on actual operating conditions is awaited.

SIMULATION ANALYSIS OF ANALOG IQ BASED LLRF CONTROL OF **RF CAVITY**

S. Basak[#], H.K. Pandey, A. Chakrabarti, VECC, Kolkata, India

Abstract

This paper presents the simulation work and results in Matlab Simulink for the analogue Inphase-Quadrature (IQ) based Low Level Radio Frequency (LLRF) control of RF cavity voltage. The RF cavity chosen here is the Radio Frequency Quadrupole (RFQ) cavity in our RIB project. All the subsystems in the IQ based RF control were modelled using the Simulink blocks/components. The envelope simulation was carried out using the IQ model of RF cavity. The PI controller was properly tuned to achieve good control performance in time. The simulation graphs showing the time evolution of the RF cavity voltage with a step change of the input reference signal is presented. The simulation graphs showing the control response time needed to correct a disturbance is presented. The simulation results showing Nichols plots of the control loop and the gain and phase margin values obtained from them are presented, which are good enough for stability considerations.

INTRODUCTION

An ISOL type RIB facility is presently being developed at VECC. At present the RIB facility comprises of a RFQ (3.4m), three rebunchers and three Linacs. A schematic of the RIB facility is shown in Figure 1. The RFO, first rebuncher, and first two Linacs operate at 37.8 MHz, while the third Linac and the other two rebunchers were designed to operate in the second harmonic at 75.8 MHz. For proper beam acceleration it is essential to regulate the RF cavity voltage at proper amplitude and phase. At present the RF cavity voltage are controlled in the conventional amplitude and phase loop control method and it is planned to upgrade the LLRF control to analog/digital IO method.



Figure 1: Schematic of RIB Facilityat VECC.

#subhasish@vecc.gov.in

In this paper the IQ based LLRF control simulation work is presented for the RFO cavity. The advantage of RFQ is its high beam transmission efficiency for low velocity, high current beams, and the fact that a single structure is used for bunching, focusing, and acceleration of ions. Some of the important RFQ parameters are presented in Table 1.

Table	1: RFQ	Parameters
-------	--------	------------

Parameter	Value
Frequency	37.83 MHz
Q _o (measured)	5393
R _{sh} (measured)	42 ΚΩ
Input energy	1.68 keV/u
Output energy	95.7 keV/u

IQ METHOD OF LLRF CONTROL

The IQ LLRF control method is based on the principle that we can control the amplitude (A) and phase (φ) of a RF voltage i.e., $V = A \sin(\omega t + \varphi) = (A \cos \varphi) \sin \omega t + \varphi$ (A sin ϕ) cos ω t = I sin ω t + O cos ω t by controlling the Inphase (I = A sin ϕ) and Quadrature (Q = A cos ϕ) components of the RF signal fed to the high power amplifier. The RF cavities are operated at a particular frequency. The advantage of this method is that I and Q components can be controlled in two paths using two set points for I and Q components. The In-Phase (I) and Quadrature (Q) signals could be obtained by passing the RF signal through a IQ demodulator device. The schematic of the analog IQ LLRF control system is shown in Figure 2.



Figure 2: Schematic of analog IQ LLRF control.

The components in analog IQ based LLRF control comprises of RF power amplifier, RF cavity, couplers (input and pick-up), IQ demodulator, IQ phase shifter, PI controllers, IQ modulator etc. In IQ method the controller processes the input I and Q components in two parallel paths and gives the appropriate output I and Q components so as to get the desired output.

Although the actual analog IQ LLRF control have the IQ demodulator (IQ modulator) devices for the downconversion (upconversion) of the RF signal to lowpass (baseband) and vice-versa. However in order to reduce the simulation time greatly we have modeled the RF cavity in IQ baseband description and performed the simulation.

The RF cavity was modeled in IQ baseband in terms of the inphase (= V_r) and quadrature (= V_r) components of cavity voltage (=V) using the standard differential equation description namely,

$$\frac{dV_i}{dt} + \omega_{1/2}V_i - \Delta \omega V_r = \omega_{1/2}R_L I_i$$
$$\frac{dV_r}{dt} + \omega_{1/2}V_r + \Delta \omega V_i = \omega_{1/2}R_L I_r$$

where $\omega_{1/2} = \omega_o/2Q_L$ is the half bandwidth of the RF cavity and $\Delta \omega = \omega_o - \omega$ represents detuning and I is the driving current.

The beam loading effects are neglected (ignored) in the simulation since the typical beam current is quite small for our RF cavities. The cavity is operated at resonance in usual 50 Ω system. The parameters of the PI controller are adjusted for the getting the desired response. The desired cavity voltage and phase is set by two I and Q parameters namely I set and Q set.

SIMULATION RESULTS

In this paper the simulation of the RFQ cavity voltage is presented for a test voltage of 12.4 kV and for synchronous phase of -30° (defined from peak of RF voltage) i.e., to say for RF phase of 60° . The simulation results for the time evolution of the RF cavity voltage and phase are shown in Fig. 3 and Fig. 4 respectively which accurately matches with the desired output voltage and phase.

The required RF power from the simulation was calculated to about 1.87 kW which closely matches with the measured power of 1.8 kW required for above voltage as was measured in a beam test [1]. In order to measure the response time of the control system a perturbation of about 15% of cavity voltage was introduced in simulation at time = 150 μ s. We observe from Figure 3 that the controller was able to correct the disturbance in about 10 us. The control system stability analysis was carried out by obtaining the Nichlos plot as shown in Figure 5.



Figure 3: RF cavity voltage amplitude versus time.

The graphs show that the cavity voltage has fast rise time (~ 10 us) and settles at the desired voltage.



Figure 4: RF cavity voltage phase versus time.

The plot shows that the locus curve is away from the critical point and also the gain margin and the phase margin are almost 13 dB and 70° respectively which indicates adequately stable system.



Figure 5:Nichols plot of the control system.

REFERENCES

 S. Dechowdhury et al., "Design and development of a radio frequency quadrupolelinac postaccelerator for the Variable Energy Cyclotron Center rare ion beam project", Review of Scientific Instruments, 81, 023301 (2010).

INTRODUCTION OF NON-STANDARD EPICS CONTROLLERS

I. Badillo, M. Eguiraun, M. del Campo, I. Arredondo, D. Piso, ESS-Bilbao Consortium, Spain J. Jugo, University of Basque Country UPV-EHU, Spain

Abstract

Although EPICS is a mature software framework, the study and validation of new configurations of EPICS systems is very valuable, since new ideas open its evolution and improvement. Therefore, the goal of the present work is to introduce new technologies under EPICS control structures and test different configurations with innovative hardware in this kind of applications. Specifically, it is intended to validate the use of non-standard EPICS controllers. This paper presents a testbench using LabVIEW together with EPICS. LabVIEW eases and speeds up the development of control structures, avoids the hardware dependent developing costs and offers almost absolute compatibility with a high variety of hardware devices used in control and data acquisition. To validate its use in a real environment, is mandatory to make a study facing this solution and EPICS standard methodology, specifically CO-DAC system used in ITER. With such objective in mind, a testbench is implemented running both configurations and its results are compared. Following this scheme, as conclusions, the next step must be to implement exactly the same hardware-level structure for both approaches to improve the comparison.

INTRODUCTION

EPICS is one of the most important and widely used control systems oriented to large scientific facilities. This is a Big Physics standard based on a middleware approach, used worldwide on large scientific plants such as particle accelerators and telescopes and and it can be described as a set of open source tools, libraries and applications to create soft real-time distributed control systems. Nowadays, it offers solutions for most of the control needs and is compatible with a large variety of hardware platforms and operating systems (Linux, Windows, VxWorks...) available in the market.

Summarizing, EPICS is a mature software framework. The study and validation of new configurations valid for EPICS systems is always a very valuable research, since new ideas open its evolution and improvement. In this sense, the main objective of the work sustaining this paper is the study of the inclusion of non-standard control structures in EPICS networks. Here, the first steps towards this direction are presented. Two main schemes are considered: the standard architecture of EPICS systems using classical IOCs under Linux machines, and a system which integrates LabVIEW and EPICS. The validation of the proposed system is made by comparing their reliability to the classical scheme. The hardware setup differs as in the LabVIEW approach I/O signals are software generated instead of using real DAQ. However, from a EPICS level view, both systems are completely analog... A testbench has been implemented for such comparing process.

The classic EPICS methodology shows a set of distributed Linux machines implementing IOCs. They are responsible for communicating EPICS with the control system tools and devices (motors, sensors, data acquisition systems, switches etc.). This approach requires the development of drivers (or equivalent mechanisms as EPICS devices) for each new device, which are the interface between EPICS records (set of process variables) and hardware (or 3rd party software). These drivers can be split into two parts: device support and driver. The first one is the interface for records and hardware independent. The second one provides low level software access to the hardware. The development of these drivers requires C language notions, EPICS knowledge (records, scan methods, Channel Access) and experience with I/O hardware (I/O registers, buses, interrupts). That means an extra effort every time a new device is added to the control structure.

The second architecture consists of using LabVIEW together with EPICS. This approach avoids the hardware dependent developing costs of the previous architecture. National Instruments (NI) hardware and software offer support for a high variety of devices and cards, and therefore, there is no need to write specific drivers. Moreover, the use of LabVIEW simplifies the development of the control structures. NI provides an EPICS server integrated in the LabVIEW solution which is based on the LabVIEW DSC module, and runs on the real time system in the PXI controller. The Real Time controller publishes EPICS PVs taking data from LabVIEW's Shared Variables. This scheme shows interesting advantages, but, before adopting this method of working, it must be validated. To do so, it is proposed to perform two parallel implementations in the testbench: the first one following the classic methodology and the other one using LabVIEW.

In the next section a thorough explanation of the two experimental setups for the validation of each proposed scheme is included, and some comments about the implementation. The last section is dedicated to the future plans for this experiment and conclusions.

EXPERIMENTAL SETUP

The experimental setup that has been developed implements a reduced local controller, which includes data acquisition, sequencing and control. The typical signals involved in a local controller are present in the emulated system. These signals will be the Process Variables (PVs) of the EPICS system. The generation of these signals can be performed by other devices such as signal generators, but, in order to have a high level of flexibility within a single system, an FPGA in the first solution as *hardware in the loop*, and software based signal emulation on the second approach have been chosen for this task. Different type of signals are generated (digital, analog), periodically changing their values as well as acting against external events in the case of the classic IOC approach. The complete set of signals consists of 1960 PVs: 400 binary inputs, 400 analog inputs and 400 calculation type PVs each approach.

As stated in the previous section, two different control systems are set up. The first one corresponds to the EPICS classical approach and the second one to the LabVIEW based solution. Both perform the same control actions and data acquisition, but the way of working is slightly different. The main settings of the test stand can be found in Table 1.

Table 1: Main Hardware and Software Settings of BothTest Stands

LabVIEW approach	EPICS classical
PXIe-1082 Chassis	PXI-1031 Chassis
PXIe-8108 Controller	PXI-8106 Controller
2.53 GHz Intel Core 2 Duo	2.16 GHz Intel Core 2 Duo
1 GB RAM	512 MB RAM
LabVIEW RT	Scientific Linux 6.1
LabVIEW 2010 SP1	EPICS R3.14.12.2

The following paragraphs describe each implementation:

- The first setup is taken as a reference. The IOC is running in a NI PXI-8106 embedded controller with Scientific Linux 6 in a NI PXI 1031 chassis.. A NI PXI-6259 DAQ card performs I/O tasks. As explained before, this device requires its own drivers, for both Linux (SL6) and EPICS. Both can be found in the ITER CODAC Core System v3.0 public version [1]. Once installed and configured, the PXI NI 6259 card is ready to be used. A plant, simulating a DC motor, is implemented as Hardware in the Loop (HIL) using NI PXI-7833R multifunction card, which includes a Virtex II FPGA. Taking advantage of the parallel computing offered by the FPGA, a complete set of signals is generated. The IOC database is defined including all the system PVs and their behavior. The number of PVs is increased sharing hardware inputs among them.
- The second hardware setup consists of a PXI chassis from National Instruments (PXIe-1082) with a PXIe8108 controller running LabVIEW Real-Time operating system. The control actions are defined in a LabVIEW (version 2010 SP1) program running in the PXI controller as well as the I/O signal set. These are written into shared variables. LabVIEW also acts as an EPICS server through the *EPICS IO Server* for LabVIEW. This publishes desired control signals to



Figure 1: Overall system description over the network.

a EPICS network via Channel Access protocol. This method allows the developer to focus on the control without worrying about drivers, since these are provided with the NI hardware and software and the process is almost transparent for the user.

The overall system is implemented in a conventional Ethernet wired Local Area Network environment in which EPICS 3.14.12.2 is present.

In order to get reliable results concerning repeatability issues, all the processed data of both systems must be archived in a database for a later study. In the present work, a HyperArchiver instance is used, which has been developed at INFN/LNL (Italy) as an extended version of the RDB Channel Archiver, but using Hypertable as its main database. Hypertable is high performance distributed non relational database focused on large scale and intensive tasks and it is an alternative to MySQL, as it is a high performance, distributed and non-relational database. Moreover, it is distributed under GNU license. The version of Hypertable used in ESS Bilbao [2] and, therefore, in the current paper, is a modified version of the original one, used in INFN/LNL. The reason that lead the ESS Bilbao team to modify the original Hypertable was the necessity of managing efficiently arrays of variables. It has been also developed a python based GUI to make the data retrieval fast and reliable. The general networked system is described in Figure 1.

RESULTS

The LabVIEW based approach should be as stable as the classic IOC one, in order to validate it. Up to the moment, several test are been carried out in the laboratory and data measurements have been made after a few days running. Since the main concerning parameters in this experiment are those related to reliability and repeatability, the obtained data and conclusions must be regarded as preliminary.

The overall EPICS system handles up to 1960 PVs, 980 from each structure, similar to the ones presented in [3]. Most of the variables are periodically processed at 1 or 5 second rate. A group of 180 PVs have a passive scan, driven by events. The retrieved data from EPICS is periodically written into the database every 10 seconds.

The extraction of the data from the database is made trough an Python based GUI as is shown in Figure 2, which



Figure 2: GUI for data retrieval.



Figure 3: Data losses in both systems.

allows to find and select the desired PVs and choose the output format (which can be a graph plot or a .csv file) in any time interval.

The plots shown in Figure 3 are related to two particular PVs. Those are "ai1" from the CODAC based system and its analog variable called "LVai1" from LabVIEW based approach. The graphs correspond to consecutive timestamps lapses, which should be 5s long in both systems measured in a 24 hour period. Preliminary data analysis show similar data losses in both systems, but they seem to be consequence of the testbench implementation (archiving and network). However, those are not EPICS-related errors, which is working flawlessly in both approaches.So, the experiment environment must be tuned up.

FUTURE WORK

The results obtained from the preliminary test encourages us to follow working in the proposed direction. The overall testbench environment must be tuned up to obtain definitive results for the validation process of the proposed architecture.

The first step to follow is to set up an identical architecture for both approaches. That means to use the same PXI models as well as connecting the input signal set of both systems to the same source in order to have exactly the same reference signals.

Secondly, the test stand must be continuously operating the two configurations in parallel for a long time period (months) in order to get reliable results concerning repeatability issues.

Another interesting idea that is going to be studied is the inclusion of synchronization. At present, this experiment has three different unsynchronized clocks, and therefore three different timestamps: one from each system and a third one from the archiver. So, the use of a NTP server is planed in a near future.

Finally, it is planed to carry out a new testbench introducing an EPICS-based wireless link system maintaining the present two architectures. The main goal is to study the effect of the wireless link, in order to facilitate the replacement of as many cables as possible to benefit from the advantages of the wireless systems [4].

ACKNOWLEDGMENT

This work has been funded by ESS-Bilbao Consortium and GAUDEE research group of the University of the Basque Country UPV/EHU.

- F. Di Maio, L. Abadie, C. Kim, K. Mahajan, P. Makijarvi, D. Stepanov, N. Utzel, A, Wallander, "The CODAC Software Distribution for the ITER Plant Systems", Proceedings of ICALEPCS2011, Grenoble, France, 10-14 October 2011.
- [2] M. del Campo, J. Jugo, M. Giacchini, L. Giovannini, "EPICS HyperArciver: Initial Tests at ESS Bilbao", Proceedings of IPAC2011, San Sebastian, Spain, 2011.
- [3] G. Shen, M. Kraimer, "Server Development for NSLS-II Physics Applications and Performance Analysis", Proceedings of PAC2011, New York, NY, USA, 2011.
- [4] I. Badillo, M. Eguiraun, J. Jugo, "NFC like Wireless Technology for Monitoring Purposes in Scientific/Industrial Facilities", Proceedings of ICALEPCS2011, Grenoble, France, 10-14 October 2011.

REACHABILITY IN A FINITE DISTRIBUTED SYSTEM PROTOCOL MODEL BY BACKWARD TRAVERSAL

Tapas Samanta, VECC, Kolkata, India Dipankar Sarkar, IIT, Kharagpur, India Samarpita Mukherjee, Jadavpur University, India

Abstract

Distributed system protocol verification has the intrinsic problem in mechanizing the reasoning pattern and the resultant state space exploration. The former arises in case of theorem proving approach due to the ingenuity involved in constructing a proof and the latter is encountered in model checking approach while carrying out composition of a large number of processes that constitute a typical distributed system. A combined approach of the above two methods has been devised that eventually considers the reachability in finite distributed system protocol model. It computes the reachability in backward traversal on the fly. In this paper a C++ implementation of the on-the-fly backward traversal algorithm is reported.

INTRODUCTION

A wide class of distributed system protocols comprises a large no. of identical processes, each with an identical behaviour having finite state-space. Behaviour of each of the processes can be captured by a state transition diagram defined in terms of a FSM structure. Since a distributed system model is having a state-space of the order \mathbf{m}^{n} where \mathbf{m} is the state-space of an individual constituent process and **n** is the number of constituent processes which is typically few hundreds. Due to this state-space explosion [1, 2] while composing the FSMs for large number of processes, the reachability analysis suffers from large computational complexity. The framework [3] explores the merits of tableau proof framework and devises a new backward traversal algorithm to improvise a completely mechanized verification technique containing the state-space explosion problem for distributed system protocols having identical participants. In [3] the observation is made that if a flat composition can be avoided and selective composition, as and when necessary, can be computed, then the state-space explosion can be contained to a great extent. In this paper, we present a C++ implementation of the backward traversal process which basically tests on-the fly the reachability of an atomic predicate in a given distributed system protocol model.

The paper is organized in the following way: Section 2 briefly explains the C++ implementation details of the Modified Backward Traversal Algorithm followed by an example of the implementation of the Backward Traversal Algorithm in the Section 3.

A C++ IMPLEMENTATION OF THE BACKWARD TRAVERSAL FOR LEADER ELECTION PROTOCOL

Data Structure Used in the Modified Algorithm

- 1. Class Table In this program, for the FSM (Finite State Machine) structure of the processes, we take the input from fsm.data file and have stored it in thee class Table. The class Table contains static vector containers (dynamic lists) those store the state, transition, guard and action conditions of the transitions as state, transition, guard and action objects respectively
- 2. Class Network To store the Network structure of the Distributed system, we have used class Network to store the input from the NetworkFile.data
- 3. Class triplet To store the Triplet node that contains a node's information in the form, <EndState, Process, TransitionIndex> we have declared the class triplet. In the class Table, there is a static vector <triplet> Triplet list that stores all the nodes of the computational paths.
- 4. Class tranCont The class tranCont stores index of two transitions (in vector<transition> Table::transitions) those are contradictory to each other. The static vector<tranCont> Table::TranContradiction stores all the contradictory transition pairs of the system.
- 5. Class stateCont The class stateCont stores index of two states (in vector<state> Table::states) those are contradictory to each other. The static vector<stateCont> Table::StateContradiction stores all the contradictory state pairs for any process of the system.
- 6. Class stReachability The class stReachability stores the lists of states of each process that has been reached so far. This helps to check if for any computational path, a process's current state contradicts with the states that had been so far reached by that process. This keeps track of the process's states so that if there is any state contradiction, then the path can be flagged as invalid.

Functional Construct of the Modified Backward Traversal Algorithm

Let ξ be the set of states where atomic predicate $\neg p_i$ holds. For each $s_j \in \xi$, construct a backward forest by invoking the functions:

1. i) FSM * fsm= new FSM("/home/.... /fsm.data"); // Creates objects of the FSM class taking input from fsm.data file ii)Network* net=new Network("/home/..../ networkFile.data"); // Creates objects of the Network class taking input from networkFile.data file

iii) Forest::Forest (FSM * fsm); // Create object of the Forest class passing FSM object.

2. i) void Forest::createRootNodes(atomicPredicate AP, Network *net); // Checks at which states atomicPredicate AP holds by calling the function vector<int> FSM::statesAPHolds(atomicPredicate AP); and create Triplet Root nodes (in the form <EndState, Process, TransitionIndex>) for each of these states.

ii) vector<int> Class triplet::childIndex; // Stores the index of child nodes in Table::Triple with parentIndx field that stores index of the parents of these child nodes.

3. i) void Forest::insert_Contradiction(Network * net); // Checks for contradictory states pairs and transition pairs by calling the function bool transition:: find_Contradiction(transition& x); and stores in Table::StateContradiction and TranContradiction; respectively.

ii) bool transition::find_Contradiction(transition& x); // Calls the function action::compare_Contradiction(guard grdCond); to check if Action of transition Ti contradicts with Guard of transition Tj.

- 4. void Forest::initPath(Network * net); // Creating children of the Root Triplets nodes based on message passing.
- 5. void Forest::createPath(Network * net) -

i) This function is recursively creating the children of the leaf triplet nodes (with childIndx.size()==0). This function creates child tripletNode similarly as the previous function void Forest::initPath(Network * net) but in a recursive way until all the tree branches close with initial states S1. As it is an Inverted path, the triplet <EndState, processId, TransitionId> contains the end state of the transition, so to find the initial state, we are looking for transition T1 (tranId==0), which will lead to the initial state S1.

ii) While creating each new child node (Figure 4), this function checks if the transition pair of parent and child triplet node, is a contradictory transition pairs.

iii) After all the inverted paths (from final state to initial state) are created, we check the list Table::Triplet to find a triplet node representing initial state (i.e. transition id==0 and end state==1) and store the index of these triplet nodes with initial state in a list vector<int> PathRoot[2].

iv) bool Forest::checkSt(int es, int pno) – While creating this inverted paths, in every node, we check the States reached by that process in the triplet node, by calling the function bool Forest::checkSt(int es, int pno). If there is any contradictory state reached, we discard that path by setting a flag, and if no contradictory state has been reached, then we store the new state of that process in the new child triplet node in the list Table::StateReached

6. This algorithm ends when all the branches of the triplet trees are closed.

Implementation Difficulties

During the implementation of the Backward Traversal Algorithm [3], following difficulties were faced.

- 1. This algorithm [3] is modelling a distributed scenario. Each tree node has a list of pointers to his child nodes and a pointer to its parent node. If we implement the algorithm exactly the way it is, then we have to move back and forth between parent and child nodes again and again to update the composite state of the current node, depending upon the value of the composite state of either the child node or the parent node of the current tree node and the vice versa. These increase the complexity of the program with the increase in the number of processes in the system.
- 2. During the Step 4. Refine Parents, and Step 6. Refine Ancestor [3], there can be more than one possible state in which a process can be, and if that is the case, then a node can have more that one incarnation. This causes the updating of not only those nodes, but also updating of the all the parent pointer information in all there child nodes and the child pointer information in the parent nodes of the newly incarnated nodes, and creation of new edges from that node to all its parent and child nodes. These situations cause the original tree to branch out and may split to create forests.
- 3. Using the child node pointers, it is easy to traverse downward in the computational tree branches, but it is complicated to traverse upward in the tree branches using parent pointers. So in case of a split in the computational branch, it becomes difficult to update all the composite states of the newly created branches or trees of the forest.
- 4. Here in this algorithm each node has a composite state, where each process in the system is in some state. But to find the reachability, we don't need the state information of all the processes at each node, for all possible tree branches as, in the future steps most of these branches would be purged.

CASE STUDY: AN UNIDIRECTIONAL RING WITH THREE PROCESSES

To find out the reachability of state where the atomic predicate L(2) holds, we start with the triplet state of the form <State_Id, Process_Id, Transition_Id> where L(2) holds. From Figure 1 we notice that S4 and S5 are the only such states for which L(2) holds, and for state S4 and S5 the respective enabled transitions are T4 and T5. So we start with two Root Triplet nodes N1 = < S4, 2, T4 > and N2 = < S5, 2, T5 > from which the backward traversal has to be taken up.

Step 1: Create Roots: Designate the Roots as $N1 = \langle S4, 2, T4 \rangle$ and $N2 = \langle S5, 2, T5 \rangle$.

Step 2: Initialize the Paths: From Figure 2 we can see that action of transition T4 of process 2 enables the guard condition of transition T2 of the neighbouring Process 1

with state S2. Similarly action of transition T5 of Process 2 enables the guard condition of transition T3 and T4 of the neighbouring Process 1 with state S3 and S4 respectively. This creates child Triplet node N3 = < S2, 1, T2 > of node N1 and child Triplet node N4 = < S3, 1, T3> and N5 = < S4, 1, T4 > of node N2.



Figure 1: Step 2 Initialize the Paths.



Figure 2: Step 3 Create Paths.

Step 3: i) Create Paths: Now similar to the previous step we keep creating children of the leaf nodes in the branches. So we create the child nodes of node N3, N4 and N5. While creating children of the leaf node, we find that from node N5 = < S4, 1, T4 > no child node can be created as the action of transition T4 of process 1 (in node N5) do not enables the guard condition of any transition without creating contradiction. So this branch with leaf node N5 is closed unsuccessfully. This creates child Triplet node N6 = < S2, 0, T2 > of node N3 and child Triplet node N7 = < S3, 0, T3> of node N4.

Step 3.i): Now we keep creating the child nodes of the leaf nodes in an iterative way, until the branches are successfully or unsuccessfully closed. The child node of N6 is N8 = < S2, 2, T1 > and N9 = < S2, 2, T2 > and child node of N7 is N10 = < S4, 2, T4 >. Here the branch with leaf node N8 is closed successfully as the Process 2 is in state S2 with transition T1 which will change the Process 2's state to initial state S1. So this branch is closed successfully (Figure 3).

Path 1: <\$2, 2, T1> -> <\$2, 0, T2> -> <\$2, 1, T2> -> <\$4, 2, T4>

Step 3.ii): Similarly we find that the branches with leaf node N15 = < S2, 2, T1 > and N16 = < S2, 2, T1 > are also closed successfully (Figure 5).



Figure 3: Path 1 successfully closed.



Figure 4: Creating Child Nodes N11 and N12.

Step 4: In this Backward Traversal is three computational paths are identified, Path 1, 2 and 3. There are also other paths but the Paths 1, 2 or 3 will be subsumed in those paths requirements.



Figure 5: Path 2 & 3 are successfully closed.

CONCLUSION AND FUTURE SCOPE

In this paper we have implemented the modified algorithm to avoid the complexities of creating the composite states and composite transitions for each node of the tree branch. This program can verify the LEP protocol for any finite number of processes.

But it has some limitations. This method works only for LCP problem with no internal variables. In future, to make it work for any Distributed System Protocols we need to make this program more general. To take care of the internal variables, we can create a global static array (of size equal to the number of processes) that can store the internal variable values for each process, so that it will work for algorithms like Flood-Max algorithm, or HS Algorithm containing some internal variables for each process. In future work, we also need to modify the function comparing the Guard and Action conditions of the transitions to work it for different protocols.

- Gerard Tel, "Introduction to Distributed Algorithms", ISBN-10: 0521794838, ISBN 13: 978-0521794831, October 2000.
- [2] N. A. Lynch, "Distributed Algorithms", 1948, ISBN-13:978-1-55860-348-6 ISBN-10:1-55860-348-4.
- [3] Tapas Samanta and D. Sarkar, "Distributed System Protocol Verification: A Tableau Based Model Checking Approach", Computer and Communication Technology (ICCCT), 2011, 2nd International Conference on: 246 – 252, 15-17 Sept. 2011, ISBN: 978-1-4577-1385-9.

DESIGN CONSIDERATIONS FOR DEVELOPMENT OF DISTRIBUTED DATA ACQUISITION AND CONTROL SYSTEM (DDACS) FOR RADIO-ACTIVE ION BEAM (RIB) FACILITY

K. Mourougayane[#], A. Balasubramaniam, G. Karna, P. S. Penilop, SAMEER, Chennai, India D.P. Dutta, T. K. Mandi, H. K. Pandey, VECC, Kolkata, India

Abstract

The Radioactive Ion Beam (RIB) facility is being constructed by the Department of Atomic Energy (DAE) of India at Variable Energy Cyclotron Centre (VECC), Kolkata. The RIB facility at VECC consists of various subsystems like Electron Cyclotron Resonance Ion Source (ECR-IS), Radio Frequency Quadrupole Linear Accelerator (RFQ LINAC), Rebunchers, Inter-digital Hmode Linear Accelerators (IH LINACs) etc., which are required to produce and accelerate the energetic beam of radioactive ions for different experiments. In technical collaboration with VECC, Kolkata, a unique system called "Distributed Data Acquisition and Control System (D-DACS)" was developed. This paper is indented to bring out the design considerations in developing distributed control systems and qualification requirements with reference to functional, environmental and Electromagnetic Compatibility (EMC) standards.

RIB FACILITY – AN OVERVIEW

The RIB facility (Fig. 1) at VECC is an ISOL type RIB facility. Radioactive isotopes are first produced in a target from nuclear reaction with the primary beam. Radioactive atoms diffusing from the target are ionized initially in an integrated 1^+ ion-source and then in ECR ion-source. After mass separation, the low energy RIB is accelerated from 1.75 KeV/u to about 100 KeV/u in a RFQ LINAC. Three IH LINAC modules raise the energy up to around 415 keV/u. Subsequent LINAC modules are to be used to achieve the final beam energy of 1 to 5 MeV/u [1].

The RIB beam line is equipped with state of the art systems to produce and accelerate Radio-active Ion Beam. The major systems are broadly listed below:

- Linear Accelerators (LINACs)
- RF Quadrupole (RFQ)
- Rebunchers
- Faraday Cup, Vacuum Pump, Gate Valve systems etc., and theses beam line systems are powered with:
- Klystron High Power Amplifier (KHPA)
- High power RF Transmitters
- High voltage sources
- High current power sources

Majority of the systems used in the RIB are potential sources of Electromagnetic Interference (EMI). to meet the functional, environmental and Electromagnetic Compatibility (EMC) standards in development of systems for RIB facility need expertise on multiple domains, covering data acquisition, instrumentation and control system.

km@cem.sameer.gov.in



Figure 1: Layout of RIB facility with subsystems.

NEED FOR DDACS SYSTEM

The RIB facility integrates many Electronics Subsystems procured from different international sources. Each system has its own interface for remote controlling the system parameters. Some of the systems like RF Transmitters do not have provision for remote control. Hence, the DDACS system is aimed to develop all necessary systems, sub-systems and interface circuits to monitor and control the RIB systems. The software developed for DDACS system helps an easy way of controlling the parameters at sub-system level and system level with necessary Graphical User Interface (GUI). An Embedded Controller planned at the Control Room (CR) shall control all the RIB systems and associated equipments [2].

DDACS SYSTEM DESIGN APPROCH

The RIB beam line systems and associated control parameters are shown in Fig.2. The design approaches followed are detailed below.

- The DDACS System is required to control all the equipments of RIB beam line systems and also to monitor the status of the systems continuously.
- The design is planned to support Distributed Data Acquisition considering the physical locations and number of associated RIB systems
- Entire RIB beam line systems are divided in to eight sections depending on the process flow and the equipments connected to it.
- Each section will have one Data Acquisition Front end module with all necessary interface circuits to

authors



Figure 2: RIB Sub-system and control parameters.

control and monitor the RIB-Sub-system in standalone mode.

- There are eight such Data Acquisition and Interface modules called "Equipment Interface Module (EIM)" which covers all the RIB Systems
- Each EIM modules independently collects the data, monitors and control their respective RIB systems
- There are two RIB Controllers, one positioned near RIB beam line and other in the Control Room. Control and monitoring the status of the RIB systems are possible from both the RIB Controllers depending on the operational requirements.
- The EIM modules and the RIB Controllers are connected through Fibre Optic Interfaces for EMI free operation
- The DDACS system is planned with three layer architecture, i) Equipment Interface Layer ii) Supervisory Control Layer iii) Operator Interface layer [3].

DEVELOPMENT OF DDACS SYSTEM

RIB beam line systems are broadly divided and classified in to eight sectors covering the following sub-systems.

- a) Electron Cyclotron Resonance Ion source (ECRIS)
- b) ECRIS -to- RFQ Beam Line
- c) Radio Frequency Quadrupole (RFQ)
- d) RFQ-to-Rebuncher Beam Line
- e) Rebuncher
- f) Rebuncher-to-LINAC-1 Beam Line
- g) LINAC-1, LINAC-2 and LINAC-3

Each section covers group of sub-systems of RIB beam line systems and equipments connected to it. A dedicated Data Acquisition module, the "EIM module" is connected with necessary hardware interface.

Equipment Interface Module (EIM)

The EIM module commands and collects data from each RIB sub-system. There are eight EIM modules connected in a distributed manner to acquire data from all the RIB systems. The data collected from each EIM module are transferred to the RIB controller through the Fiber optic interface. The EIM Modules with an integrated graphical touch screen display are shown in Fig 3. A sample view of the GUI developed for EIM module is shown in Fig 4.



Figure 3: EIM modules.

RIB Controller

The RIB controller is a rugged Control Console aimed to collect the data from all the EIM modules. The

application software is developed with GUI for easy monitoring and control of selected parameters or subsystems. The RIB Controller is shown in Fig. 5 and the sample of GUI panel is shown in Fig. 6.



Figure 4: Graphical User Interface in EIM Display.



Figure 7: FC Controller.



Figure 8: Slit controller.



Figure 5: RIB Controllers.



Figure 6: Sample GUI panel display.

Other DDACS Controllers

The Faraday Cup (FC) Controller, which switches and measures the current through the Faraday Cup, is shown in Fig.7. The Slit Controller which controls the slit position in the RIB chain is shown in Fig. 8. The RF Transmitter Remote Interface Module (RIM) and Gate Valve (GV) Controller are shown in Fig. 9 and Fig. 10.



Figure 9: RF Tx- RIM.



Figure 10: GV Controller.

RIB System Simulator

The RIB System simulator is developed as a Test Jig, which can provide test data in the prescribed format for evaluation of the D-DACS System. The RIB Simulator is shown in Fig. 11



Figure 11: RIB System Simulator.

SYSTEM QUALIFICATION ASPECTS

The DDACS system is designed and developed considering the end application and operating environment. The following three aspects were addressed during the system design and implementation phase.

- a. Functional specifications and operational requirements
- b. Environmental specifications
- c. Electromagnetic Compatibility(EMC) requirements

Functional Specifications

The Functional and operational requirements are based on the system specifications provided by the end user. The DDACS system design is based on the parameters to be controlled and monitored in each RIB sub-systems. A Critical Design Report (CDR) was prepared with the details of System Specifications, Configuration, Data Communication Protocols and Software development plan. An Acceptance Test Procedure (ATP) document was submitted with the details of functional tests.

Environmental Specifications

The DDACS system is meant for operation in the RIB facility and hence the system should be qualified for operation at elevated temperature. Hence, the system was designed and tested for limited environmental specifications, like continuous operation at high

temperature at 60 deg C, low temperature at 0 deg C, Dry heat test at 45 deg C, 90% Relative Humidity (RH).

EMC Specifications

As the System should operate in the Interference prone environment due to simultaneous operation of high voltage/high current sources, magnetic sources, high power RF Transmitters, it is mandatory that the DDACS system shall be qualified for EMC standards.

As the DDACS System can be a potential victim of Electromagnetic Interference generated from the RIB systems, the following EMC tests were carried out as per IEC Standards and the system was qualified.

- i. Radiated Susceptibility (RS) test IEC 61000-4-3
- ii. Power Frequency Magnetic Field test IEC 61000-4-8
- iii. Conducted RF Immunity test -IEC 61000-4-6

SYSTEM EVALUATION AND INTEGRATED TESTING

The DDACS system was qualified for functional specifications, environmental specifications and EMC standards before installation of the system in the RIB facility. Next, the system was installed and interfaced to all the RIB Systems as shown in Fig.12. The integrated testing was carried out as detailed below:

- a. System control operation from EIM modules
- b. System control from the RIB Controller positioned in RIB Room
- c. System control from the RIB Controller positioned in the Control Room.

CONCLUSION

The DDACS is an indigenous system developed by Society for Applied Microwave Electronics Engineering & Research (SAMEER), Centre for Electromagnetics, Chennai in technical collaboration with the VECC, Kolkata. The system was developed, qualified for functional, environmental and EMC specifications. The system was installed and interfaced to the RIB systems and integrated tests were carried out. The system is being used in the RIB facility at VECC, Kolkata.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to Director, VECC and Director, Programme Director, SAMEER for the technical and financial support extended for this project. The authors also thankfully acknowledge the overall guidance of Dr. Alok Chakraborti, Head, Radioactive Ion Beam Facilities Group, VECC. All the officers and staff members of RIB Group, VECC and SAMEER who are involved in this project deserve special thanks for their sincere efforts towards achieving the goal.

Copyright ${
m (C)}$ 2012 by the respective authors



Figure 12: D-DACS system Integration.

- [1] VECC, Kolkata and SAMEER, Chennai, Critical Design Review (CDR) Document of Distributed Data Acquisition and Control System (DDACS) of RIB Facility at VECC, Version: 1.1, November 2009.
- [2] Technical Manual of Distributed Data Acquisition and Control System (DDACS) of RIB Facility at VECC, Version: 001, January 2011.
- [3] K. Datta et al, Development of Distributed Data Acquisition and Control System for Radioactive Ion Beam facility at Variable Energy Cyclotron Centre, Kolkata, MOPMU014, Proc. ICALEPCS2011, Grenoble, France (2011).

FPGA BASED AMPLITUDE CONTROL SYSTEM FOR ACCELERATING **CAVITIES**

M. Dev, A. Singh, S. Ghosh, A. Mandal, S. Seth, S. Som, VECC, Kolkata, India

Abstract

The FPGA (Field Programmable Gate Array) based digital controller has been implemented for low level RF voltage control of a 650 MHz cavity. Implementing amplitude control in a compact single board solution, FPGA is chosen. The Superconducting cavity is designed to be operated at 650 MHz and 30kW, CW mode. The voltage from pick-up coil has been fed to the controller after down conversion; the signal is digitized using high speed ADCs. The controller has been simulated with different set points and gain parameters. The FPGA signal processing has been simulated according to the required strategy of the reference controller. Some simulation results have been presented for different cavity operational conditions.

BRIEF THEORY OF TRANSFER FUNCTION

Suppose the input to the cavity be an amplitude modulated (AM) signal with carrier frequency ω_c , exactly tuned to the cavity. The AM input signal is expressed mathematically as:

$$V_i = A_i [1 + a_i(t)] \cos \omega_c t$$

The output of cavity will also be an AM signal with carrier same as above and is expressed mathematically as:

$$\mathbf{V}_{0} = A_{o}[1 + a_{o}(t)]\cos\omega_{c}t$$

On demodulating the output AM signal we will get the output baseband signal $a_o(t)$. It can be verified that the cavity action on baseband input signal $a_i(t)$ is a low pass filter whose cut-off depends upon the cavity resonant frequency and quality factor. Thus the baseband transfer function of the cavity model can be expressed as:

$$G(s) = \frac{a_o(s)}{a_i(s)} = \frac{1}{1 + s / \sigma}$$
, where $\sigma = \frac{\omega r}{2Q_L}$

SYSTEM LEVEL IMPLEMENTATION ON **FPGA**

The implementation of the cavity control loop requires a modulator and demodulator along with a controller with an external set point and gain. Figure 1 shows the system level block diagram. There are three parts in the block diagram comprising of firmware, RF cavity and the portion that implemented outside the FPGA. The components outside the FPGA firmware are the ADC-DAC add-on modules, PC running the software and the RF cavity system.

The control loop parameters like set-point, gain and loop control action are controlled by the user through an RS232 serial port. Control loop implemented using FPGA firmware and a proportional controller at the current stage as in Figure 2, the set-point is compared with the demodulated cavity output in FPGA, and the error data after multiplying by the gain parameter 'G', goes outside the FPGA for modulation since the DAC input is ACcoupled. The DAC output after modulation with a 5 MHz carrier goes into the cavity.



Figure 1: System level block diagram.

FPGA Hardware and Firmware Details

Hardware details: The hardware components of the implementation mainly consist of a FPGA (Virtex4) MB development board and an ADC-DAC add-on module. ADC accepts analog input of 1.8 Vpp, and outputs digital data in 2's complement format. It has a conversion delay of 17.5xT_{CLK}.ADC output clock acts as the FPGA clock @ 100 MHz; DAC accepts a 16-bit digital input in 2's complement format and outputs pk-pk value of 500mV.



Figure 2: Block diagram of the control loop implementation using FPGA.

2012] Firmware details: The FPGA firmware is designed to capture the high speed ADC data followed by its 0 demodulation and then applying the digital control action.

ght

Demodulating an AM signal requires a rectifier followed by the envelope detector with a peak detector module. The peak detector module acts as an envelope detector. The rectifier is introduced just to clip out the negative value data so as to reduce the number of data samples in a window. The peak detector is based on window based peak detection. Controller, it compares the demodulated output with the set-point value and the gain. Modulation of the controller output is done outside the FPGA, and the modulated signal is given back into FPGA for conversion to analog value using DAC inside the add-on module.

SIMULATION RESULTS

The above control loop has been simulated in MATLAB Simulink simulation tool. Various parameters are downscaled by factor of 10^6 for convenience. The carrier level control loop in Figure 2 can be converted to its equivalent baseband model by replacing the components.



Figure 3: Baseband equivalent model of control loop.

It can be shown in Figure 3; the output stability is limited by the gain G and delay T_0 . The simulation output of actual carrier level and baseband model is shown in Figure 4(a) to 4(e) for various values of gain and delays. The gain and set-point of carrier level model is limited by product G x S.P. <32767 (for 16-bit DAC). All the simulation outputs in the following figures are having a set-point of decimal 3000 for the carrier level model and a set-point of unity for the baseband model.



03 08 07 06 05 04 03

Figure 4(b): Baseband model Simulink output, G=5, T_0 =0.02 seconds.

As seen from the Figure4 (a), the output is at the demodulator in response to the step decimal input of 3000. The output gets stabilized to 1800 value. The output shows little ripple due to the carrier noise after demodulation. The ripple is of order of 5.5 mV for ADC input range of 1.8Vpp, which amounts to 0.28% only. Figure 4(b) shows the similar response to the step input to the equivalent baseband model. The output is a factor 5/6 of input i.e. 0.833.



Figure 4(c): Carrier model Simulink output, G=10.

Similarly, the output response of Figure 4 (c) shows that if gain is increased by 10, then output gets stabilized to 2300. The output shows transient oscillatory response due to the decrease in stability caused by the increase of gain. Similar oscillatory response we get in the baseband model with gain of 10, and delay 0.05 seconds.



Figure 4(d): Baseband model Simulink output, G=10, T_0 =0.05 seconds.



Figure 4(e): Baseband model Simulink output, G=10, $T_0=0.2$ seconds.

Figure 4(e) shows the output of baseband model with an increased delay gets unstable, while gain remains unchanged.

TESTING DEMODULATOR

The control loop shown in figure 2 has been simulated. The demodulator along with the controller is tested in the lab, and found working. As shown in the Figure 5, the AM signal is applied to the ADC input, and after demodulation in FPGA firmware, and the controller section, the data is fed directly to DAC to test firmware implementation. The modulation is bypassed in this testing since it has to be implemented outside the FPGA.



Figure 5: A 5 MHz carrier signal modulated by a 50 kHz baseband sine wave demodulated by a demodulator module implemented in FPGA.

The simulation work was complete and the model is codified in vhdl. The design is not yet put on the actual cavity. The actual set-up and the performance study are undergoing. The concept and the details of the work have been taken into account from [1, 2, 3, 4].

- [1] Tomasz Czarski, Krzysztof Pozniak, Ryszard Romaniuk, Instt. of Electronic Systems, Warsaw University of Technology, Poland, Stefan Simrock, DESY, TESLA, Hamburg, Germany, "Cavity control system model simulations for TESLA linear accelerator".
- [2] A. Gallo, INFN LNF, "Basics of RF electronics", CERN Yellow Report CERN-2011-007, pp. 223-275.
- [3] Erk Jensen, "RF Cavity Design", CERN BE/RF, CERN Accelerator School, Accelerator Physics, Chios 2011.
- [4] Wojcieh M. Zabolotny, et al, "FPGA based Cavity Simulation for Tesla Test Facility", Tesla Report 2003-22.

THE NEW WHITE RABBIT BASED TIMING SYSTEM FOR THE FAIR FACILITY

Dietrich Beck, Ralph Bär, Mathias Kreider, Cesar Prados, Stefan Rauch, Wesley Terpstra, Marcus Zweig, GSI Helmholtzzentrum für Schwerionenforschung, D-64291 Darmstadt, Germany

Abstract

A new timestamp and event distribution system for the upcoming FAIR facility is being developed at GSI. This timing system is based on White Rabbit (WR), which is a fully deterministic Ethernet-based network for clock and time distribution. WR is developed by CERN, GSI and other institutes as well as partners from industry based on Synchronous Ethernet and PTP [1].

The main tasks of the General Machine Timing (GMT) system are time synchronization of more than 2000 nodes with nanosecond accuracy, distribution of timing messages and subsequent generation of real-time actions (interrupts, digital signals...) by the nodes of the timing system. This allows precise real-time control of the accelerator equipment according to the beam production schedule. Furthermore the timing system must support other accelerator systems like post-mortem [2] and interlock [3]. It also provides interfaces between the accelerator control system and experiments at FAIR.

INTRODUCTION

The GMT triggers and synchronizes accelerator equipment, timed according to the accelerator cycles [4]. Cycle lengths range from 20 ms (present UNILAC), several seconds (synchrotrons SIS18 and SIS100/300) to several hours (storage rings). An important concept of the accelerator control system is the one of the so called beam production chain, which describes the production of a beam from an ion source through the accelerators to a target. Properties of such a beam production chain include the ion type (from protons to uranium), energy, intensity, focus and emittance at the final destination and other parameters. This is mapped to the GMT, which has an integral view on the tightly synchronized accelerators and beam transfer sections. The GMT must take into account the execution of several beam production chains in the accelerator complex at the same time. For each part of the machine, switching between different beam production chains will be possible between cycles, which implies a high degree of true parallel operation.

For all components, set values or ramp data for the different beam production chains are preloaded in the Front-End Controllers (FECs). This is done by the control system via its general purpose network. The task of the GMT is to trigger and synchronize the FECs in real-time via *timing events*, which carry additional information like machine ID or event numbers. After parameterization of the facility by the top and middle layer of the control system, it is finally the GMT which autonomously operates the accelerator complex in real-time. Besides the GMT, the so-called Bunch phase Timing System (BuTiS) serves as a campus-wide distribution system for clock signals. BuTiS is focussed at synchronizing the radio-frequency systems of the accelerator and provides distributed and synchronized clock signals with a precision in the low picoseconds and jitter in the low femtoseconds range [5].

REQUIREMENTS

The requirements of the GMT have been described in the detailed specifications [6]. About 2000 FECs and other equipment are connected to the GMT, a distance of up to 2 km between the nodes has to be covered. In most of the cases a precision of 1 μ s is sufficient. However, some equipment like kicker magnets for transferring bunches between machines require nanosecond precision.

Besides supporting control system features like equipment triggering and synchronization, parallel execution, varying machine cycle times and scalability, other key features are the following: Robustness - At most one timing event per year may be lost. Determinism - Time critical information from must be distributed to the nodes with an upper bound latency. Redundancy - Core components of the GMT must be implemented with redundant equipment. Availability - The GMT must be capable of distributing events for testing and commissioning equipment, even when the accelerator does not produce beam. Plan B Execution - The GMT must react on external signals like interlock signals or malfunctioning equipment by executing predefined alternatives in the schedule. Other important features include integration of the existing machines and control system as well as interfaces to many other subsystems like the interlock system and BuTiS.

DESIGN

The main idea is to build the GMT based on the notion of absolute time. Nodes in a network share the same clock and time. By this, distribution of information and timely triggering are decoupled. This is achieved by preprogramming timing receivers for autonomous execution of actions at a given time and date.

White Rabbit as Field Bus

The new timing system for FAIR is based on a White Rabbit network. The focus of WR is on clock and timestamp distribution, thus synchronizing nodes of a network [1, 7]. WR employs Gigabit-Ethernet, IEEE1588-2008 (PTP), precise knowledge of the link delay and Synchronous Ethernet. The idea is to adjust the clock phase
(125 MHz carrier frequency) and offset (timestamps in Coordinated Universal Time - UTC or International Atomic Time - TAI) of all network nodes to that of a common grandmaster clock. Today, synchronization in the one nanosecond range with a jitter in the low picoseconds range is achieved using commercial products that have been developed within the Open Hardware Project [8].

Timing System

The main components of the GMT are a *timing master*, a *timing network*, and *timing receiver nodes*.

The timing master acts as a grandmaster clock providing the time for all the nodes in the GMT. It furthermore fulfills the task of the real-time control of the accelerator complex by generating and distributing *timing messages* according the schedule provided by a settings management system. Timing messages are broadcasted to all nodes in the network. The timing master has interfaces to other system of the accelerator control system like the settings management system [9], the interlock system [3] or the the post mortem system [2]. GMT benefits from BuTiS by using it as reference clock input to the grandmaster clock of the timing master. As BuTiS uses a GPS Disciplined Oscillator (GPSDO) for long term stability, the GMT itself uses and distributes timestamps based on GPS time.

A dedicated timing network distinct from the general controls network is required. First, WR-PTP uses the physical carrier signal of GigE for propagation of clock and phase from the timing master to the nodes, requiring dedicated switches. Second, the timing network must have real-time capabilities transmitting timing messages with a guaranteed upper bound latency. Third, the timing network support robustness against loss of packets and bit errors.

FECs are connected to two distinct networks. First, the general purpose network of the accelerator control system is used to transmit set values in order to preconfigure a FEC in advance. Second, the timing receiver, which is typically an interface card connected via the host-bus bridge of a FEC, has a link to the timing network. The main form factor for FECs at FAIR is the Scalable Control Unit (SCU) [10]. As timing receivers receive all timing messages broadcasted by the timing master, they must filter the identifiers, that are contained in the messages stream. Filtering and processing is done in real time. Once a relevant message has been identified, an action is scheduled for execution at a given time and date. An action could be issuing an interrupt request via the host system bus, which triggers a pre-configured real-time action in the front-end software of the FEC. Of course, timing receivers can generate digital signals for direct hardware triggering in cases, where a precision in the order of nanoseconds is required.

IMPLEMENTATION PLAN

From the point of view of the hardware, White Rabbit is quite simple and based on fairly cheap electronic components which are commonly available. WR interface cards can be obtained from different companies. The development of switches for WR networks is completed and such switches will be available as commercial products in 2013. Thus, the challenge for building the FAIR timing system is not so much the development of new technologies but to combine existing technologies using White Rabbit technology as a field-bus. Especially the scaling of the timing system to more than 2000 nodes as required for the FAIR facility must be addressed with care [6].

The solution is to develop the GMT in an iterative and incremental way. Each iteration cycle results in a running system. The first iteration cycle must demonstrate the principal feasibility of critical components, while successive iterations implement additional features until the desired total functionality is reached.

From Existing Machines to Fair

Proton Linac Source In summer 2013 GSI will deliver a complete control system for testing the source of the proton linac, that will later serve as an additional injector into the existing synchrotron SIS18 at GSI/FAIR. At this stage, the timing system will only be a simple pulse generator for a handful of FECs. However, this will be the first productive timing system that uses prototypes of the FAIR timing system, such as a timing master, a timing network and timing receivers.

CRYRING The next step will be a timing system for the CRYRING, a small synchrotron and storage ring located at the Manne Siegbahn Laboratory in Stockholm [11]. This ring will be moved to GSI, where it will be setup next to the Experimental Storage Ring (ESR). One of the motivations behind the CRYRING project is its explicit usage for FAIR development tests. Although the operation of the ring will commence in stand-alone mode, it covers nearly all relevant aspects of an accelerator facility. Thus, it presents an ideal test ground for new sub-systems of the accelerator control system such as the timing system. Here, new development and features for FAIR can be tested without the overhead required for routine operation. It is estimated that between 20 and 50 FECs have to be connected to a first timing system. Although limited in features, deployed components for the CRYRING timing system will be close to the final ones for FAIR.

New Timing System for SIS18 and ESR The timing systems for the proton linac source or CRYRING have been either very small in scale our operated under experimental conditions. Replacing the existing timing system at the synchrotron SIS18 and the storage ring ESR is a important milestone, since the ongoing experimental program requires the timing system to work reliably in routine operation 24/7. The old timing system, which is based on an extension of the MIL-STD-1553 bus, needs to be replaced by the new GMT: A common solution for the timing system for all ring machines should be used to guarantee and efficient transfer of ion bunches from one ring machine to the next. **Integration of UNILAC** Next to the new proton linac, the existing heavy ion linear accelerator UNILAC is the second injector into the FAIR accelerator complex. Here, the existing MIL-based timing system will most likely not be replaced for the reasons of cost and effort. Moreover, the UNILAC is special with respect to timing. First, it is operated at 50 Hz and phase locked to the mains voltage delivered to GSI. Second, the ion sources feeding the UNILAC require fixed repetition rates for reasons such as thermal stability. However, injecting the UNILAC beam into the SIS18 requires linking the UNILAC timing to the schedule of the FAIR accelerator complex. This must be achieved and tested prior to commissioning the FAIR facility.

Final Timing System All of the previous instances of the new GMT had different aspects: Proof of principle at the proton linac source, prototyping and development of final solutions at the CRYRING and reliability of routine operation at the SIS18 and the ESR. One important step towards the final timing system is the combination of all these aspects. This is eased, since the timing master and the first two layers of switches will be physically located in an existing building close to the SIS18. This allows to set up first components of the timing master for CRYRING, SIS18 and ESR already. By this, the timing master, its infrastructure and its interfaces will be continuously developed, improved and tested at its final location over a few years.

Another important step is to address the issue of scaling the new GMT to the size required for the final FAIR facility. As it is planned to purchase the equipment not just before FAIR machine commissioning but over a period of several years, scalability can be addressed and tested in several steps. One option would be to use test areas in buildings of the existing facility.

STATUS

During the past years, the main focus of the work done at GSI has been on the design, specification and prototyping of components for the GMT.

As the timing master for the FAIR facility will be located in an existing building at GSI, space was allocated and racks for electronics have been set up. First cables for fiber links have been laid to connect the timing master to three different locations on the GSI campus. GPS antennas and a GPSDO have been installed and are being commissioned. The GPSDO serves as a stabilized external reference clock with low phase noise and provides GPS timestamps. At the same location, a White Rabbit switch has been installed which synchronizes to the GPSDO. Today, this switch can today be used as grand-master clock for first tests to supply three other locations at GSI with White Rabbit links. Connected to these links are White Rabbit receiver nodes.

White Rabbit PTP just provides clock and time synchronisation. However, timing receivers for the GMT need to implement the functionality to schedule and execute actions for synchronizing the accelerator equipment. This is a major effort, which has already been addressed. As one of the results, the so-called *GSI Timing Starter Kit* has been developed which already today allows features like digital outputs at a pre-programmed time and date as well as latching of timestamps [12]. The SCU serves as the main platform for the development of timing receivers at GSI.

SUMMARY AND OUTLOOK

A new general machine timing system for FAIR is presently being developed in iteration cycles, where each iteration provides a functional timing system focusing on a certain aspect of the final timing system. The first iteration yielded a timing starter kit. The next iterations will implement timing system for the source of the FAIR proton linac, followed by the CRYRING at GSI. The replacement of the existing timing system at the SIS18 and the ESR combined with addressing scalability will pave the way to the new timing system for the FAIR facility.

ACKNOWLEDGMENT

The authors gratefully acknowledge the creativity and support of the *CERN White Rabbit Team*, the driving force behind the development of White Rabbit PTP. Furthermore the authors acknowledge the help of the department of *Experiment Electronics* at GSI, who provided us with hardware modules and consulting.

- J. Serrano, P. Alvarez, M. Cattin, E.G. Cota, P.M.J.H. Lewis, T. Włostowski et al., "The White Rabbit Project" in Proceedings of ICALEPCS TUC004, Kobe, Japan, 2009.
- [2] J. Fitzek et al., F-DS-C-13e, FAIR Detailed Specification "Post Mortem System".
- [3] J. Fitzek et al., F-DS-C-08e, FAIR Detailed Specification "Interlock System".
- [4] T. Fleck et al., FAIR Accelerator Control System Baseline Technical Report, General Machine Timing System, 2012.
- [5] P. Moritz et al., F-CS-RF-14e, FAIR Common Specification "Bunch Phase Timing System".
- [6] D. Beck et al., F-DS-C-05e, FAIR Detailed Specification "General Machine Timing System", 2012.
- [7] T. Włostowski, "Precise time and frequency transfer in a White Rabbit network" Warsaw University of Technology, May 2011.
- [8] The Open Hardware Repository, http://www.ohwr.org
- [9] J. Fitzek et al., F-DS-C-03e, FAIR Detailed Specification "Settings Management System".
- [10] M. Thieme et al., F-DS-C-02e, FAIR Detailed Specification "Equipment Interface and Control".
- [11] M. Lestinsky et al, "CRYRING@ESR:A study group report", http://www.gsi.de Darmstadt, July 26, 2012
- [12] W. Terpstra et al, "The GSI Timing Starter Kit", http:// www.ohwr.org/projects/gsi-timing-starter-kit/ wiki

STATUS REPORT, FUTURE PLANS AND MAINTENANCE ISSUES OF VME BASED CRYOGENIC CONTROL SYSTEM AT IUAC

Joby Antony, D.S. Mathuria, T.S.Datta

Inter University Accelerator Centre, Aruna Asaf Ali Marg, New Delhi, India

Abstract

The Cryogenic Data Acquisition and Control system (CRYO-DACS) at IUAC was commissioned successfully in the year 2002 and has been continuously in operation since then with uptime better than 95%. The aim of CRY-DACS is to control and acquire many critical analogue and digital cryogenic parameters of super conducting LINAC and related equipments like beam-line cryostats, helium compressors, liquefier, cryogenic distribution line etc. from a central control room, without fail. The system monitors analog parameters like cryogenic temperature, pressure, vacuum and controls cryogenic fluid levels inside the cryostats and performs closed loop controls of cryogenic valves. The complete system is implemented using two distributed VME crates, housing I/O modules, placed far apart and interconnected using Ethernet. The entire system is still operational with a very few failure issues in last ten years due to high dust environment. The software implementation and maintenance have also been easy and trouble-free as we used some commercial software tools from m/s VMIC called IOWORKS for development. In summary, this paper will elaborate the implementation, use and related failures faced for last 10 years and the subsequent corrective actions taken to keep the system running for such a long time round the clock along with some future plans.

INTRODUCTION

The Super conducting booster linac project [1] at Inter University Accelerator Centre (IUAC) uses niobium quarter wave resonators for the acceleration of heavy ions to high energies [2]. A cryogenic facility has been setup for maintaining the above system at helium temperatures during normal operations. This facility include five beam line cryostats namely Buncher, Linac1, Linac2, Linac3 & Re-buncher, Nitrogen & Helium plants and distribution lines etc. In order to monitor, control and analyze all the above system from a central control room, the Cryogenic data Control network has been setup as shown in Fig. 1.

DEVELOPMENT TOOLS

IOWorks is a suite of PC-based software development and run-time tools for end-users and systems integrators to create affordable, scalable, and high-performance automation applications. IOWorks takes advantage of the PC platform and its wide range of supported software applications to provide an open system for automation and control environments. IOWorks board drivers are M/s VMIC (now GE) Intelligent Platform's data acquisition and control (DAQC) software driver. It is a hardware and



Figure 1: A view of Cryogenic control room at IUAC.

operating system-independent driver designed to support the widest variety of I/O in the industry.

CRYO-DACS used VMIC's Intel-based VME CPUs (VMIVME-7698) using VMIC's VMISFT-package, which consists of a set of functions and utilities to easily develop, debug, and run standard VxWorks applications that access VMIC's broad line of I/O boards.

The VMISFT supported Visual Basic, C/C++, the IOWorks Soft Logic Link, and other languages that can call functions in a dynamic link library or connect to a DDE server. C function library provided a common interface to VMIC I/O products for reading, writing, and configuring.

VME H/W of CRYO-DACS

The hardware architecture of CRYO-DACS is singlehost multi-crate distributed VME with CPU(VMIVME-7698) running embedded VxWorks, all linked by workstation clients in 100 Mbps private LAN for distributed logging, historical trending, analysis, alarm management and control GUIs. The crates and modules are procured from M/s VMIC USA. The modules used are Analog inputs, Digital inputs and outputs. The AI modules used are VMIVME 3113-A & 3801 both 12 bit differential ADCs and VMIVME 2536 for digital inputoutputs. The CPU used is INTEL based VMIVME 7698. Two windows PCs have been installed in the central cryogenic control room to act as development and operator consoles.

Front-end Instrumentation

To accomplish the task of low temperature measurements, each cryostat is installed with several silicon diode sensors and are connected to one or more sixteen channel temperature linearizer units which have built-in current sources and 0 to 10V dc linear outputs. Cryogenic level meters with 4-20 mA outputs for level measurements are also developed in-house. Voltage outputs are processed using VME ADCs.

FRCA02

Other I/O Parameters

The main cryostat analog parameters measured are Temperature [4.2 to 350 K], LN2 pressures [0-4 bar], LHe pressures[-1 to +1 bar], vacuum [mbar], LN2 levels [%] and LHe levels [%] whereas the digital inputs are status read backs from valves and vacuum systems. The main Helium compressor parameters acquired are suction & discharged pressures(-15 to +15 PSI), Oil temperatures (deg.C), Power (kWh), loading (%), unloading (%) etc. and a large number of digital input-outputs for loading, unloading, start, stop, reset etc.

THE SOFTWARE

The Software development is done in two parts. The first part i.e. IOWORKS MANAGER, a development center, has been used to configure an NT based host which acts as a development system for VxWorks based targets. The tool VISUAL IOWORKS has been used for the development of real-time VME bus access using ladder-logic which supports a group of libraries for VxWorks target. The six logic modules, each specific to any one cryostat viz. buncher, linac1, linac2, linac3, rebuncher and compressor are hot-swapped into the target online. The second part, a graphics package and OPC client, has been developed in VISUAL BASIC 6.0 for the real time trends, analysis and control GUIs.The main screen of CRYO-DACS is shown in Fig. 2.

The Microsofts OPC (OLE for process control) serverclient communication method is used to collect data from targets and record into an RDBMS backend (ACCESS-2000) and further retrieved using ADO for graphical analysis. The control GUIs to control valves in automatic and manual modes, remote controls of compressors etc. are done from control room consoles. Closed loop controls of LN2, LHe valves have been tested successfully. The LLT and ULT settings of closed loop controls can be dynamically varied anytime online as shown in Fig.2. Many advanced software features are recently added to the system which include an alarm server, currently operated for emergency parameters, with the capability to generate sms automatically.





Networking

A Windows-2000 server has been configured as a host and a gateway. This gateway separates local-cryogenic LAN (192.1.3.XX) from main IUAC-LAN (192.0.3.XX) using route and remote access service (RRAS). Another server running WIN-2000 is configured as a backup domain controller to the above Primary domain controller in CRYODACS-DOMAIN. Both servers run terminal services. Any user who wants to run experiment near to any cryostat can simply connect a LAPTOP/PC running WINDOWS or LINUX-rdeskTop to the LAN port provided near to each cryostat and start logging and analyse data locally.

Signal Conditioning

Output signals from most transducers required conditioning in order to adjust the signal level to be compatible with the assigned 12 bit ADCs in the range of 0-10 V dc. The current signals in the form of 4-20 mA and 0-20 mA are converted to multiple isolated 0 -10 V dc using MTL make signal conditioners. Such 20 signal conditioners are installed in our system. All analog signals are filtered at 40 Hz prior to feeding into ADCs internally. A separate clean earth has been laid only for CRYO-DACS to isolate all electrical noises in the system. The cabling of all the signals (approximately 2000 terminations) are done using individual and overall shielded multicore instrumentation cables. All the VME digital input, outputs are optically isolated 24-28 volt standard.

FAILURES AND REMEDIES

Though VME is proven to be a rugged hardware, there were a few issues related to the hardware failures due to high dust environment. The dust accumulated in VME crates made the power supplies and I/O modules fail. Subsequently further action was taken for the routine maintenance in every three months and hence problems disappeared.

RESULTS

The initial basic system was installed in the year 2002 and then several additions and modifications were carried out from time to time e.g. front end sensor electronics, alarm systems, heater interface, sms etc. The system has been working stable. The hot swapping feature of software modules has been extremely useful, to modify the running system online. The Debug feature to debug or force the variables has also been useful. Maintaining the system needed virus free network as it is operated in windows environment but linux terminal server clients could run cryodacs clients virus free. The bought out VME hardware is always a bottle neck but we have been good as there are no major hardware failures for last 8 years except VME create power supply failures which needed emergency replacement.

is, valve controls					_ @ ×
BUNCHER		UNAC1		UNAC2	
C manual MAN C manual MAN C automatic Fosce C open C close	DIE VALVE C manual MAN C automatic force C open C close	DN2 VLAVE C marual C automatic Iorce C open C close	C menual MAN C sutematic force C open C close	LN2 VLAVE C marcal C automatic Force C open C close	UHE VALVE C monusi MAN C automatic force C open C close
BUNCHER LN2 VALVE	BUNCHER DIE VALVE	UNACT LAZ VALVE	UNACT DHE VALVE		UNACE DHE VALVE
UNAC3		REBUNCHER		REBUNCHER VAL	/E-2
LIKE VALVE Cardinatio Cardinatio Cardinatio Cardinatio Cardination Cardination Cardination Cardination Cardination Cardination Cardination Cardination Cardination Cardination Cardinatio Cardinat	LIF VALVE valoratic valoratic	Tace REBUNCIES IN 22 VANTS Captorials Face REBUNCIES IN 22 VANTS DI 20 VANTS REBUNCIES IN 22 VANTS DI 20 VANTS DI	ET	N Liz VAACE manual	AUTO

Figure 3: The valve control screen.

SUMMARY

The development and operational experience of such a system is that, it is easily designable, expandable, and stable data acquisition and control system where users can quickly setup additional CRYO-DACS clients without modifying any server parameter anywhere in the network. The front-end GUIs can be modified by operator himself as it is easy to be written in VB. The VME hardware has been proved very rugged. A separate private 100MBPS cryogenic LAN is setup only for CRYO-DACS server.

FUTURE PLANS

Considering the past experiences of maintaining CRYO-DACS system and most components of the system is becoming obsolete like. Operating system, VME IOWORKS, VISUAL BASIC and VISUAL C++ tools, hardware VME CPU & Modules etc., we are working out plans to slowly replace the entire system by indigenous hardware, firmware and software being developed inhouse.

Following are some design plans of the newly proposed system:

- 1. To reduce dependency on vendors
- 2. Distributed system should be preferred instead of centralized system to reduce cabling&easy trouble shooting.
- 3. Sensor transfer function should be embedded to device itself, control system PC should not worry about it.
- 4. Remote device reboot feature.
- 5. Inter bus communication between devices w/o any PC.
- 6. All control loops must run embedded in device level itself.
- 7. USB ports in each device for re-programming firmware
- 8. Multiple clients should be able to control and monitoring
- 9. Must support heterogeneous OS & languages like labview, c, c++, java,.NET, HTML, javascript etc.
- 10. Remote variables must be accessible across LAN.
- 11. Compact, low cost SMD design, RoHS, reflow soldered boards to be manufactured

ACKNOWLEDGEMENTS

The authors would like to acknowledge the electrical group of IUAC for their support in providing the electrical interfaces required for CRYO-DACS.

- A. Roy et. al., Proceedings of the 8th International Conference on Heavy Ion Accelerator Technology (HIAT) October 5-9, 1998, Argonne National Laboratory, Argonee, IL, AIP conference proceedings 473, p. 267
- [2] S. Ghosh et al, Physical Review Special Topic– Accelerator and Beams, 12.040101 (2009)

DEVELOPMENT OF THE CAR-BORNE SURVEY SYSTEM KURAMA

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

Abstract

We have developed a car-borne survey system named as KURAMA (Kyoto University RAdiation MApping system) for the establishment of air dose rate map in Fukushima and surrounding area as a response to the nuclear accident at TEPCO Fukushima Daiichi Nuclear Power Plant on March 11, 2011. KURAMA is developed with LabVIEW. The monitoring data tagged by GPS location data are shared with remote servers over 3G mobile network, then processed by servers for a real time plot on Google Earth and other various purposes. A CompactRIObased KURAMA-II is developed for the autonomous operation in public vehicles. More than a hundred of KU-RAMA and KURAMA-II now serve for the drawing up the radiation map in the East Japan by the Ministry of Education, Culture, Sports, Science and Technology in Japan. The outline and present status of KURAMA and KURAMA-II are introduced.

INTRODUCTION

The magnitude-9 earthquake in east Japan and the following massive tsunami caused the serious nuclear disaster of Fukushima Daiichi nuclear power plant, which Japan had never experienced before. Huge amounts of radioactive isotopes were released in Fukushima and the surrounding prefectures.

In such nuclear disasters, air dose rate maps are quite important to help take measures to deal with the incident, such as assessing the radiological dose to the public, making plans for minimizing exposure to the public, or establishing procedures for environmental reclamation. The carborne γ -ray survey technique is known to be one of the effective methods to make air dose-rate maps [1]. In this technique, a continuous radiation measurement with location data throughout the subject area is performed by one or more monitoring cars equipped with radiation detectors. Unfortunately, the existing monitoring system didn't work well in the incident. Such monitoring cars tend to be multifunctional, thus too expensive to own multiple monitoring cars in a prefecture. Fukushima was the case, and to their worse, the only monitoring car and the data center were contaminated by radioactive materials released by the hydrogen explosions of the nuclear power plant.

KURAMA was developed to overcome such difficulties in radiation surveys and for establishing air dose-rate maps during the present incident. KURAMA was designed based on consumer products, enabling a lot of in-vehicle apparatus to be prepared within a short period. KURAMA real-



Figure 1: KURAMA system. Monitoring cars and servers are connected over the Internet by cloud technology.

izes high flexibility in the configuration of data-processing hubs or monitoring cars with the help of cloud technology. In the present paper, an outline of KURAMA as well as discussions regarding car-borne surveys performed by KU-RAMA are presented.

KURAMA

KURAMA is a γ -ray survey system with GPS and up-todate network technologies developed for a primary use of car-borne surveys. A typical configuration of KURAMA is shown in Fig. 1.

An in-vehicle unit of KURAMA consists of a conventional NaI scintillation survey meter with an appropriate energy compensation, an interface box for the analog voltage output of the detector to a USB port of PC, a GPS unit, a laptop PC, and a mobile wi-fi router (Fig. 2). Its simple and compact configuration allows users to set up a in-vehicle unit in a common automobile. The software of in-vehicle part is developed with LabVIEW. The radiation data collected every three seconds is tagged by its respective location data obtained by the Global Positioning System (GPS) and stored in a csv file. The csv files updated by respective monitoring cars are simultaneously shared with remote servers by Dropbox over a 3G network, unlike other typical car-borne survey systems in which a special telemetry system or peer-to-peer radio communication is used. With this feature, the system obtains much more flexibility in the configuration and operation than other conventional car-borne systems. The radiation data is displayed in real time on Google Earth in client PCs after the dynamic generation of KML files in servers (Fig. 3).

KURAMA has served for monitoring activities in Fukushima and surrounding prefectures employed by Fukushima prefectural government and the Ministry of Ed-

^{*}e-mail:tanigaki@rri.kyoto-u.ac.jp



Figure 2: The in-vehicle part is compactly composed of mostly commercial components. 1) GPS unit, 2) 3G mobile wi-fi router, 3) MAKUNOUCHI, 4) NaI survey meter, 5) PC.



Figure 3: The data is simultaneously plotted on Google Earth. The color of each dot represents the air dose rate at respective point.

ucation, Culture, Sports, Science and Technology in Japan (MEXT). The team of Fukushima prefectural government makes precise radiation maps of major cities in Fukushima prefecture mainly for the gHot Spoth search [2], while MEXT periodically performs car-borne surveys in the eastern Japan including Tokyo metropolitan area (Fig. 4) [3].

KURAMA-II

Long term (several tens years) and detailed surveillance of radiations are required in the living areas that are exposed to the radioactive materials. Such monitoring can be realized if moving vehicles in living areas such as buses, delivery vans or bikes for mail delivery have KURAMA onboard. KURAMA-II is designed for such purpose.

KURAMA-II stands on the architecture of KURAMA, but the in-vehicle part is totally re-designed. The platform is based on CompactRIO series of National Instruments to obtain better toughness, stability and compactness. The radiation detection part is replaced from the conventional NaI survey meter to a Hamamatsu C12137 detector, a CsI detector characterized as its compactness, high efficiency, di-

Map of Air Dose Rates on Roads Measured through Vehicle-borne Survey (Whole area)



*The parts indicated in white demarcated with solid lines on the map are covered with snow, and air dose rates over 1 meter above ground level in these areas way norzhite ha lower than when there way no coverna.

Figure 4: The air dose rate mapped by KURAMA in Dec. 2011 [3]. MEXT performed the first and second car-borne surveys of the East-Japan by KURAMA in June and December 2011, respectively.

rect ADC output and USB bus power operation. The direct ADC output enables to obtain γ -ray enegy spectra during the operation. The mobile network and GPS functions are handled by a Gxxx module for CompactRIO by SEA.

The software for KURAMA-II is basically the same code as that of original KURAMA. Additional developments are employed in several components such as device control softwares for newly introduced C12137 detector and Gxxx module, the start up and initialize sequences for autonomous operation, and the file transfer protocol.

As for the file transfer protocol, a simple file transfer protocol based on RESTful was developed since Dropbox doesn't support VxWorks, the operating system of CompactRIO. The present file transfer protocol are required to



Figure 5: The in-vehicle unit of KURAMA-II. A CsI detector and a CompactRIO are compactly placed in a tool box with the size of 34.5 cm 17.5 cm 19.5 cm.



Figure 6: KURAMA-II under the field test on a city bus.

send the data without any lost under the poor coverage of the mobile network in rural part in Fukushima. In this protocol, a chunk of data as a timestamped file in csv format is produced for every three measuring points. The size of a chunk is about 400 bytes typically. Then every chunk is transferred to a remote "gateway server" by POST method. The gateway server combines received chunks to the data file, which is shared by remote servers using Dropbox as did in original KURAMA, and returns the name list of chunks which are successfully combined to the data file. The chunks in CompactRIO will not be deleted unless those names are confirmed in the returned list from the gateway server. Unsent chunks are archived at every one hundred of them as a single zip file and these are sent as soon as the network connection is recovered.

A field test on a city bus has been started since December 2011 in collaboration with Fukushima Kotsu Co. Ltd., one of the largest bus operator in Fukushima prefecture (Fig. 6). City buses are suitable for continuous monitoring purpose because of their fixed routes in the center of living area, and routine operations. Up to now, this field test is successful. We also have found that the present file transfer protocol successfully manages data transmission under the actual mobile network.

Based on the success in the field test on a city bus, MEXT introduced 100 KURAMA-II to municipalities in the eastern Japan and performed the car-borne survey for a month in March 2012 (Fig. 7). The survey was successful and proved the scalability of KURAMA-II system. The second survey in the same area by using KURAMA-II was performed in September 2012, and the result will soon be released from MEXT.

CompactRIO is designed for applications in harsh environments and small places. Therefore, KURAMA-II can be loaded on a motorcycles. This feature enables surveys by motorcycles not only for mail delivery, but also for the monitoring in regions where conventional cars can not drive, such as footpaths between rice fields, paths through forests. Also, KURAMA-II with DGPS unite is about to be used for the precise mapping by walk in rice fields, parks and playgrounds in Fukushima.



Figure 7: Map of air dose rates on roads measured by KURAMA-II [4].

ACKNOWLEDGMENTS

Authors are grateful to Dr. Mizuno, Mr. Abe, Mr. Koyama and staff members of the KURAMA operation team at the Fukushima prefectural government for the continuous support to field tests of KURAMA in Fukushima. The development of KURAMA-II is adopted by "Japan recovery grant program" from National Instruments Japan. Authors are indebted to Mr. and Mrs. Takahashi and the staff members at Matsushimaya Inn, Fukushima, for their heartwarming hospitality during the activities in Fukushima regardless of their severe circumstances due to the earthquake and the following nuclear accident.

- "Guidelines for Radioelement Mapping Using Gamma Ray Spectrometry Data", IAEA-TECDOC-1363, International Atomic Energy Agency, 2003, p. 40.
- [2] http://www.pref.fukushima.jp/j/soukoukekka.htm (in Japanese).
- [3] Press Release, "Results of Continuous Measurement of Air Dose Rates through a Vehicle-borne Survey by MEXT (as of December 2011)", March 21, 2012, MEXT, http://radioactivity.mext.go.jp/en/contents/ 5000/4688/view.html
- [4] Press Release, "Results of Continuous Measurement of Air Dose Rates through a Vehicle-borne Survey by MEXT (as of March 2012)", September 12, 2012, MEXT, http://radioactivity.mext.go.jp/en/contents/ 6000/5637/view.html

CONTOL SYSTEM FOR BARC-TIFR PELLETRON

S. Singh, P. Singh, J. Gore, S. Kulkarni, BARC, Mumbai, India

Abstract

BARC-TIFR Pelletron is a 14 MV tandem accelerator in operation from more than 20 years. It was having a DOS based control system software which was running on a 486 PC and it was not possible to port it on new PCs. It was based on serial highway and Uport adapter based CAMAC crate controller which are now not available and all spares were used. Hence We have changed CAMAC controller with in house developed Ethernet based CAMAC controller and new software has been developed. New Control system software is based on LINUX operating system with Graphical user interface developed using Trolltech's QT API,but can be easily ported on MS Windows.

INTRODUCTION

Pelletron is a 14 MV tandem accelerator located in TIFR, Colaba, Mumbai. It's an electrostatics accelerator which is further augmented by a superconducting booster. Both the facilities have their own control system and it was not possible to integrate both in older control system of pelletron. Control system is a tool to sense the status and to achieve the desired beam from the machine. It consists of control electronics hardware, control software and user interface. It also consists of databases to store system configuration and machine status.

Pelletron control system has been developed on LINUX operating system. LINUX is becoming very popular for particle accelerator control system, because of its stability and Open source nature [1].

SYSTEM ARCHITECTURE

A layered Architecture has been used in this system Figure 1. In a standard way control system can be partitioned in three layers presentation layer (operator interface), application layer (Server Layer/Device control unit) and resource layer (Equipment Interface Unit) [2], [3].

Equipment interface unit is the lowest level of control system which is interconnected to the accelerator equipment and sensors. In Pelletron CAMAC (Computer automated Measurement and control) is used.

Application Layer (Device control unit) provides different services to the other part of the control system like remote booting and configuration of Equipment layer and provides service to operator interface.

Operator interface unit is the layer which connects the accelerator with human and provides interaction with the machine. It consists of many displays (graphical screens, Scopes etc.) and knobs (Virtual panels and hardware knobs) for control and monitoring of the control system.

There is a communication link and communication protocol is the basis of multilayer system. Depending upon interaction between different layers it can be categorised as centralized system or distributed system. The Pelletron control system is a centralized control system. It has three CAMAC crates which are physically distributed at three different places (Ion source room, Bottom of accelerator tank and pelletron beam HALL) and are interconnected via Ethernet LAN communicating via TCP/IP protocol suit. Though the crates are distributed at different places but still it is a centralized control system as all communication between different crates and operator user interface are possible only via a centralized



Figure 1: System Architecture.

Front End Equipment Interface Unit

CAMAC crates are the major constituent of this unit, with ADC, DAC and Digital input /output modules. Field instruments and sensors /actuators are connected to CAMAC cards. Crate Controller is having an embedded processor with CAMAC controller electronics which interacts with CAMAC modules and provides interface to the user (Device Control Unit) over 100 mbps Ethernet link. It provides connection oriented TCP server Socket interface to interact with the CAMAC system. It can be operated in two modes singles mode and list mode. In Singles mode channels from CAMAC modules can be read or written. In list mode a list of NAF can be uploaded and all the devices can be read in single shot. List mode has been used to read all the status and read backs from CAMAC modules and control parameters are operated in singles.

Device Control Unit

This unit consists of a PC running on LINUX. This unit interacts with the all CAMAC crates. It provides services to Operator interface units. Device control unit is responsible for safe and reliable operation of machine. Fault tolerance and redundancy is an important requirement for this unit. This system stores all field values in a fixed memory block, which is persistent till PC is powered off. Because of this persistent memory system can be restarted after any crash or with new configuration without disturbing the current status of the machine. The control system is a configurable system from a system database. Different data base files are created to store the CAMAC configuration, Device and Definition and operator interface signal page configuration are stored in Device control unit.

Device control unit has UPS backup to handle the sudden mains power outage.

Operator Interface Unit

This layer includes Operator interface soft panel running on a PC and hardware panel meters and scopes for monitoring and control of the accelerator subsystems. Different pages have been configured via page configuration database for different subsystems of the accelerator like vacuum page, ion source, beam line etc. Any new page can be created by adding entries in to page configuration database on Device control unit. Virtual panel consists of assignable multifunction slider control for analog control and assignable metres for for read backs. Multifunction sliders can be controlled via mouse or keyboard.

SOFTWARE ARCHITECTURE

Control system software [4] is a database driven system all system parameters and control parameters and their behaviour and look and colours are defined in ASCII database files. These files can be edited by using any spread sheet software or text editor. These database files are stored in password protected area and by default they have only read permission.

Front end instrumentation unit (CAMAC Controller) runs on real time operating system developed by electronics division BARC. It provides a TCP/IP connection at a fixed port.

Device control unit is developed on LINUX operating system following POSIX standard, so that it can be portable on any POSIX compliance operating systems. It is multi-threaded software which uses mutex for synchronization of critical sections. The overall software architecture is given in Figure 2.



Figure 2: Software Architecture.

The Device control software named as scanner is divided in three layers lowest layer is Device interface layer which is a TCP/IP client for CAMAC crates. The no of threads and mutex are variable depending on database configuration. At present there are three crates hence three different threads and mutexes are launched. Each tread is responsible for connection and periodic scan of field data from individual CAMAC units. Command interpreter layer processes the request passed by communication layer and provides appropriate information to Operator interface unit. It communicates to device interface only for changing the values on field device via CAMAC DAC and Digital output units after acquiring the respective mutex.

The communication server layer is responsible for handling the connection request from OIF clients and passes messages and data through and from the OIF clients.

Operator interface is a GUI programme developed in Trolltech's Qt API version 3.0. Qt is a C++ based API which is source code portable between MS Windows and the Linux operating system.

CONCLUSIONS

The System is in operation from past two years and working satisfactorily. It can be operated from existing LINAC control panel, hence providing integrability between Pelletron and LINAC control system.

ACKNOWLEDGEMENT

We sincerely thank Mr Kuldeep jha, Mrs. Anita Behere, and Mr M.P. Diwakar from electronics division, BARC for providing state of Art CAMAC controller modules. We thanks all Pelletron operation staff specially Mr. Ramalal, Mr. Gunekar and Q.A. Ansari for their active support in installation of control system in Pelletron facility.

- [1] A.S. Chepurnov et al., "Operating System Linux as developing and runtime plateform for control system of particle accelerator", Proceeding of EPAC 2000, p. 1832.
- [2] L. Casalegno et al., "Software Development and Maintenance: An approach for large Accelerator Control system", Nuclear Instrumentation Methods in Physics Research A293, p. 268-372.
- [3] B.P. Ajith Kumar et al. "The NSC 16 MV tandem accelerator Control System", Nuclear Instrumentation Methods in Physics Research A 343 (1994), p. 327-330.
- [4] S.K. Schaffer et al., "Software Engineering Practice for Control System Reliability", MOP32, PAC1999, New York.

MAINTAINING AN EFFECTIVE AND EFFICIENT CONTROL SYSTEM FOR THE ELECTROMAGNETIC CALORIMETER OF THE COMPACT MUON SOLENOID EXPERIMENT DURING LONG-TERM OPERATIONS OF CERN'S LARGE HADRON COLLIDER*

O. Holme, D. Di Calafiori, G. Dissertori, W. Lustermann, ETH Zurich, Switzerland S. Zelepoukine, ETH Zurich, Switzerland and University of Wisconsin-Madison, U.S.A.

Abstract

The sub-detectors of the Compact Muon Solenoid (CMS) multi-purpose particle detector at the CERN Large Hadron Collider (LHC) have been collecting physics data from particle collisions for almost three years. During this period, the CMS Electromagnetic Calorimeter (ECAL) Detector Control System (DCS) has contributed to the high level of availability of the experiment. This paper presents the current architecture of this distributed and heterogeneous control system alongside plans and developments for future improvements. To ensure that the system can efficiently operate and adapt to changes throughout the required operation lifetime of more than a decade, the potential legacy aspects of this kind of control system must be carefully managed. Such issues include evolving system requirements, turnover of staff members, potential benefits from new technologies and the need to follow release schedules of external software dependencies. The techniques and results of the work to continually maintain, improve and streamline the control system are presented, including the use of metrics to evaluate the impact of this effort.

INTRODUCTION

The CMS ECAL consists of three partitions: the barrel (EB), the endcaps (EE) and the preshower (ES). The EB and EE are based on lead tungstate scintillator crystals, with the generated light being measured by Avalanche Photo Diodes (APDs) and Vacuum Photo Triodes (VPTs) respectively. The ES uses reversed biased silicon sensors as the detection devices. The differing technologies of the three partitions imply specific control system needs; however there are common features such as the requirement for low voltage (LV) to power the data acquisition (DAQ) electronics and bias voltage (BV) to bias the particular type of sensor. All CMS ECAL partitions are vulnerable to overheating and require constant cooling during operation. Additionally, the humidity level of the air in the detector can provide further important information. For these reasons, the cooling systems and environmental conditions inside each of the partitions must be carefully monitored.

The prevention of damage to the detector is delegated to Programmable Logic Controller (PLC) based safety systems [1], which monitor detector conditions and can act by interlocking the powering devices and the cooling systems. The DCS monitors the same data as used by the safety PLCs and can take pre-emptive actions before safety actions would be required. Importantly, this enables less abrupt actions to be performed, reducing risks of hardware damage and in some cases speeding up recovery to a fully operational state.

ARCHITECTURE

The CMS ECAL DCS consists of both software and hardware elements. The architecture was implemented with existing designs and technology where possible, either from commercial vendors or from high energy physics collaborations. This approach accelerated and simplified the implementation and continues to have a positive impact on the maintenance load of the system.

The role of the hardware is to host the software and to gather or send the data required to operate and monitor the detector. Currently one DELL PE2950 and 14 DELL PE1950 servers running Windows XP are used to execute the DCS software. The readout of data is performed using Ethernet and Controller Area Network (CAN) bus, according to the supported interfaces of existing hardware. In order to connect the CAN buses with the server PCs a suitable interfacing device must be chosen, with the existing solution being a CAN to USB adapter.

The BV and LV power supplies used in the CMS ECAL DCS are provided by commercial vendors together with Open Platform Communications Data Access (OPC DA) servers for monitoring and controlling the devices. The safety systems are based on Siemens PLCs and the standard S7 readout over Ethernet is used. For the remaining monitoring systems, custom electronics have been developed to read out several types of sensor inside the detector, such as the negative temperature coefficient (NTC) probes used to precisely monitor the EB and EE environmental temperature. Where possible, development of completely custom electronics was avoided by making use of the Embedded Local Monitor Board (ELMB) [2] which is an input/output electronics board equipped with a CAN transceiver. Use of the ELMB enabled a considerable simplification in the DCS electronics design.

The software layer is used to combine data from all the sources and handle it according to the requirements. This data handling includes:

- Filtering and smoothing.
- Archiving.
- Generating alarms on abnormal conditions.
- Visualisation of the data in user interfaces panels, including graphical representations and plots.

^{*}Work supported by the Swiss National Science Foundation

The software is built with the commercial Supervisory Control and Data Acquisition (SCADA) toolkit called Simatic WinCC Open Architecture (WinCC OA) 3.8 SP2 developed by ETM Professional Control [3]. In addition, a framework of tools, called the Joint COntrols Project FrameWork (JCOP FW) [4] and developed by CERN in collaboration with LHC experiments, was used to assist the development with WinCC OA in the high-energy physics domain.

The CMS ECAL DCS software is divided into several applications, with each focussed primarily on one system or device type. For example, separate applications are provided for interacting with the BV, LV, safety system PLCs and the cooling system. This concept of modular applications enables parts of the DCS to be developed independently, giving flexibility in allocation of tasks to developers as well as simplifying the testing procedures.

CONSOLIDATION PHASE

The CMS ECAL DCS was developed over several years in parallel with detector construction activities. Early prototypes were essential for testing all detector components before installation in CMS and experience from these prototypes was incorporated into later versions. Due to the length of the development period and the fact that several of the team members were involved for only short periods, there was a large turnover of developers. As the project moved from a development to maintenance phase when the LHC started physics operations in 2009, the number of developers was reduced, leading to a significant support load that had to be lessened in order to be sustainable in the long term.

The applications, developed in parallel by different developers, had evolved to varying levels of maturity and were based on distinct programming styles and development philosophies. As a result, it was apparent that a phase of consolidation could produce a more homogenous, higher quality software that was simpler to maintain and more robust to failure while still providing the same level of functionality.

The consolidation of the software was initiated by a software analysis project [5], aimed at homogenising the application implementations, factoring out common functionality into reusable modules and removing features that were no longer relevant. This process yielded a significantly smaller code base with considerably fewer differences in design and development style between the individual applications of the ECAL DCS.

The control system requirements continue to develop and grow with time as the system is methodically improved in response to operational experience. For this reason, there is continual development of new software and measures had to be put into place to ensure that these new developments did not counteract the gains made with the software consolidation. To achieve this, a continuous quality assessment methodology was introduced with the code quality regularly being measured in terms of specified metrics.

Code Quality Assessment

The tool chosen to perform this evaluation was the Continuous Quality Assessment Toolkit (ConQAT) [6]. ConQAT analyses code against defined metrics and produces visual representations of the results, making it easy for developers to identify and resolve issues where code does not meet the required standards. A new component was developed by the CMS ECAL DCS team to enable ConQAT to assess software developed with WinCC OA and the JCOP FW. As an additional benefit of this integration between ConQAT and WinCC OA, it became possible to precisely quantify the DCS software in terms of physical source lines of code (SLOC).

The following metrics were chosen to assess the quality of the code and indicate areas to improve:

- SLOC in segments of code that are duplicated.
- Functions exceeding a given number of source lines.
- Files with code block nesting level over a threshold.

The acceptable thresholds to determine these values were initially chosen based on the existing state of the code in order to ensure that a significant, but not overwhelming, number of potential improvements would be identified. As the code quality improved, the thresholds were adjusted to be more demanding and to continue to drive the improvement in code quality.

Using the latest thresholds, it is possible to reanalyse previous software versions stored in the CMS ECAL DCS Subversion software repository to chart the software quantity and quality over time, as shown in Table 1.

Table 1: Evolution of ECAL DCS Code Metrics with Time

Metric	January 2011	October 2011	April 2012	October 2012
Total SLOC	67,532	59,665	39,044	37,124
Duplicated SLOC	31,237	26,328	7,386	6,623
Long functions	183	127	81	54
Files with high nested block level	40	33	25	18

It is interesting to observe that before the introduction of ConQAT in November 2011, a significant reduction in code quantity and duplication had occurred, which was due to the consolidation of the code based on manual inspection. However, a large amount of duplicated code had not been detected and ConQAT enabled the discovery, visualisation and resolution of these issues.

The process of improving the code quality is an ongoing task that is performed in parallel to maintenance, operation and development work. While removing duplicated segments of code, it was noted several times that bugs had been fixed in one place in the code but not in a duplicated segment elsewhere in the software. This type of issue highlights the importance of the continuous quality approach in reducing the number of bugs and making the code easier to maintain in the future.

CHANGING REOUIREMENTS

During the operation of the CMS ECAL DCS, there has been significant learning about the behaviour of the detector. This knowledge has been used to extend the control system to get further knowledge of the detector characteristics and to enable easier operation. For instance, more detailed monitoring of the EE and ES BV distribution has been added since the start of LHC operations. Additionally, the original humidity monitoring of the EB and EE is being upgraded to improve the readout range.

This evolution of the DCS scope is a significant driver for change and has to be carefully managed to ensure that the system remains maintainable. The approach to deal with this is to encourage the reuse of known hardware technologies and to adopt communication protocols that are supported by WinCC OA as standard. By enforcing this strategy, the CMS ECAL DCS team can extend the functionality while making maximum benefit of existing knowledge and ensuring that the system does not become increasingly heterogeneous and difficult to support.

As an example, ELMBs, which were already used in the EB and EE temperature monitoring, were chosen for the BV distribution monitoring systems. However, for the improved humidity readout hardware, a fully custom electronic front end was necessary. To make sure that the software could easily access this system, the standard MODBUS protocol was specified, as this is supported by WinCC OA and could easily be implemented on the microcontroller in the front end electronics.

NEW TECHNOLOGIES

A major challenge in 2012 has been to prepare for the next generation of computers, selected by CMS, to host the DCS. Due to the significant increase in memory and processing power of these DELL M610 blade servers, it will be possible to run the complete CMS ECAL DCS on four machines. The new computers will run Windows 7 and it is anticipated that a new version of WinCC OA will be adopted in the near future. These software upgrades are essential in order to avoid running with commercial software versions that are no longer supported.

In addition to the obvious need for testing and revalidation associated with the new hardware and software platform, this change also requires the merging of applications onto fewer machines. In turn, this requires that the applications are compatible to run side-by-side with each other. This was not a requirement in the past development of the software and investigations exposed several incompatibilities, mostly relating to object name clashes, which had to be resolved.

Another significant improvement planned for the CMS DCS is the move to a redundant WinCC OA © configuration where each application runs simultaneously on two machines. With two running instances, a failure of a single host PC will no longer degrade functionality. The most significant impact of this new technology is that all DCS data communication must be via Ethernet so that either of the redundant instances can access all devices when required. This has a particular impact on the current CAN bus interface, which is based on USB. Work is under way to evaluate a suitable interface that enables CAN readout via Ethernet [1].

To keep pace with the opportunities and challenges associated with new technologies, a policy of active research and development has been a priority in order to discover and resolve issues well before the planned deployment in CMS. A replica laboratory setup of the new computer platform has been purchased and configured to develop, test and certify future software versions and hardware interfaces. Previous experience has proven that this methodology minimises problems and downtime following new DCS component deployments.

CONCLUSION

From the initial design phase of the CMS ECAL DCS, the system has made optimal use of pre-existing hardware implementations, software frameworks and commercial products. The success of this decision has been demonstrated by the excellent performance during more than four years of LHC operations. As new requirements emerge, the same approach is adopted to reuse existing technologies to take maximum benefit from experience and prevent increasing heterogeneity of the system.

To ensure the software remains maintainable, a continuous quality control approach has been implemented and the resolution of existing quality issues is a persistent background development task.

With these approaches and active research and development into new technologies, the maintenance load is controlled and the benefits of new technologies can be gained without compromising the robustness and effectiveness of the system.

REFERENCES

- [1] D. Di Calafiori et al., "Maintaining and improving safety the control and systems for the Electromagnetic Calorimeter of the CMS experiment", CHEP'2012, New York, U.S.A. (2012).
- [2] H. Boterenbrood and B. Hallgran, "The Development of the Embedded Local Monitor Board (ELMB)", 9th Workshop on Electronics for LHC Experiments, Amsterdam, Netherlands, (2003).
- [3] ETM professional control GmbH; http://www.etm.at
- [4] O. Holme et al., "The JCOP Framework", ICALEPCS'2005, Geneva, Switzerland, (2005).
- [5] O. Holme et al., "The Evolution of the Control System for the Electromagnetic Calorimeter of the Compact Muon Solenoid Experiment at the Large Hadron Collider", ICALEPCS'2011, Grenoble, France, (2011).
- [6] ConOAT, Continuous Quality Assessment Toolkit; http://www.congat.org

the

0

DEVELOPMENT OF THE CONTROL SYSTEM FOR PEFP 100 MeV PROTON LINEAR ACCELERATOR*

Y.G. Song[#], H.J. Kwon, J.H. Jang, Y.S. Cho, KAERI, Daejon, Korea

Abstract

The 100 MeV proton linear accelerator of the Proton Engineering Frontier Project (PEFP) has been developed and will be installed in Gyeong-ju site. After the installation, the beam commissioning of the 100 MeV linac will be performed. The PEFP is currently developing control systems including the machine control system and user interface for remote control and monitoring. The final goal of the PEFP control system is to construct a network attached, distributed control system, and a standard communication protocol among the local subsystems. In this paper, we will present the details of the distributed control system development for PEFP 100 MeV proton linac.

INTRODUCTION

Proton Engineering Frontier Project is the 100 MeV proton linac development project which was launched at 2002 as a 21st century frontier R&D program of Korean government [1, 2]. The final goals of the project are constructing a proton linear accelerator with the final energy of 100 MeV and the peak beam current of 20 mA, developing technologies for the proton beam utilizations and the accelerator applications, and promoting industrial applications with the developed technologies.

The PEFP 100 MeV proton linac consists of two parts as shown in Fig. 1. One is the low energy components including a 50 keV ion source, a low energy beam transport (LEBT), a 3 MeV radio frequency quadrupole (RFQ), and a 20 MeV drift tube linac (DTL). The other part is the high energy components including seven DTL tanks which accelerator proton beams from 20 MeV to 100 MeV. The PEFP experiment hall includes 10 beam lines, 5 for 20 MeV beams and 5 for 100 MeV beams. One of the main characteristics of PEFP beam lines is using AC magnet to distribute proton beams into 3 target rooms in both 20 MeV and 100 MeV beam lines.



Figure 1: PEFP 100 MeV linac and extraction beam lines.

The 20 MeV linac has been successfully installed and

tested at the Daejeon site of Korea Atomic Energy Research Institute (KAERI). In this operation of the linac from 2007 to 2011, the 20 MeV linac control systems were used to control and monitor the facility for 5 years. The control system consists of ion source, vacuum, magnet power supply, and low-level RF. The 20 MeV linac control systems have been disintegrated and moved to Gyeongju site. Some of the control systems for 20 MeV linac will be expanded for 100 MeV proton linac and the control systems including vacuum, beam monitoring, and timing are upgraded. There are also many tools that can be used to make operator interface and to make an archiving system.

CONTROL SYSTEM OVERVIEW

For the networked control system, network architecture is very important as a key infrastructure which dominates total performance. Backbone switches and workgroup switches are installed as shown in Fig. 2. The remote control systems of the local devices should be connected to a local control network for Ethernet communication. Most control systems are developed with VME-based IOCs which include VME I/O boards and VME PMC boards. The integrated operation will be realized by the network connections and by the interactions among control systems.



Figure 2: Schematic diagram of network architecture for PEFP 100 MeV proton linear accelerator.

The verified development tools, control infrastructure, and efficient maintenance are requirements for the PEFP software development tool. The EPICS tool provides the Channel Access (CA) communication protocol to make TCP/IP connections and transfer process variables among controllers [3, 4]. Figure 3 shows the schematic layout for the PEFP control system which is based on local control network, interlock network, and timing network. Most of communication is based on CA proton for a distributed control system.

^{*}This work was supported by Ministry of Education, Science and

Technology of the Korean government

[#]ygsong@kaeri.re.kr



Figure 3: Block diagram of networked control system.

Timing System

The 20-MeV timing system is composed of pulse delay generators, including DG535 from Stanford Research Systems, BNC565 from Berkeley Nucleonic Corporation, and fan-outs that distribute a trigger delivered from the delay generator. The stability can be improved by changing the DG535 from an internal to an external reference signal of a 10-MHz reference clock supplied by the master oscillator. However, this timing system has a defect providing the fixed clock and trigger from a pulse delay generator. This means the fan-out distributes a fixed delay and width of the timing signal received from a delay generator, with is not changeable on each channel. The timing system is required to improve the performance and functionality to deliver the diverse timing signals to the accelerator facility and beam lines.

Hence, we decided to upgrade the timing system for efficiency and reliability. Basically the timing system provides clock synchronization, RF gate generation, beam gate generation, triggering for beam diagnostics, and AC magnet gate generation. MRF event timing system has been decided as a main timing system, which can be triggered out of different clocks with a very low jitter. The event system consists of a VMEEVG-230, seven VME-EVR-230RFs with RF recovery, a VME-FOUT-12, TTL, and optical fiber output modules. Figure 4 shows a schematic layout of the timing system using MRF event system.



Figure 4: Architecture of upgraded timing system.

The event system has various types of outputs to avoid external level converters. Most triggers require TTL levels terminated into 50 Ω and a few optical links. All triggers should have a low jitter performance of a few hundreds of picoseconds.

Data Acquisition

The proton linac has various electronic devices, such as beam current monitor (BCM), beam position monitor (BPM), and beam loss monitor (BLM). The Log-ratio BPM electronics module of the Bergoz Instrumentation directs beam position from the pickup signals. The signal from the BPM is through the 350 MHz band pass filter and signal divider for measured beam phase, after then the x-axis and y-axis position signals are produced from electronics modules. At 100MeV proton accelerator operation, 12 BLM are installed to inform the beam loss to operator and transmit the signal to the MPS (Machine Protection System).

For data acquisition, Industrial Pack (IP) ADC boards are adopted. The IP ADC has the specification like 16 bit resolution, 16 channel, and Max 200k sampling rate. A VME carrier board makes it possible to equip a total of four IP ADCs. The ADC can process a total of 64 input signals. Figure 5 shows schematic layout for beam monitoring.



Figure 5: Schematics layout of EPICS system for beam monitoring.

Application Software

In the control room, console machine must support several functions for the linac remote operation and the entire management. The PEFP operator interface systems support the remote control and monitoring of the operation parameters from the local control systems, such as beam, radio frequency, vacuum, magnet power supply, cooling, ion source, etc.

For 20 MeV linac operations, EPISC Extensions tools were used as shown in Fig. 6. It includes EDM, Strip tool, and ALH. The storage system was based on MySQL and Web viewer using EPICS command line tool as shown in Fig. 7.



Figure 6: Operator interfaces for 20-MeV proton linac.



Figure 7: Web viewer for 20-MeV proton linac.

For the PEFP proton linac operation, an eclipse-based Control System Studio (CSS) will be used [5]. We have studied how the CSS can be used for the operator interfaces and which features in CSS can be used for specific control requirements. The RDB Archiver with MySQL and the alarm viewer using CSS is under test as shown in Fig. 8 and 9.

To reduce communication traffic on the control network, OPIs and IOCs are connected by using the CA gateway [6]. Internally, the CA gateway makes a virtual connection to reduce the number of CA connections on the EPICS IOC.



Figure 8: CSS viewer with RDB and archive engine.

siam Area Pain	ALC: NO						
	1	25 Alarm Table 1					A 1 1 1 1 1
		Current Alarma (2	1		Delect		100
	Terms	FY Take (Second Sec. (Name	Description galcadense Rang	Atom Tore 2020/2011/2022	Current Co	ner A - Aser Ada - Marine - Marine Marine - Marine - Marine Marine - Marine	Autorian Autorian
Karn Tree (1							
te • Area Tomate							
· Pic lines							
		Actrowindped At	arme 13			2	

Figure 9: CSS alarm viewer.

The PEFP control system using the EPICS CA protocol allows operators in the control room to focus an accelerator operation and control management easily and efficiently.

CONCLUSION

The 20 MeV linac control system was developed to operate the linac for 5 years and debugged and upgraded for 100 MeV linac control system. The 20 MeV linac control systems have been extended for the PEFP 100 MeV proton linac. The 100 MeV proton accelerator control system has been developed with EPICS tool. All of the IOCs can be integrated by using CA protocol. There are also new control system like timing system and beam monitoring system. The control system will be commissioned and used for beam commissioning. In the future, the control system will be continually managed and upgraded.

ACKNOWLEDGEMENTS

This work is supported by the Ministry of Education, Science and Technology of the Korean Government.

- B. H. Choi, "The Proton Engineering Frontier Project", Proc. of International Particle Accelerator Conference, Kyoto, Japan, 2010.
- [2] H. J. Kwon, "Operation of the PEFP 20MeV Proton Linac at KAERI", Proc. of International Particle Accelerator Conference, K-yoto, Japan, 2010.
- [3] Martin R. Kraimer: "EPICS IOC Application Developer's Guide, "APS/ANL, 1998.
- [4] Philp Stanley, et al.: "EPICS Record Reference Manual", LANL, APS/ANL, 1995.
- [5] K. Kasemir, "Control System Studio Applications", ICALEPCS'07, Knoxville, Oct. 2007.
- [6] Kenneth Evans and Martin Smith, "Experience with the EPICS PV Gateway at the APS," Proc. 2005 Particle Accelerator Conf., Knoxville, TN, May 16 -20, 2005, to be published.

RF CONTROL SYSTEM FOR 400 keV RFQ

Gopal Joshi, Shyam Mohan, Sandeep Kashinath Bharade, Motiwala Paresh, C.I. Sujo, T.S. Ananthkrishnan, Chandra Kant Pithawa Electronics Division, Bhabha Atomic Research Centre, Mumbai

Abstract

An RF control system has been developed for the 400 keV, 350 MHz RFQ coming up at BARC. This single cavity system consists of the functionalities of amplitude stabilization and frequency tracking for both continuous and pulsed mode of operation. The amplitude stabilization is implemented by modulating the attenuation across a fast modulator placed in the drive path. The frequency tracking is achieved by driving the FM port of a signal generator with a signal proportional to the phase shift across the resonator. The whole system is under computer control via CAMAC hardware. The paper describes the system architecture, housing & wiring of the system in a single instrumentation rack and development & testing of computer control.

INTRODUCTION

A 400 keV, 350 MHz Radio Frequency Quadrupole (RFQ) accelerator is being built at BARC, Mumbai as part of LEHIPA project. An RF control system is under development for this accelerator, which has only a single resonator, at Electronics Division, BARC [1]. The functionalities of amplitude stabilisation and frequency tracking are incorporated in the system [2]. The amplitude stabilisation is needed for a stable energy gain from the resonator. Frequency tracking system keeps the drive source-resonator system at resonance, leading to an efficient utilization of RF power to set up electric field in the resonator. Close to the resonant frequency, the phase difference between the transmitted wave and the forward wave is a sensitive indicator of the resonance condition. The system utilizes this phase difference to track the resonant frequency of the resonator by modulating the output frequency of the signal generator.

The RF control electronics is divided in a number of simple signal processing modules. This sub-division has made a number of signals available for monitoring during operation and setting up and also allowed parallel development of the hardware. These signal processing modules are under computer control via CAMAC interface.

SYSTEM DESCRIPTION

Figure 1 shows the block diagram of the overall RF control. The signal generator, operating at a constant output level, feeds RF power to the resonator via amplitude control elements. The amplitude loop compares the pick-up in the resonator with set point and generates the drive signal for the amplitude modulator. The quiescent values of the drive to the two power amplifiers used in the system can be independently set in both

amplitude and phase. For frequency tracking the resonator pick-up and the drive signal are down-converted to a convenient value of 10 MHz. The vector modulator generates the 10 MHz offset frequency required for downconversion. The frequency loop generates the drive signal for the fm port of the signal generates by comparing the phase difference between these 10 MHz signals. The pulse generator, a part of the RF signal generator, is used for the pulsed operation of the system. All the control and monitoring signals of the amplitude and frequency loops are under computer control via CAMAC interface. The system utilises a number of functionalities available in the signal generator. These are under computer control as well. In Fig. 2 we show the control system view of the Low Level RF (LLRF) system.



Fig. 1: RF Control System - hardware



Figure 2: RF Control System – computer control.

CONTROL ARCHITECTURE

The control system is designed as 3 tier architecture as depicted in Fig 3. The OWS layer hosts the presentation applications and run manager which is used by operators

for machine run. The middle layer is the server layer where servers like command control, parameter and configuration are hosted. The command control server is responsible for permitting commands to the equipment server depending upon macro level machine state logic. Parameter server is data concentrator which collects data from different equipment servers, arrange data as per properties and present it to OWS layer. The equipment layer is the lowest layer. It is presented to the server and OWS layer through standard interface, it receives commands from higher layers and multicast its parameters. It hosts the access to the hardware.



Figure 3: Control Architecture.

LLRF EQUIPMENT SERVER

The LLRF equipment server is implemented using standard equipment server framework as depicted in Fig. 4. The LLRF equipment server downloads system configuration from Configuration server. The system configuration contains system parameters information like physical connection, type, safe output value, conversion etc. The Network Manager listens on command port and multicast parameters on parameter port. The scanner thread when spawned will scan input parameters with configured periodicity and stores them at shared memory. The Command Executor thread processes command and drive output parameters. It also stores changed output parameters in the shared memory. Shared Memory usage initially helps for getting safe output parameters for the system. Secondly for some reason thread exit and respawns then the current machine context is retained in shared memory, so the system can resume its functionality without disturbing present machine state. The hardware interface thread listens on message channel; it allows setting and getting parameters. This helps in making interface hardware independent. Presently the LLRF crates are CAMAC system. CPCI and VME based DLLRF hardware is under development. This design will ensure minimal effort transition to the new hardware system in future.



Figure 4: Equipment Server.

LLRF INSTRUMENTATION RACK

The complete signal processing is divided into a number of simple modules. These modules are housed in an instrumentation BIN as shown in Fig. 4. This subdivision has made a large number of signals available on the front panel for monitoring. This has proved very useful during testing in the lab. It has also made possible a parallel development of simple signal processing modules. The RF bin is part of an instrumentation rack, which also houses, a signal generator, signal router, control computer and CAMAC bin. The signal router collects various monitoring and control signals from all the modules and depending on the type of the signal (analog I/O, digital I/O) sends it to the respective CAMAC module. The organisation of these hardware blocks is shown in Figure 5. Figure 6 shows the instrumentation rack, housing the required subsystems.



Figure 5: RF BIN.



Figure 6: Organisation of RF Instrumentation RACK.



Figure 7: Instrumentation RACK.

LAB TEST SETUP

Various parameters such as system state continuous/conditioning, feedbacks on/off, signal generator frequency/level, signal path attenuation, phase shift, feedback gains etc., are set using the front-end application. A number of system parameters, such as, Cavity field, Forward power, Frequency error or detuning etc. are displayed on the control computer. Figure 8 shows the operator screen. RF and high band-width analog signals are available on the front panel of the modules. The system has been tested at low power in the lab using a simple test 350 MHz resonator. Presently the system is being used for the pulse-conditioning of the RF power coupler, coupled to a test cavity.



Figure 8: Operator Screen.

ACKNOWLEDGMENT

We are thankful to Shri G.P. Srivastava, Director Electronics and Instrumentation Group, BARC for the encouragement and support.

- [1] Gopal Joshi, C.I. Sujo, P.D. Motiwala, Shyam Mohan, P. Singh, Girish Kumar and Vivek Agarwal, "RF Control Electronics for FOR 400 keV RFQ", NSNI 2010, BARC, Mumbai.
- [2] Gopal Joshi, P. Singh, Vivek Agarwal, Girish Kumar, "Resonant Frequency Tracking System", Indian Particle Accelerator Conference, Indore, February 10-13, 2009.

VEPP-2000 COLLIDER CONTROL SYSTEM*

A.Senchenko^{1,#}, D.Berkaev^{1,2}, O.Gorbatenko¹, A.Kasaev¹, I.Koop^{1,2}, V.Kozak¹, A.Kyrpotin¹,

A. Lysenko¹, Yu. Rogovsky^{1,2}, A.Romanov¹, P. Shatunov¹, A. Stankevich¹, Yu. Shatunov^{1,2}

¹BINP SB RAS, Novosibirsk, Russia

²Novosibirsk State University, Novosibirsk, Russia

Abstract

Electron-positron collider VEPP-2000 has been commissioned at Budker Institute of Nuclear Physics. The first experiments on high energy physics has been started at the end of 2009. The paper presents architecture, implementation and functionality of hardware and software of the collider control system. The hardware of the system consists of high current main field power supplies, steering coils power supplies, pulse-elements, RF subsystems and some other special subsystems (such as vacuum, temperature, etc.). The system is based on modern industrial protocol CAN-bus and specialized electronic BINP manufactured blocks according the standard. The paper describes implementation of different subsystems based on CANbus devices, and operating characteristics and their possibilities. Other standards and protocols like CAMAC, VME and so on also used in the system. The software according to hardware system consists of interacting subsystems responding on different acceleration facility parts. Control system software is based on several TCP/IP connected PC platforms under operating system Linux and uses client-server techniques.

VEPP-2000 PROJECT

VEPP-2000 is a new collider with luminosity up to 10^{32} cm⁻²s⁻¹ and the beam energy up to 2×1 GeV [1, 2].



Figure 1: VEPP-2000 facility layout.

This project is a development of a previous facility of VEPP-2M which has worked at BINP over 25 years in energy range up to 1.4 GeV in c.m.s. and has collected of about 75 pb⁻¹ integrated luminosity. New collider uses the existing beam production chain of accelerators: ILU – a pulsed RF cavity with a voltage of 2.5 - 3 MeV, a 250 MeV synchrotron B-3M and a booster storage ring BEP with the maximum project beam energy of 900 MeV (see Figure 1 and Table 1). The lattice of VEPP-2000 has a

senchenko.alexander@gmail.com

two-fold symmetry with two experimental straight sections of 3m length, where Cryogenic Magnetic Detector [3] and Spherical Neutral Detector [4] are located. Two other long straights (2.5m) are designed for injection of beams and RF cavity, and 4 short technical straight sections accommodate triplets of quadrupole.

To avoid dispersion in the detectors, RF cavity and injection straights, a pair of dipoles together with the triplet in between constitute 4 achromats. Chromaticity corrections are performed by two families of sextupole magnets located in the technical straight section, where the dispersion is high.

The closed orbit steering and gradient corrections are done with 1-2% coils placed in the dipole and quadrupole magnets.

The accelerating HOM-damping RF cavity operates at the 14-th harmonic of the revolution frequency (172.0MHz). It provides for a bunch length of about 3 cm at the top energy and stability of design bunch current of 200 mA.

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, there are also 4 pick-up stations in the technical straight sections and one current transformer as an absolute current monitor.

The magnetic field of 2.4 T in the bends is required to reach the design energy of 1 GeV in the constrained area of the experimental hall.

Table 1:	VEPP-2000	Collider	Main Parameters

Parameter	Value
Circumference, П	24.39 m
Energy, E	1 GeV
Betatron functions at IP, $\beta^*_{x,z}$	10 cm
Betatron tunes, $v_{x,z}$	4.1, 2.1
Beam emittance, ε	$1.4 \times 10^{-7} \text{ m} \cdot \text{rad}$
Particles per beam, N	1×10 ¹¹
Momentum compaction, α	0.036
Synchrotron tune, v _s	0.0035
Energy spread, $\sigma_{\Delta E/E}$	6.4×10 ⁻⁴
Beam-beam parameters, ξ	0.075
Luminosity, L	$10^{32} \text{ cm}^{-2} \text{s}^{-1}$

^{*} Work partially supported by Russian Ministry of Education and Science, basic project of BINP SB RAS 13.3.1, Physics branch of RAS project OFN.1.1.2, Scientific school NS-5207.2912.2 and Grant of the Novosibirsk region Government 2012



Figure 2: VEPP-2000 acceleration complex network layout.

VEPP-2000 HARDWARE

From the point of view of automation accelerator complex VEPP-2000 is a complicated system. Over than 1200 control channels and 2400 monitoring channels and their joint usage apply rigid restriction on control system.

Acceleration complex VEPP-2000 consists of five main subsystems: ILU and channel to B-3M, B-3M and channel to BEP, BEP booster ring, channels from BEP to VEPP-2000 for electrons and positrons and VEPP-2000 collider ring. Each of subsystems includes several subsubsystems: pulse elements, steady elements, RF-systems, and some additional systems like vacuum, temperature or radiation monitoring, as well as beam parameters measurement system which description is beyond the scope of this paper.

The control system of the VEPP-2000 complex is based on several PC platforms under Linux operating system, connected into VEPP-2000 private net. The layout of the system is presented in Figure 2.

As it was mentioned above VEPP-2000 complex is a modernization of old VEPP-2M facility. This leads to necessity of taking into account a wealth of 20-year experience on the one hand and using modern automation solutions on the other hand.

On choice of hardware protocol one should be guided by capacity, support and standards. For automation system for VEPP-2000 facility two main protocols was chosen: well-known CAMAC and modern CAN-Bus [5]. Table 2 specifies subsystems and used protocols with approximate number of channels.

Control units, used at VEPP-2000 complex, are made in BINP [6,7]. CAMAC standard is used at the places where heavy traffic is needed (fast ADCs) and also at the systems where migration on a new standards during modernization was considered inadequate (old systems like ILU or B-3M) which will be removed in future. CAN-Bus protocol is used at the complex automation as a base protocol. It is very convenient for spatially distributed automation systems and allows reduce wire commutations significantly. More one standard – VME – is used only in beam parameters measurement systems: pickups, etc. A very simplified scheme of the control system hardware is shown in Figure 3. Beam measurement hardware of VEPP-2000 is beyond this paper so it is not shown in the figure in detail¹.

Table 2: Subsystems and Channels

System	Subsystem	Protocol	N of chan.
ILU & channel	Pulse	CAMAC	50
		CAN-Bus	40
	DC	CAN-Bus	40
B-3M &	Pulse	CAMAC	50
channel		CAN-Bus	40
	DC	CAN-Bus	40
	RF	CAN-Bus	20
BEP	Steady	CAN-Bus	500
	RF	CAN-Bus	20
BEP-VEPP-	Pulse	CAMAC	30
2000 channels		CAN-Bus	100
	DC	CAN-Bus	50
VEPP-2000	DC	CAN-Bus	500
	RF	CAN-Bus	20
Technological	Vacuum	CAN-Bus	50
	Temperature	CAN-Bus	80
	Cryogenics	CAN-Bus	50
	Radiation	CAMAC	10

VEPP-2000 SOFTWARE ARCHITECTURE

Principles of software construction arose from the hardware architecture. The architecture of VEPP-2000

[[] More information about VEPP2-2000 Beam Measurement Systems one can find at [8,9].

Proceedings of PCaPAC2012, Kolkata, India



Figure 3: VEPP-2000 Control hardware.

software is based on traditional three-layer structure. Specialized services (hardware layer) control one or

several CAN or CAMAC buses and allow client applications to have access to hardware.

The main application of Middleware layer is VCAS (VEPP-2000 Channel Access Server). VCAS is a universal communication media, which provides access to the data to any other application in the control system or to the other purposes, i.e. for detectors control systems or for status pages in the Web, etc.

Third level is presented with GUI applications, which provides to facility operator powerful and convenient instrumentation for beam tuning and diagnostics of possible systems malfunctioning.

For the high loaded data channels like control systems of magnetic structure of storage and collider rings it is possible direct communication between GUI and hardware layers.

Another important application in the middleware layer in specially designed Log Server [10]. Its purpose is to archive all necessary automation system data to special hard disk storage. All foresaid is schematically shown on Figure 4.



Figure 4: VEPP-2000 Software scheme.

The point is to hide from operator details of hardware configuration and implementation of each hardware service. All information about hardware configuration contains in special databases or configuration files, personal for each server. Each control system application can store its own data of properties in personal databases or files if it is necessary.

The second point is modular principle of VEPP-2000 automation system construction. This allows including new features and hardware without significant reconstruction of on-line operating software from one hand and fast and clean searching and removal of any hardware of software faults.

Thus, software system allows facility operator to manage and monitor different hardware subsystems sometimes very different from each other in one manner. Operator uses very similar applications GUI-styled as well as console-like to steer different parts of VEPP-2000 acceleration complex.

VCAS APPLICATION

The key concept of VCAS application architecture is the *Data channel* concept. Data channel is named data storage and distribution unit which has several properties addition to the unique name: value, units, description, type and some others. Properties of VCAS data channel are described in Table 3.

Application, needed to manipulate with VCAS channels data, can set value to the channel, get value, get full channel data including all channel properties, subscribe the channel (data will arrive to application only being set to VCAS), release (after subscribe, when the channel is no more needed) and free (for exclusive channels).

There are some utility data channels of VCAS itself: "*ChannelsList*", containing the list of the names of all VCAS data channel, "*error*", which represents the last error in VCAS operation and others.

The schematic of VCAS operation is shown at Figure 5. First, incoming data arrive into the network subsystem of the server, where they channel own is analysed. In the absence of errors, the data is transferred to internal structure of addressed channel. Then the data are sent to subscribers and to the logging system if it is necessary another network interface.

Property	Description
name	The unique name, addressing the channel
type	Channel permissions: rw, ro or ex ¹
descr	Description of the channel
time	The time of last update
value	The channel value
units	Units of the channel
is_log	Flag, meaning channel logging necessity

 Table 3: VCAS Data Channel Properties

To access to VCAS data simple text protocol based on TCP/IP stack and contains *key:value* pairs is used. During the server development several text protocols like JSON [11] or XML [12] was analysed. But, finally, pairs of text fields of key-value were chosen due to the parsing and encoding simplicity and, in turn, their fast operation from the one hand, and sufficiency of such approach to most of the automation tasks from the other.



Figure 5: VCAS and Logging service scheme.

At the present time VCAS application is implemented in C++ code using QT framework [13]. This framework was chosen for several reasons: author's wide experience with C++ and QT development, perfect documentation of the framework, fast application development and some others. The performance of VCAS implementation allows serving 5000-7000 channels updates per second on officeclass PC with reasonable client application subscribed to VCAS data.

VEPP-2000 GUI APPLICATION EXAMPLES

The software of VEPP-2000 automation system consists of more than 70 different applications. Most of them are GUI ones. Here are presented two different programs: one from the beam parameters measurement system – CCD data (see Figure 6) and other from control system of VEPP-2000 (see Figure 7).





b)

Figure 6: Electron a) and positron b) beams images from two CCDs of the beam parameter measurement system. Large betatron oscillations at the coupling resonance make the beam unusual ring-shaped (at top). Data from every 16 CCD are available to the control system via VCAS server.



Figure 7: GUI for VEPP-2000 collider inflectors.

rw – any application can read or write the channel value, ro – applications can read only the value (utility channels of VCAS itself), ex – exclusive, only the first updated the value application can write to the channel.



Figure 8: VEPP-2000 main control room. A. Senchenko at the middle ground.

CONCLUSION

VEPP-2000 automation system is based on hardware made at BINP (Novosibirsk) in the proven CAMAC and modern CAN-Bus field bus standards. The software system substantially corresponds to hardware system and has a three-layer architecture.

The separation of hardware peculiarities from GUI with so-called middleware allows to operator independently steering of different VEPP-2000 subsystems like ILU, B-3M and BEP, VEPP-2000 and beam channels.

New VCAS application was specially designed to provide to other automation system programs universal communication media – so called data channels.

Two experimental seasons in 2009-2012 have been done and two detectors SND and CMD-3 have collected about 30 pb^{-1} of integrated luminosity.

The control system of VEPP-2000 collider (see Figure 8) proved good flexibility and allows very fast development of a new, more powerful application.

- Yu.M.Shatunov et al., "Status of VEPP-2000 Collider", RuPAC'12,St.-Petersburg, Russia, September 2012; MOYCH01, http:// www.JACoW.org
- [2] "VEPP-2000 Project", http://vepp2k.inp.nsk.su
- [3] "CMD-3 Project", http://cmd.inp.nsk.su
- [4] "SND Project", http://wwwsnd.inp.nsk.su
- [5] "CAN In Automation", http://www.can-cia.org
- [6] "BINP Automation Division", http://www.inp.nsk.su/activity/automation/.
- [7] V.Kozak, at al., "Embedded device set for control systems. Implementation and applications", RuPAC-2006, Novosibirsk, Russia, 2006; THDO05, http://www.JACoW.org
- [8] D. Berkaev et al., "Beam Measuremet System for VEPP-2000", ICALEPS'09, Kobe, Japan, October 2009, TUP026, p. 143, http://www.JACoW.org
- [9] D. Berkaev et al., "Beam Measurement System of VEPP-2000 Injection Channels" ICALEPS'09, Kobe, Japan, October 2009, THP032, p. 155, http://www.JACoW.org
- [10] A. Senchenko, D. Berkaev, "VEPP-2000 Logging System", PCaPAC'12, Kolkata, India, December 2012, WEPD14, http://www.JACoW.org
- [11] JSON (Java Script Object Notation), http://www.json.org
- [12] XML (eXtensible Markup Language), http://w3.org/XML
- [13] QT framework, http://qt.digia.com/.

DESIGN OF THE DATA ACQUISITION SYSTEM FOR THE NUCLEAR PHYSICS EXPERIMENTS AT VECC

P. Dhara, A. Roy, P. Maity, P. Singhai, P. S. Roy, VECC, Kolkata, India

Abstract

The beam from K130 room temperature cyclotron is being extensively used for nuclear physics experiments for last three decades. The typical beam energy for the experiments is approximately 7-10MeV/nucleon for heavy ions and 8-20MeV/nucleon for light ions. The number of detectors used, may vary from one channel to few hundreds of detector channels. The proposed detector system for experiments with the superconducting cyclotron may have more than 1200 detector channels, and may be generating more than one million parameters per second. The VME (Versa Module Europa) and CAMAC (Computer Automated Measurement and Control) based data acquisition system (DAQ) is being used to cater the experimental needs. The current system has been designed based on various commercially available modules in NIM (Nuclear Instrumentation Module), CAMAC and VME form factor. This type of setup becomes very complicated to maintain for large number of detectors. Alternatively, the distributed DAQ system based on embedded technology is proposed. The traditional analog processing may be replaced by digital filters based FPGA (Field Programmable Gate Array) boards. This paper describes the design of current DAQ system and the status of the proposed scheme for distributed DAQ system with capability of handling heterogeneous detector systems.

INTRODUCTION

The data acquisition system at VECC (Variable Energy Cyclotron Centre) has been designed using standard commercially available hardware modules suitable for low energy nuclear physics experiments. The software has been indigenously developed at VECC for VME [1] and CAMAC [2] based system. Though the system is well proven for all the experimental needs; but the setup and management of this system for large number of detector channels becomes difficult job. The digital filter based FPGA boards may be very conveniently used instead of large number of analog modules. Different types of detector system may be used independently in the same experiment. This type of heterogeneous detector system needs a high precision time-stamping mechanism to synchronise the data.

CAMAC SYSTEM

The CAMAC is an old standard developed by ESONE (European Standards on Nuclear Electronics) committee in 1972. It is a 19" crate system with 25 slots to attach the modules. The slot 24 and 25 is used by the CAMAC crate controller. The CAMAC modules are independent functional units that can be accessed through crate controller. The crate controller is generally interfaced with the computer through a PCI card. The Ethernet based crate controller are also available, and popularly being used for slow control of different devices. The CAMAC solution consists of a Hytec 5331 CAMAC controller with PCI interface and a Hytec 1341 List Processor.

Each CAMAC module is addressed by a group of three signals N, A and F. The N is the slot number. Each module can host sixteen sub-units; which are selected by 4-bit signal A. Each sub-unit can perform 32 functions; which are selected by 5-bit signal F. There are separate 24bit *Read* and *Write* bus. The crate controller also issues six control signals, SI (strobe), S2 (strobe), Z (initialize), C (clear), B (busy) and I (inhibit). The CAMAC module responds with three signals; L (Look At Me, LAM), X (command status) and Q (operation status).

The typical CAMAC read cycle can be executed by the controller in 1µsec. In effect, a 24bit data read will result maximum data transfer rate of 3MB/sec. But the actual data transfer rate from the application software is determined by the software overhead. The single read takes 10µsec per word. The faster data rate is achieved by using List Processor module. This module can perform CAMAC operations as auxiliary controller and store the data in local buffer. The data is then read from the computer in block transfer. It takes average 2.14µsec for single read.

CAMAC Software

The CAMAC system is being used over decades at VECC. The CAMAC applications are built using the ESONE standard API. This makes the CAMAC system highly portable on different controller and operating systems. There are two versions of software:

- *T32:* It is the 32bit version of CAMAC developed using Windows SDK. The development life-cycle is over and only a limited support is offered for some old installations.
- *CAMACDAQ:* This is the currently supported version of the CAMAC system. It runs on both 32bit Linux and Windows machine. The software is the common framework for different DAQ hardware both VME and CAMAC system. A CAMAC crate is viewed as a single hardware module with multiple channels and inbuilt buffer. The list processor acquires the data independently from various modules inside the crate and stores them into the buffer. The software reads data from the list processor. So, the individual modules are isolated from the software perspective; thus, a wide variety of hardware modules are easily supported.

VME SYSTEM

The VME is a computer bus system, consists of address bus, data bus, control bus, utility bus and user I/O bus. The VME Bus can be efficiently utilised by a microprocessor based VME controller board. The VME is also a 19" crate based system with 21 slots and a common backplane for all the connectivity. The VME modules are available in different form factors 3U, 6U or 9U height (1U=1.75inch). The VME DAQ system uses 6U form factor VME64/VME64x crates. Following are the salient features of VME64X crate

- VME is asynchronous bus system, working in master/slave mode. Data transfer is done in handshaking mode. The bus can be used in both multiplexed and non-multiplexed mode.
- The backplane supports 32bit data bus and 32 bit address bus. The data and address bus may be multiplexed to provide 64 bit data and 64bit address. The VME system supports different addressing mode A16, A24, A32, A40 and A64. Data can also be transferred in D8, D16, D32 and D64 bits.
- The minimum read cycle time is 10ns, provided the slave could operate at the same speed as master. So, D32 mode may have maximum data transfer rate of 40MB/sec. The D64 mode may have maximum transfer rate of 80MB/sec.
- The VME Bus supports multiprocessing and four levels of interrupt. There must be one system controller at slot 1 to perform the bus arbitration. Multiple masters may request for bus, and the bus arbitration logic grants the bus access to masters.
- A typical VME master initiates bus access in Read/Write cycle, Read-modify-write cycle, block transfer cycle (BLT) or multiplexed block transfer (MBLT) cycle. The block transfer cycle may be in FIFO mode also, where an address cycle is followed by a number of data cycles.
- VME offers Chained Block Transfer mode (CBLT) of operation. In this mode, the block transfer call may span across a number of VME modules in the crate. The modules configured in this mode send acquired data one after another against a single block transfer call. The first module starts sending data, after it is purged out, the IACK signal is passed to next module for sending data.

VMEDAQ System

The VMEDAQ system has been implemented using CAEN V2718 Controller and SIS3100 controller. None of these controllers have processors for user application programming. They are driven from a connecting PC through PCI card and fibre optics cable. System supports CAEN V785 peak sensing ADC (analog to digital converter), V775 time to digital converter, V792 charge to digital converter, Mesytec MADC32 peak sensing ADC, MDI2 peak sensing ADC, SIS3820 scalar etc.

VMEDAQ Software

The VMEDAQ software is a multi-threaded C++ application developed on Linux. The software has been designed in layered architecture for easy maintenance and support. There are three layers *front-end*, *associator and back-end*.

- The *front-end* layer contains the graphical user interface, using QT-3 toolkit. It has the main control window, histogram windows and configuration management etc.
- The *associator* layer establishes the data and control communication between front-end and back-end. It stores common information shared by all layers.
- The *back-end* layer is also written in C++ with STL (standard template library), which reads data from the DAQ hardware. It also sorts the data and generates the global event. Afterward, the events are stored and processed.

The back-end layer implements three independent threads *read-out, event-generator* and *process,* which are working in producer-consumer configuration (see Fig. 1). The *read-out* thread reads the from VME or CAMAC hardware in block transfer mode (CBLT for multi-module VME setup). The data is stored in one of a dual buffer system. The buffers are shared by read-out and eventgenerator threads using two pairs of read/write semaphores. Same mechanism of data sharing exists between event-generator and the process threads. The event-generator thread generates the global event by organizing the fragments of event registered by different modules. Next, the process-thread stores the data into file and generates the one-dimensional and two-dimensional histograms as per function list.



Figure 1: Event processing in threads.

DAQ Setup

The DAQ software works in common dead time mode. The detector output signal first goes through the front end electronics (FEE) circuit, which does the signal conditioning suitable for particular type of acquisition. Then the signal is connected to an ADC channel. The VME785 peak sensing ADC has 32 input channels, which can digitize the input data within 5.7µsec for all channels. There is a gate generator logic which generates a logical GATE signal to validate an event. The ADC modules are enabled by the GATE signal and the input coming within (the GATE signal are digitized and stored as valid events. There may be any number of modules involved, depending upon the number of signal channels. All modules work with the same GATE signal. In this common dead time configuration, the system dead time τ is the maximum dead time of all individual components (see Fig. 2).



Figure 2: Common dead time configuration.

The DAQ software is configured by a single text based configuration file. There are four directives to be declared. The *System* directive defines the overall system parameters like transfer mode, event trigger etc. The *module* directive is used to declare hardware modules with parameters. The *function* directive declares the processing functions like histograms, add etc. The *gate* directive declares conditions. The software supports all the necessary functionality to manipulate histograms for physics research.

The Multi-crate DAQ Setup

The CAEN V2718 VME crate controllers can be connected in a daisy chain, for maximum eight crates per PCI card (see Fig. 3). The optical fibre interface works at a speed of 70MB/sec. The address space of each crate is isolated and modules in each crate can be read independently. The GATE signal is common to all the modules. A synchronization module has been designed in-house, to block all the successive GATE signals, until the current GATE signal is processed by all crates.



Figure 3: Multi-crate system.

The DAQ system is capable of handling 1.2Mparameters/sec in a dual-crate configuration.

FUTURE DAQ

The modern experiments are using a large number of detectors of various kinds into a single experiment. The module based systems are not convenient any more. The response time of different detectors may render the common dead mode configuration impractical. The following development works are under progress.

The Digital Processing

The detector signal processing with digital filter in FPGA is under development. A 16-channel DPP (digital pulse processing) broad has been planned for spectroscopic application.

The ASIC Board

The availability of ASIC (Application Specific Integrated Circuit) chip for pulse shaping & processing is also being explored. The traditional FEE chain may be replaced by the ASIC board.

The Heterogeneous DAQ with Timestamp

The heterogeneous DAQ setup uses different detectors arrays working together. They have different response time and resolution. There may be independent DAQ for each detector array. The events are marked with a global unique timestamp. The data may later merge together using the timestamps.

The work is going on to design the timestamp distribution module. The offline sorting and merging algorithm has already been developed.

CONCLUSION

The DAQ system has been intensively used by user communities at VECC in their experiments with K130 cyclotron beam. The system is ready to do experiments with larger detector arrays with K500 beam. There are some improvement proposed for the software to make is efficient in handling large number of histograms.

The DPP or ASIC based DAQ board will make the system more sophisticated for future use.

ACKNOWLEDGMENT

We thankfully acknowledge the help and contribution of all our colleagues of DAQ & Dev Section, VECC. Also we thankfully acknowledge the encouragement and feedback from Experimental Nuclear Physics Division of VECC.

- P. Dhara et al., "Multi-threaded object-oriented VME data acquisition system on Linux", DAE-BRNS Symp on Nuclear Physics, Vol. 46B, p. 530-531 (2003).
- [2] P. Dhara et al., "Integration of CAMAC system in VME based DAQ Software", DAE-BRNS Symp on Nuclear Physics, Vol 50, p. 454 (2005).

A FPGA BASED HIGH SPEED DATA ACQUISITION CARD

J.A. Gore, S.G. Kulkarni, K. Mahata, A. Shrivastava, V. Parkar, S.K. Pandit, A. Chatterjee, P.V. Bhagwat, S. Kailas, BARC, Mumbai, India

Abstract

A FPGA based, high speed, two channel, analog input card with a maximum input sampling rate of 1 Giga samples per second (Gsps) per channel has been designed and tested. The card has got an on-board cPCI interface but has been designed in a way that it can also work as a stand-alone system. The card can function as a platform for developing and evaluating different FPGA based hardware designs. Recently, the card has been used to develop a direct sampling Low Level RF (LLRF) controller for controlling the electromagnetic fields of a prototype heavy ion RFQ. It has also been tested for acquisition of data in nuclear physics experiments. Pulses from surface barrier and silicon strip detectors were acquired at an input sampling rate of 1 Gs/s employing ²⁴¹Am and Am-Pu sources. The design developed for this makes use of pre-triggering. This paper discusses the functionality, salient design issues and features of the card. Finally the hardware designs of above mentioned applications related to different areas of LLRF control and nuclear pulse acquisition are explained and the results obtained are presented.

INTRODUCTION

A data acquisition system consists of a ADC driver, a programmable data processing and storage unit, and also a data transfer stage. The advent of FPGA and subsequent progress in related technology has enabled the data processing stage to be easily adapted to different applications.

The FPGA based high speed data acquisition card can function as a platform for designing and evaluating different FPGA based hardware designs. Features and design issues relevant to data acquisition are briefly presented. The card has been used to develop a LLRF controller and also has been used to acquire data in nuclear physics experiments. These applications are briefly described.

FUNCTIONALITY & SALIENT DESIGN ISSUES

The card consists of the following features:

- FPGA:VIRTEX-5 XC5VFX100T
- cPCI interface: 32 bit, 33MHz master interface with DMA transfer capability using *PCI 9054* from PLX Technology.
- Clock: 156.25 MHz clock oscillator for SFP; 40 MHz clock oscillator for PCI; 32 MHz system clock; Clock distribution for ADC and DAC.
- Rocket IO interface: Two SFP connectors are provided for SFP modules

- Memory: 2GB DDR2 SDRAM; 256Mb flash Memory
- Analog Input: Two channels using 12 bit, 1.0 GSPS ADC: ADC12D1000 from National Semiconductor
- Configuration: JTAG Mode; Using On-board Configuration *PROM*.

Analog input data can be sampled and stored in the DDR2 memory. After the memory is filled, data transfer is done using the PCI interface in the DMA mode. The design thus involves sampling of incoming data in ADC (any one channel at a time) and acquiring the digital samples inside FPGA. The digitized data is then stored for further processing. Thus the path of data traversal is from ADC to FPGA to DDR2 to back to FPGA. Afterwards, the data can be processed at a relatively slow speed and then transferred to the PC via a cPCI interface.

The data acquisition hardware design consists of the following hardware entities:

- 1. Configuration and calibration of external clock synthesizer (AD9520 -3)
- 2. Configuration of ADC (ADC12D1000)
- 3. Data acquisition of digital samples within the FPGA.
- 4. Sample rearrangement and concatenation for writing to DDR2
- 5. Write operation to DDR2 memory via Memory Interface Generator (MIG) core [1].
- 6. Read operation from DDR2 memory via MIG core
- 7. Rearrangement of samples to obtain the final stream of digital samples.
- 8. Transferring data via cPCI interface.

The block diagram of the hardware design is shown in Figure 1.



Figure 1: Hardware design for data acquisition.



Figure 4: Direct sampling LLRF controller block diagram.

An output reference clock of 250 MHz is provided by the ADC to the FPGA for acquiring data. The data coming from the ADC is **DDR** (double data rate) on all the four buses. Hence the effective data rate is 250 X 2 = 500 **MHz** on each of the data bus (refer to Figure 2 and Figure 3 for more details).



Figure 2: Sample interleaving before write operation.



Figure 3: Sample interleaving before read operation.

Data acquisition inside FPGA is done at a speed of 250 MHz clock frequency. ADC provides the reference clock to the FPGA for each channel (I and Q) and one has to latch the samples in on every rising edge of the clock pulse. Initially DDR data is sent by the ADC which is converted to SDR data by using the IDDR primitive. IDDR is used to convert the input DDR data into SDR data. The output is provided on two separate parallel buses of the ADC for further processing. The hardware design described above is common to all the applications. This design makes available the sampled data for further processing. Data is sampled by the ADC, stored in the DDR2 memory. This memory is then read and the data is processed according to the relevant algorithms typical to the application being developed. The processed data is then transferred to the PC via the cPCI link. In the subsequent section algorithms related to two applications which have been implemented in the FPGA are briefly discussed.

APPLICATIONS

Direct Sampling LLRF Controller

A direct sampling LLRF controller hardware design has been developed for the 75 MHz heavy-ion RFQ (as shown in Figure 4). This design digitizes the pick-up of the resonant cavity, detects the In-phase (I) and Inquadrature(Q) components, generates the phase and amplitude error signals and finally generates a sine wave output signal which when converted to the analog domain using a DAC can be amplified. This signal then can drive the resonant cavity.

The design consists of the following hardware components:

- Direct digital converter(DDC)
- Phase and magnitude compare (PMC)
- Moving average loop filter(MA)
- Direct digital synthesizer(DDS)

The DDC entity performs the digital mix-down of the cavity pick-up input signal to baseband. It also generates the internal I/Q samples. The *phase_inc* value controls the internal DDS frequency and should be set to the resonant

FRCC02

frequency of the cavity. Assuming a resonant frequency of ~ 75 MHz, after the digital mix-down this will generate tones at DC (75-75) = 0 Hz and also at 75+75 = 150 MHz. At a sampling rate of 250 MHz then the unwanted secondary tone at 150 MHz will also have an alias at 100 MHz (250-150). To this end, the high-performance FIR low-pass filters have been designed to block any frequency above 25 MHz.

The PMC entity generates a known good phase and amplitude from the detected I and Q components. This is done with either the CORDIC algorithm, using the arcTan function or by Look-up-Table approach. The output of this entity is the phase and amplitude scale factor information.

The MA entity acts as a loop-filter. The purpose of this entity is to prevent abrupt changes to the *phase_shift* and *scale_fact* signals which may cause instability or oscillation in the loop. A simple moving-average filter has been implemented with 16-taps. The number of taps is currently fixed, but we can modify this if necessary.

The moving-average filter also has the secondary effect of being a low-pass filter.

The DDS entity generates the output sinusoid waveform. The core uses a 12-bit LUT and has a programmable *phase_inc* input that controls the frequency of the output sinusoid. The component also has additional *phase_shift* and *scale_fact* inputs that can control the relative phase

and amplitude of the output signal. The *phase_shift* input is a 32-bit unsigned value. The full 32-bits represent a shift of 360° . The *scale_fact* input is a 16-bit scale factor that scales the output waveform. It is a signed value in s1.14 format.

This hardware design has been implemented in the FPGA. Individual components have been successfully tested.

Pulse Acquisition

Pulses from surface barrier and silicon strip detectors were acquired at an input sampling rate of 1 Gs/s employing ²⁴¹Am and Am-Pu sources. The amplitude spectrum for the Am-Pu source is shown below.



Figure 5: Amplitude spectrum of Am-Pu source.

The design developed for this makes use of pretriggering. Three trigger modes have been implemented viz. normal trigger, post trigger and pre-trigger modes.

In the normal trigger mode, the data acquisition starts slightly after the occurrence of the rising edge of the trigger.

In the post trigger mode, the data acquisition begins after a programmable number of samples of occurrence of the rising edge of the trigger. The number of samples after which the acquisition should start is determined by the value stored in a register (trigger width).

In the pre trigger mode, the data acquisition starts a programmable number of samples before the rising edge of the trigger comes. The number of samples is determined by another register (pre-trigger width).

The trigger logic has been designed using a circular buffer architecture. A FIFO is used as a temporary buffer to store the samples for pre trigger mode and when the mode is enabled, it starts getting written into. Once the samples are equal to the pre-trigger width value, the FIFO gets into free read/write mode. That is the read and write start at the speed of the data input rate. This makes sure that the fixed number of samples is always in the buffer and they are updated at every clock edge. In post trigger mode, the data acquisition waits for the number of samples written in the respective register for the start for the start of acquisition. Once the samples surpass this value, data acquisition starts. In normal mode, all these are bypassed and the data acquisition occurs at the rising edge of the trigger.

The amplitude spectrum as shown in Figure 5 was generated after acquiring 2GB of pulse data. This data was stored in the on-board DDR2 memory and then transferred via a cPCI link to the PC.

CONCLUSION

The card has been successfully used to implement two applications (viz LLRF controller and nuclear pulse acquisition). For both these applications data was successfully sampled at an effective sampling rate of 1Gsps. Individual components of the direct sampling LLRF controller have been successfully tested. Integration and testing of controller is in progress. The Am-Pu amplitude spectrum has a resolution of 75 keV. Development of FPGA based filtering algorithms for nuclear pulse processing is on-going.

REFERENCES

[1] UG 086 memory Interface generator (MIG), www.xilinx.com

DEVELOPMENT AND PERFORMANCE ANALYSIS OF EPICS CHANNEL ACCESS SERVER ON FPGA BASED SOFT-CORE PROCESSOR

S. Sahoo[#], T. Bhattacharjee, S. Pal, VECC, Kolkata, India

Abstract

A soft core processor is a flexible hardware description language (HDL) model of a specific processor (CPU) that can be customized for a given application and synthesized for an FPGA as opposed to a hard core processor which is fixed in silicon. Combined with an on-board ethernet port, the technology incorporates integrating the IOC and digital control hardware within a single FPGA thus reducing the overall hardware complexities of field devices. In this paper, the technical details of porting EPICS Channel Access Server on MicroBlaze soft-core processor are explained. The EPICS performance on the MicroBlaze processor is analyzed. For this, the CPU load and server processing time for different numbers of Process Variables (PVs) have been studied for this platform. On the basis of the analysis, critical parameters of EPICS on this embedded platform have been derived and a few modifications in the channel access protocol are proposed for MicroBlaze soft-core processor.

INTRODUCTION

Experimental Physics and Industrial Control System (EPICS) has been used in many accelerator laboratories to design the control system of accelerator under a unified architecture for better reliability, integrity and security of the overall control system of an accelerator.

The Input Output Controller (IOC) is the heart of an EPICS distributed control system and many such IOCs can be distributed over the control network. These IOCs are generally loaded in Personal Computers (PCs) running windows or linux operating system and placed near the field devices. Nowadays, the PC-based EPICS IOC is used in many laboratories, but it has many maintainability issues.

A soft core processor is a flexible CPU architecture that is configured in the FPGA as opposed to a hard core processor which is fixed in silicon. Combined with an on-board Ethernet port, the technology incorporates the IOC and digital control hardware within a single FPGA [1]. Nios II (by Altera), MicroBlaze (by Xilinx), OpenRISC 1200 (by OpenCores.org), LatticeMicro32 (by Lattice Semiconductors) and Cortex-M1 (by ARM Limited) are some of the soft-core processors that are targeted mainly for FPGA implementation. On the basis of various features like portability of Linux operating system, availability of FPU (Floating Point Unit) and MMU (Memory management Unit) and number of logic elements occupied, it has been seen that MicroBlaze

ssahoo@vecc.gov.in

processor core is the most optimized target soft-core processor for our application. The use of MicroBlaze processor and the uC-linux operating system has been very successful to date [2]. Also placing the processor and the user-defined hardware on the same device does offer many benefits and better reflects the state-of-the-art in systems-on-chip [3]. Our 62.5 MHz MicroBlaze provides a great deal of processing power, the Spartan-3A FPGA provides the capability to implement a significant amount of user-logic, and the uC-linux operating system provides a good platform for software development and debugging.

The scope of this paper includes porting EPICS on FPGA based MicroBlaze soft-core processor and analyze the EPICS record processing and channel access performance on it. To achieve this, broad steps which are necessary to be performed are described in the subsequent sections.

BUILDING MICROBLAZE PROCESSOR

We have used mainly Xilinx Platform studio for building the MicroBlaze system [4] (i.e. the processor along with the peripherals and interconnects). Figure1 is the basic flow diagram representing the steps involved in building a MicroBlaze system that has been customized as per requirements in our case.



Figure 1: Flow Diagram for Building MicroBlaze System.

Je

C

PORTING LINUX ON MICROBLAZE

The uC-linux 2.6 has been ported [5] on the MicroBlaze system built for our purpose. The original uC-linux was a derivative of Linux 2.0 kernel intended for microprocessor based systems without a Memory Management Unit (MMU). Though the present version of uC-linux 2.6 also supports MMUs, however, in our system we have implemented it without an MMU. Figure 2 describes broadly the steps followed by us for the porting of linux kernel on the MicroBlaze soft-core processor.

PORTING EPICS ON MICROBLAZE

Portable Channel Access Server was cross compiled manually for MicroBlaze architecture. Successful testing was performed on Xilinx Spartan-3A DSP 1800 evaluation board. It was found that following are the steps need to be followed for porting channel access server on MicroBlaze.

Building GNU Cross Compiler Tool-chain for MicroBlaze-uC-linux Platform

Xilinx MicroBlaze GNU tools source package was built which includes binutils-2.16, gcc-4.1.2, gdb-6.5 and newlib-1.14.0.

Building the Necessary Library Packages for MicroBlaze-uC-linux Platform

The libCom, libca, libcas, libgdd and librt libraries were needed to be built in addition to prebuilt libraries in the compiler tool-chain.

Compiling the Portable Channel Access Sourcecodes Using MicroBlaze-uC-linux Tool-chain

Portable Channel Access source codes provided in makeBaseApp are compiled using a suitable makefile, created manually.

Building the uC-linux Kernel Image Along with Server Application

The server application created above is included in the kernel image of uC-linux 2.6.

Downloading the Kernel Image into FPGA

The resultant kernel image is downloaded to the configuration flash memory of the FPGA board.

Configuration	Pipelining	I-Cache	D-Cache
Conf. 1	3 Stage	2 kB	2 kB
Conf. 2	3 Stage	8 kB	8 kB
Conf. 3	5 Stage	2 kB	2 kB
Conf. 4	5 Stage	8 kB	8 kB



Figure 2: Flow Diagram for porting uC-linux.

PERFORMANCE ANALYSIS OF EPICS

We have calculated mainly two parameters while doing the performance analysis of Channel Access Server on MicroBlaze processor [6].

- 1. Server CPU Load: This is percentage utilization of CPU resource at the server end while servicing to the client requests.
- 2. Server Processing Time: Time required by the server to accept the client requests, process them and publish them back to the client.

The Channel Access Protocol [7] is consisting broadly of four steps viz. Channel Connect, Put, Get and free. The individual Server CPU load and processing time for each of these steps are measured for MicroBlaze and compared to ARM9 processor which has a similar architecture of MicroBlaze but having a fixed-core that lacks the facility of changing the processor architecture and peripherals unlike soft-core processors. For each processor type, the maximum number of PVs in the database is calculated from the Server Processing Time and safe limit of number of PVs is estimated from the CPU load parameter. Since MicroBlaze processor is fully customizable soft-core target, so four different configurations are built and tested in order to optimize the results. The four configurations are listed in Table 1.

Figure 3 and 4 shows the graph of CPU Load (in %) to number of PV requests at the server machine for ARM9 and MicroBlaze processor (Configuration 4) respectively. For the other configurations of MicroBlaze processor we have got similar results.



Figure 3: CPU load in ARM9.



Figure 4: CPU load in MicroBlaze (Conf. 4).

Figure 5 and 6 shows the server processing time per PV for channel access get and put on different platforms. As expected with higher cache memory and greater number of stages of pipelining the performance of the MicroBlaze processor is enhanced. But the ARM9 processor is almost twice as fast as MicroBlaze (8 kB cache and 5 stage pipelining). The probable reasons for this are as follows:

- 1. No memory management unit in MicroBlaze processor.
- 2. Fork is not supported.
- 3. Higher processor speed (200 MHz for ARM9 processor and 62.5 MHz for MicroBlaze Processor).



Figure 5: Server processing time per PV for CA Put.



Figure 6: Server processing time per PV for CA Get.

From the server processing time per PV graph, we have calculated the maximum number of PVs that are allowed in the server database. Results are calculated for three different scan periods of PVs viz. 0.1, 1 and 10 seconds and listed in Table 2. These values are estimated considering only Channel Access Get. Also we have estimated a safe limit of PVs from Figure 3 and 4 which determines the number of PV requests from client up to which the server can process without overloading itself.

Table 2: Results for Maximum Number of PVs at Different Scan Rate and Safe ILimit of PVs

Config.	Max. PV Limit(0.1s)	Max. PV Limit (1s)	Max. PV Limit (10s)	Safe PV Limit
Conf. 1	80	1200	12000	2500
Conf. 2	80	2000	20000	3000
Conf. 3	80	1300	13000	2500
Conf. 4	80	2200	22000	4000
ARM9	400	4000	40000	5000

SOME ABNORMAL BEHAVIOURS IN LOWER RANGE OF PVS

Figure 7, 8 and 9 shows the graph between processing time per PV vs. No. of PVs in Channel Connect, Put and Get respectively. The graphs show the detailed trend of curve in lower range of PVs (i.e. between 1 to 1000). In all the three graphs we see that a sudden rise in server processing time per PV takes place at different number of PVs for MicroBlaze processors. Only configuration 1 and 4 are shown here but similar nature is obtained for all the MicroBlaze configurations. For Channel Connect there is a sharp rise in server processing time per PV when the number of PV requests crosses 430. Similar peaks can be seen at 60 and 100 for Channel Put and Get respectively. It can be noted that for the ARM9 processor such peaks are not seen.



Figure 7: CA Connect in MicroBlaze.



Figure 8: CA Put in MicroBlaze.



Figure 9: CA Get in MicroBlaze.

After analyzing the reason behind these peaks, it has been found that repetitive retransmission of TCP segments of the TCP/IP stack is taking place while communication with MicroBlaze processor. This is making per PV transaction time nondeterministic in a congestion free network channel, hence the soft real time response of EPICS channel access is being violated which is normally achieved in a network with 30% load, possibly with minimum or no collision domain in the network design by intelligent network switches.

0	💿 eth0: Capturing - Wireshark					
dit '	View Go C	Capture Analyze	Statistics Telepho	ny Too	ls Help	
ä	M		xce	9	🔶 🍦 🍹 🛓 🔳 🖪 🔍	
tcr	0				Expression Clear Apply	
Т	ime	Source	Destination	Protoco	al Info	
7 0 8 0 9 0 10 0 12 0 13 0 15 0 16 0 17 0 18 0 19 0 20 0 21 0	0.013525 .014778 .014798 .014866 .028579 .028625 .216525 .216525 .251668 .256489 .256489 .256267 .322743 .323204	192.168.0.30 192.168.0.10 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.10 192.168.0.30 192.168.0.10 19	192.168.0.10 192.168.0.30 192.168.0.10 192.168.0.10 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.30	TCP TCP TCP TCP TCP TCP TCP TCP TCP TCP	$\begin{array}{llllllllllllllllllllllllllllllllllll$	
22 0 23 0 24 0 25 0 26 0 27 0 28 0 29 0 30 0 31 0	. 323211 . 324509 . 528529 . 529906 . 529942 . 531404 . 616451 . 616710 . 616725 . 617872	192.168.0.30 192.168.0.10 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.10 192.168.0.10 192.168.0.30 192.168.0.30 192.168.0.30	192.168.0.10 192.168.0.10 192.168.0.10 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.30 192.168.0.10 192.168.0.10 192.168.0.30	TCP TCP TCP TCP TCP TCP TCP TCP TCP	32909 - c-1 LEH, ACK seq=4993 ACK TTCP Dup ACK 20#17_ca1 > 32909 ACK (TCP Netransmission) 32909 > c-1 K (TCP Netransmission) 32909 > c-1 K Ca1 - 32909 ACK Seq=785 ACK=4293 (TCP Netransmission) 32909 > c-1 C Ca1 - 32909 ACK Seq=785 ACK=6425 Ca1 - 32909 ACK Seq=6425 ACK 32909 > c-1 PSH, ACK Seq=6425 ACK 32909 > c-1 PSH, ACK Seq=6425 ACK 32909 > c-1 PSH, ACK Seq=6435 ACK 32909 > c-1 PSH, ACK Seq=6435 ACK 32909 > c-1 PSH, ACK Seq=6439 ACK=6833 ACK Seq=6409 ACK Seq=6409 ACK=6833 ACK Seq=6409 ACK Seq=6400 ACK Seq	

Figure 10: Wireshark screenshot for TCP communication with MicroBlaze Server.

It has been seen that in MicroBlaze Processor due to limited memory resources and lack of memory management unit and a known bug in soft irq process, a few data packets are often lost which triggers the retransmission in the client socket. This phenomenon is totally unreliable as it depends on the non-deterministic transmission of packets delay with adaptive retransmission algorithm. In the case of less number of PV requests there is no retransmission taking place because only one TCP data packet (1448 Bytes) is enough and as soon as multiple TCP packets are sent, retransmission is happening due to the inability of multiple data packet handling of the MicroBlaze processor. The repetitive retransmission enables the retransmission algorithm to increase the retransmission timeout. Moreover, in some cases Nagles algorithm comes into picture which tries to increase the packet size in order to compensate the time loss. This worsens the situation because the server (MicroBlaze) is not capable of handling bigger data packets which again causes retransmission. This phenomenon deteriorates the overall network performance.

PROPOSED IMPROVEMENT

Looking into the above scenario of retransmission a solution can be proposed which will be effective for any embedded system lacking the hardware resource which causes non-deterministic network communication with EPICS channel access protocol.



Figure 11: General Message Header in Channel Access.

In the EPICS channel access a technique, the concept of message buffering is introduced which increases the efficiency of the channel by grouping individual messages in a single message segment. In congestion free network where repetitive retransmission is a common phenomenon due to the inefficiency of handling multiple TCP packets by the server socket, this message buffering concept is a bottleneck. It can be straight way proposed for the MicroBlaze processor that the optimized channel access performance would be with the message buffer size equal to 1448 with BSD socket (derived from wireshark results). Nonetheless, considering this retransmission phenomena in other embedded systems lacking resources, we can think of an adaptive message buffering algorithm in channel access to change the payload size dynamically during the initial connection with server or on notification from the server which can be described as follows:

- Start with a fixed payload size (say Payload Original) for channel access message buffer.
- If retransmissions are detected in the media, reduce the payload size to half of its previous value. i.e. Payload _{New} = Payload _{Original} / 2
- Detect for any retransmission in the media. If retransmission is detected reduce the payload size further by half of its previous value.
- Thus after N iterations when there is no retransmission in the media, the final payload size becomes

$$\begin{array}{l} Payload \ _{Final} = Payload \ _{Original} / \ 2^{N} \\ N = log_{2} \ \ \underline{\begin{array}{c} Payload \ Original \\ Payload \ Final \end{array}} \end{array}$$

Payload_{Final} is the optimum payload size for message buffer in channel access for no retransmission.

This algorithm has to be embedded with the general message buffering available in channel access as the same client may connect a IOC server where the performance is better with the general message buffer and hence the additional logic in the embedded server should be defined to notify the client that it needs the proposed adaptive message buffering algorithm to establish connection.

CONCLUSION

In this project we have successfully ported EPICS channel access server on MicroBlaze soft-core processor. To understand the suitability of embedded systems in real-time environment, EPICS channel access and record processing performances have been analyzed for MicroBlaze platform and the results are compared with standard ARM9. The CPU load and server processing time for different numbers of client requests have been studied. Maximum number of PV limit in a server database is calculated for both Microblaze and ARM9 server machine. The performance in MicroBlaze soft-core processor can be considerably improved by suitably tuning the processor architecture (like size of cache memory and number of stages of pipelining). Another improvement being proposed for embedded system is the change in message buffer size of EPICS channel access which quite frequently causes retransmission and thus degrading the real-time performance of the system.

- [1] D. Curry, A. Hofler, H. Dong, T. Allison, C. Hovater, K. Mahoney,"Implementation of an EPICS IOC on an Embedded Soft Core Processor Using Field Programmable Gate Arrays", Proceedings of 10th ICALEPCS, Geneva, 2005.
- [2] J.G. Tong, Ian D.L. Anderson, Md. A. S. Khalid, "Soft-Core Processors for Embedded Systems", Proceedings of 18th International Conference on Microelectronics (ICM), 2006
- R. H. Klenke, "Experiences Using the Xilinx MicroBlaze [3] Soft-core Processor and uC-linux in Computer Engineering Capstone Senior Design Projects", Proceedings of IEEE International Conf. on Microelectronic Systems Education, 2007
- [4] R. Jesman , F. M. Vallina and J. Saniie, "MicroBlaze Tutorial for Creating a Simple Embedded System and Adding Custom Peripherals Using Xilinx EDK Software Tools", 2006.
- [5] J. Wu, I. Syed and J. Williams, "Creating a simple uC-linux readv MicroBlaze Design version 1 05a" http://itee.uq.edu.au/~wu/downloads/uClinux ready Microblaze design.pdf
- [6] J. Odagiri, A. Akiyama, N. Yamamoto and T. Katoh, "Performance Evaluation of EPICS on PowerPC" Proceedings of ICALEPCS, Beijing, 1997.
- [7] K. Žagar, "Channel Access Protocol Specification", http:// epics.cosylab.com/cosyjava/JCA-Common/ Documentation/ CAproto.html, 2003.
DIGITAL PULSE PROCESSING TECHNIQUES FOR HIGH RESOLUTION AMPLITUDE MEASUREMENT OF RADIATION DETECTOR

P. Singhai, A. Roy[#], P. Dhara, VECC, Kolkata, India S. Chatterjee, Heritage Institute of Technology, Kolkata, India

Abstract

The digital pulse processing techniques for high resolution amplitude measurement of radiation detector pulse is an effective replacement of expensive and bulky analog processing as the digital domain offers higher channel density and at the same time it is cheaper. We have demonstrated a prototype digital setup with highspeed sampling ADC with sampling frequency of 80-125 MHz followed by series of IIR filters for pulse shaping in a trigger-less acquisition mode. The IIR filters, peak detection algorithm and the data write-out logic was written on VHDL and implemented on FPGA. We used CAMAC as the read out platform. In conjunction with the full hardware implementation we also used a mixedplatform with VME digitizer card with raw-sample read out using C code. The rationale behind this mixed platform is to test out various filter algorithms quickly on C and also to benchmark the performance of the chip level ADCs against the standard commercial digitizer in terms of noise or resolution. The paper describes implementation of both the methods with performance obtained in both the methods.

INTRODUCTION

In digital pulse processing technique, detector or preamplifier output signals are directly digitized and processed to extract quantities of interest. Our concern is to detect energy of the particle, which is proportional to amplitude of preamplifier output pulse. Traditional analog electronics is here replaced by programmable digital filters. Most of the typical processing features, e.g., polezero cancellation, baseline restoration, and shaping are digitally implemented and optimized. Costly peak sensing ADC is replaced by low cost fast ADC. It has been suggested that, typical fast AD converter should have resolution of 10-12 effective number of bits and sampling frequency 50-200 MSPS, for accurate energy measurement [1, 2]. Two digital pulse processing techniques software based and hardware based, using such ADCs, are proposed and demonstrated for high resolution amplitude measurement of radiation detector pulse.

SOFTWARE BASED PULSE PROCESSING

In first technique of digital pulse processing, digital filters and peak detection algorithm was designed and implemented at software level. It is easy to start with such flexible system as various pulse processing features like pulse shape, shaping time, pole-zero compensation can be easily modified and tested by varying the parameters for various detector systems and various ADCs. Figure 1

#amitav@vecc.gov.in

shows the block diagram of software based pulse processing system.



Figure 1: Software based pulse processing system.

Data is sampled by a 14 bit 100 MSPS VME ADC. The data stream is continuously written in a circular memory buffer and read via VME. The major problem of software based system is that the amount of data to readout is extremely high. It is not possible to sustain a continuous acquisition and transfer all the data to the computers. Therefore the data is acquired through block transfer mode at the cost of loss of many events. Fragmented data is discarded at the beginning and end of the block. All pulse processing features are designed at software level using 'C' programming. Codes are written to implement digital filters for CR-RC shaping of pulse, along with pole-zero compensation, whose difference equation is represented by equation 1.

$$y(n) = b_1 \times y(n-1) + b_2 \times y(n-2) + a_0 \times x(n) + a_1 \times x(n-1) + a_2 \times x(n-2)$$
(1)

Shaping time of filter is decided by coefficients b_1 , b_2 and pole-zero compensation depends on a_0 , a_1 and a_2 for a certain sampling frequency.

Figure 2(a) shows two overlapped preamplifier pulses at the input of digital CR-RC filter and 2(b) shows clearly resolved pulses with better SNR at the output of filter. A trigger less peak detection logic is developed to measure amplitude of filtered pulse. The peak detector logic is one

279

of the most critical parts of this program, since in involves eliminating the jitter noise which present in almost every natural system. After detecting the peak amplitude of filtered pulse, it is plotted on histogram to analyse energy information of particle.



Figure 2: Input and output of digital CR-RC filter.

HARDWARE BASED PULSE PROCESSING USING FPGA

In this scheme all of the pulse processing blocks used in software based system is translated into a hardware system for real time implementation. This system is having advantage of zero dead time and no loss of events.

A continuous running, trigger-less, digital system is optimally designed and implemented at hardware level to perform typical processing tasks i.e. automatic pole-zero cancellation, CR-RC shaping and peak detection. Spectroscopy performances are achieved using low-cost and relatively low-resolution (12-bit) AD converters at high sampling frequency (80-100MSPS). It is not possible to run filter loop within 10-12 ns with the digital signal processors, therefore we choose FPGA to implement the design in the hardware level itself. For current implementation, XC4VLX25 Virtex 4 FPGA [3] has been chosen as target device. Figure 3 shows block diagram of this system.

Preamplifier pulse is sampled using fast digitizer. This sampled data is then transferred to FPGA for digital pulse processing. VHDL codes have been written to implement automatic detection of fall time, digital IIR filter for polezero compensation and CR-RC shaping (represented by Equation 1) and trigger less peak detection. These codes are first simulated successfully and then implemented on FPGA. Output of peak detection is read by CAMAC based data acquisition system and plotted on histogram for further energy analysis.



Figure 3: Hardware based pulse processing system.

EXPERIMENTAL RESULTS AND ANALYSIS

Performance of the digital systems is tested with, first data from precision pulsar and then from actual radioactive sources using semiconductor detectors. In order to measure the resolution of proposed digital systems, HPGe detector (resolution nearly equal to 2 KeV) is used with a source of ⁶⁰Co. Performances of both the systems are compared with traditional analog spectroscopy system.

Figure 4(a) shows spectrum of 60 Co source, obtained from traditional analog spectroscopy amplifier, using CAMAC based 13 bit peak sensing ADC (Ortec 413). Resolution of this system at spectral line 1332 KeV is **0.192%**. Figures 4(b) and 4(c) shows the spectrum obtained from software and hardware based digital pulse processing system respectively. A linear amplifier with no pulse processing feature is used with both the systems to enhance the preamplifier output signal. Resolution obtained with VME ADC based digital system was **0.532%** and that with FPGA based system was **0.578%**.



Figure 4: Spectrum of ⁶⁰Co.

FUTURE SCOPE

It is planned to develop the whole digital pulse processing system including anti aliasing filter, fast digitizer and various digital filters implemented on FPGA on a single board, in the near future. This will help to remove the distortion in the clock and other signal because of wires used to interface various devices. Also using proper design and shielding, noise contributed by ground can be reduced to result better resolution. The board is planned to design for multi-channel system by optimizing the resources available in FPGA.

ACKNOWLEDGMENT

'C' program codes for VME ADC based digital system were developed with the help of Pranab Singha Roy, VECC, Kolkata and T. Sivananda Reddy, RRCAT, Indore, India.

REFERENCES

- L. Bardelli and G. Poggi. Digital-sampling systems in highresolution and wide dynamic-range energy measurements: Comparison with peak sensing adcs. Nuclear Instruments Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 560(2):517–523, 2006.
- [2] L. Bardelli and G. Poggi. Digital-sampling systems in highresolution and wide dynamic-range energy measurements: Finite time window, baseline effects, and experimental tests. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 560(2):524–538, 2006
- [3] Users Guide, TSW1200EVM: High-Speed LVDS Deserializer and Analysis System. TI Incorporated. http://www.ti.com/lit/ug/slau212a/slau212a.pdf

INTRODUCING THE !CHAOS CONTROL SYSTEM FRAMEWORK

L. Catani, F. Zani, INFN-Roma Tor Vergata, Roma, Italy C. Bisegni, D. Di Giovenale, G. Di Pirro, L. Foggetta, M. Mara, G. Mazzitelli, A. Stecchi INFN-LNF, Frascati, Italy

Abstract

The analysis of most recent developments on highperformance software technologies suggests that new a design of distributed control systems (DCS) for particle accelerators and large experimental apparatuses can profit from solutions borrowed from cutting-edge Internet services. To fully profit from this new technologies the DCS model should be reconsidered, thus leading to the definition of a new paradigm. In this paper we present the conceptual design of a new control system for a particle accelerator and associated machine data acquisition system (DAQ), based on a synergic combination of a non-relational key/value database (KVDB) and network distributed object caching (DOC). The use of these technologies, to implement continuous data archiving and data distribution between components respectively, brought about the definition of a new control system concept offering a number of interesting features such as a high level of abstraction of services and components and their integration in a framework that can be seen as a comprehensive control services provider for GUI applications, front-end controllers, measurement and feedback procedures etc. The work is under development by a collaboration of INFN-LNF and INFN-Roma Tor Vergata with growing contributions from other academic and industrial partners.

THE !CHAOS FRAMEWORK

A typical example of software technology emerging from developments of Internet services is the class of nonrelational databases known as key/value database. They offer an alternative to relational databases (RDMS) that is having a growing success and interest among developers of web services due to of their high throughput, scalability and flexibility. Another example are the distributed memory object caching systems. They provide in-memory key/value store for small chunks of frequently requested sets of information in order to both respond faster to requests and to distribute the load of the main server to a scalable cluster of cache servers.

These two software technologies represent the core components in the design of this new control system we named !CHAOS [1] (i.e. "not" CHAOS, where CHAOS acronym stands for Control system based on Highly Abstract Open Structure) [2, 3].

In particular, the KVDB is used by DAQ for managing what we call *history data*, while the DOC implements the service for distributing *live data* from the front-end controllers to clients, thus replacing the client/server communication.

Datasets that need to be updated are identically pushed,



Figure 1: The "control service provider" model and the !CHAOS framework abstraction boundaries.

by front-end controllers, to both DOC and KVDB servers by issuing *set* commands. It means that data collection mechanism for DAQ is inherently included in the !CHAOS communication framework because both *live* and *history* data are pushed by the data source (the front-end controllers) to similarly distributed caching and storage systems. Moreover, since both DOC and KVDB use key/value data storage, formatting and serialization of datasets can be done once for both.

It is important noting that both the client applications and the front-end controllers are simple clients of the distributed object caching and DAQ. In particular, provided the DOC has an object container for each dataset of the DCS, defined by its unique key, a GUI client simply sends to the !CHAOS DOC service a *get* request for the object identified by that particular key, i.e. the dataset describing the associated device. On the other side the controller responsible for that device updates the correspondent dataset, according to the push rate defined for it, by issuing *set* commands to the DOC.

Data refresh rates, as well as other meta-data such as global parameters, CU configuration, commands and data syntax and semantic etc. will be managed by the meta-data server (MDS). The Meta Data Server will be also the central authority for !CHAOS components. It will keep track, for instance, of the Control Units instantiations. As supervisor of their initialization, it will manage, at start up, the registration of CUs' services and datasets providing them with a systemwide unique reference to be properly addressed by client applications.

It is worth mentioning that in !CHAOS the DOC layer is not operated as an object caching in a strict sense since the distributed memory is not populated after clients' requests. Instead, each dataset is by default stored in the DOC and continuously updated by the front-end controllers. Nevertheless the datasets transfer from front-end to clients can still profit from the high performance of the distributed caching systems that, in addition, prevents front-end controllers from overload originated by multiple clients' requests [4].

Another benefit of the !CHAOS design is that the frontend controllers don't need to run servers to provide data to clients since they themselves are clients of the data distribution and storage services. That improves their robustness and portability.

A fundamental property of both DOCs and KVDBs is their intrinsic scalability that allows distributing a single service over several computers. Moreover, dynamical keys re-distribution allows automatic failover by redirecting to other servers the load of a failed one.

The data-pushing strategy allows to further extend the abstraction boundary at the front-end. The Controller's functionalities can be simplified and standardized by introducing the *Control Unit* (see next paragraph), a manager and a supervisor of the software modules implementing the device's specific control procedures.

In addition, abstraction of services will be implemented throughout (Fig. 1). Access to live or history data, for instance, will be provided by !CHAOS APIs wrapping the service specific APIs in such a way that client's access to services, and even internal communications, will not be affected by the modification or replacement of any core component.

Serializations of datasets and of information passed between components (e.g. command's parameters, result of queries, etc.) will further improve the abstraction of services by using a binary string as the common format for the methods' payload [5].

The commands dispatching and the events notification services complement the communication and interconnection between !CHAOS components. A cross-language RPC-like software (i.e. *msgpack* [6]), included in the !CHAOS libraries, will be used by client applications for sending commands to front-end controllers.

In conclusion !CHAOS is a scalable control system framework providing, at a high level of abstraction, all the services needed for communication, data archiving, timing, etc.; GUI applications and front-end controllers access the framework services and expand its functionalities.

CONTROL UNITS

Figure 2 shows the logical structure of the software running in a front-end controller. The Control Unit (CU), the



Figure 2: !CHAOS components for the front-end controllers.

CU Toolkit and the included Common Toolkit are components of the !CHAOS framework while the device management modules (DMM) are software modules that complement the !CHAOS framework functionalities by providing the interface to the device. The development of these components is expected either as a contribution, or as a responsibility, of the device experts. They will simply focus their work on the development of control loops, commands execution etc. while all the other operations will be delegated to the !CHAOS Framework.

One or more instances of CU can run simultaneously, although completely independent, in a front-end controller. Each CU should be dedicated to a particular device or a family of devices and specialized for that particular component by means of appropriated device management modules. The latter is a set of routines implementing the device's specific functionalities grouped into five general modules: *initialization*, *de-initialization*, *stop*, *control loop* and *commands execution*.

When a command issued by a client application is received by the CU, the *command execution* module will be invoked for executing it.

The command is delivered to the CU Toolkit running the *command server* for all the CUs managed by that particular controller. The CU Toolkit, by analyzing command's *domain*, identifies the CU to which it is targeted and appends it to the correspondent command's queue.

If the application issuing the command requires a direct read-back from the CU this can be provided by returning the command's results to the client's call. Alternatively, since also the UI Toolkit will host a *msgpack* server, the client application could be notified, yet asynchronously, on the results of the command's execution by a message delivered from the CU. The latter, actually, is the preferred solution. Upon receipt of the command the CU verifies that the method *alias* indicated in the command's header is available and it can be executed (i.e. there is no other blocking method pending) and then launches the *commands execution* module. The rest of the serialized information containing the instructions for the action to be taken is passed as-it-is to the command's execution module.

The implementation of separated threads assures that requested periodicity of dataset refreshing is preserved even during any commands' execution. In addition the serialization of the command's descriptor (i.e. the command's header, method name, parameters etc.) passed to the execution module allows a common interface for all the methods to be implemented.

During the command execution, if needed, the refresh rate of the device can be set, at least temporary, to an higher value providing the operator and the history data archiving system with a more detailed description of the attribute evolution. Modification of parameters like the data refresh rate are superintend by the Meta Data Server. All components concerned with this change will receive notification by means of the events notification system.

HST ENGINE

In !CHAOS the DAQ, i.e. the machine data acquisition system, is provided by the service we call History (HST) Engine. A distributed file system is used to store data produced by machine operations while a KVDB manages the indexes structure; candidates are Hadoop [7] and MongoDB [8] respectively. The functionalities of !CHAOS HST Engine are allocated to three dedicated components, or nodes, namely the CQL Proxy (where CQL stands for CHAOS Query Language), the Indexer and the Maintainer.

Figure 3 shows the data flow and the role of the before before mentioned nodes in data writing (red) and reading/querying operations. Grey lines are used to indicate internal actions and data flow.

A CU starts the writing process by sending a dataset to the CQL Proxy indicated, from the MDS, as its primary HST server (1). Upon receipt of the package, the proxy interprets the CQL command and writes the data into the file system (2). Hadoop automatically replicates the data in the other servers of the cluster (grey lines). Then the CQL Proxy informs the pool of Indexer nodes about the new entry (3) and the first available Indexer appends the task to its queue. When processing the entry, the Indexer first reads the packet (i.e. the dataset) from the first available Hadoop node (4), analyzes it and, according to the indexing rules, updates the corresponding indexes on the MongoDB (5). The default indexing strategy will be by chronological order, i.e. based on the timestamp and bunch/packet number within timestamp intervals.

Queries to HST are triggered from client applications by sending a CQL command (1) to the proxy with the highest priority in its list. The proxy node decodes the request and passes it to the first available Indexer (2) that in turn, by querying the Indexes DB, receives the positions of data



Figure 3: !CHAOS History Engine and its components.

packets (3) satisfying the query's conditions (e.g. all data packets within a certain time interval) and sends them to the CQL Proxy (4). The packages are then collected (5) from various FS Servers and sent (6) to the client.

It's worth mentioning that since responses to queries are asynchronous and tasks can be distributed among different nodes, data packets resulting from a query can be provided to the client application also by CQL Proxies different from the one that originally received the request.

CONCLUSION

The design of the !CHAOS Framework is approaching its final stage. The whole architecture has been completed in details and prototypes of the main services are already under test. Other components and concepts not presented in this paper, namely Execution Unit, I/O Unit, Notification Service etc., have also been developed. Porting of !CHAOS libraries to various platforms, including ARM based boards, has been successfully completed. Moreover some !CHAOS components have been already adopted by the DAFNE and SPARC control systems at INFN-LNF and successfully operated since several months.

REFERENCES

- [1] http://chaos.infn.it
- [2] G. Mazzitelli *et.al.*, "High Performance Web Applications for Particle Accelerator Control Systems", Proceedings of IPAC2011, San Sebastian, Spain, pp.2322-2324, http://www.JACoW.org
- [3] L. Catani *et.al.*, "Exploring a New Paradigm for Accelerators and Large Experimental Apparatus Control Systems", Proceedings of ICALEPCS2011, Grenoble, France, http://www.JACoW.org
- [4] http://memcached.org
- [5] http://bsonspec.org
- [6] http://msgpack.org
- [7] http://hadoop.apache.org
- [8] http://www.mongodb.org

PROCESS CONTROL FOR PARALLEL RUN OF TWO HELIUM LIQUEFIERS AT VEC CENTRE, KOLKATA

Sandip Pal *, Umashankar Panda, and Rourav Basak Variable Energy Cyclotron Centre, Kolkata, India

Abstract

Two helium liquefiers are working in tandem while one is always connected with the superconducting cyclotron at VECC. High pressure (HP) and low pressure (LP) controls are necessary to maintain varying helium flow to the cold box. Since these two liquefiers share the same HP and LP pipelines, any pressure fluctuation due to rapid change in flow sometimes causes trip of the liquefiers. To overcome this problem there is a need for fast responsive HP control. Introduction of derivative gain in the PID loop for fast action is not desirable as it creates instability to the control system. This problem was rectified by introducing a novel control scheme based on the forced opening of the unloading valve to push back helium gas to buffer tank by changing the offset of PI control as a function of Buffer Tank pressure. A simulation using Matlab Simulink was performed initially to check the performance of pressure control loop. The same is implemented in the control loop of the new liquefier and an experiment was performed. The experimental results obtained will be discussed in this paper.

INTRODUCTION

One helium liquefier is operational in VECC since 2001 [1]. The main application of this liquefier was to cater the cryoloads of the superconducting cyclotron in the form of superconducting magnet and cryopanels situated in main beam chamber for acceleration [2]. Another liquefier has been commissioned in VECC for redundancy purposes and for catering more refrigeration load [3]. Helium liquefiers are running in parallel - one running in refrigeration mode for the superconducting cyclotron and other running in liquefaction mode for supplying liquid helium to other users. The older liquefier cold box requires 50 g/s helium flow rate which is fed by either of the two cycle compressors of same capacity. The newly commissioned liquefier demands maximum 85 g/s flow rate of helium at refrigeration mode of operation. This requirement could be fulfilled either by a higher capacity compressor (85 g/s or above) or by two parallel connected compressors of lower capacity (50 g/s) as available earlier at VECC. In order to save expenditure, add reliability to the system and make use of the available compressors, a compressor of similar capacity was procured along with the new liquefier. For incorporating maximum flexibility to the cryogenic process, common warm pipelines are adopted and compressors are selected on ad hoc basis - one for old liquefier and two for new liquefier [4]. On the other hand, any flow instability, e.g. emergency stop of the compressor or cold box disconnection creates pressure fluctuation, that in turn causes trip of other liquefier. The trip of the liquefier connected with the cyclotron is totally undesirable as it results in uncontrolled release of pure helium gas from the cryostat and converts a significant amount of pure helium gas to impure form. A fast responsive control is necessary to cater these types of fluctuations. A newly adopted method is discussed here both with the simulation and experimental results.

SYSTEM DESCRIPTION

Figure 1 shows the cryogenic system of the superconducting cyclotron, it has pure gas management system (buffer tanks and pure cylinder quad for low and high pressure storage respectively), Oil Removal System (ORS), pressure control loops, cold boxes of two liquefiers, helium storage Dewars, distribution box and cryoloads.

The gas management system consists of two control valves,

- one unloading valve which automatically sends excess helium inventory to the buffers when the helium gas generated is in excess compared to the refrigeration system capacity; and

- one loading valve which adds helium inventory to the system from the buffer tank when the liquefaction capacity is higher than the flow recovered from the cryomodules.

These two valves work in concert with the by-pass valve, which automatically recycles excess flow from compressor discharge to suction.

Pressure control loops play a significant role to maintain the compressor suction and delivery pressure (LP and HP respectively) so that the flow rate to one cold box is independent to the fluctuation of the other. The pressure variation from the set-point also degrades the liquefier performance.

There are few alarms in compressor and cold box control depending on the suction and delivery pressure. Each cold box is equipped with two turbo-expanders for isentropic expansion and subsequent cool down of helium gas. These turbo-expanders' operation is very sensitive to pressure, as high inlet wheel pressure, high delivery pressure, low bearing pressure, break pressure beyond safe limit may cause damage to the turbines. Therefore, every fluctuation in pressure causes stoppage of the turboexpanders leading to the disconnection of the cold box.

^{*} sandip@vecc.gov.in



Figure 1: Schematic showing the cryogenic system consisting of buffer tanks, compressors, cold boxes, distribution boxes and cryogenic loads.

ROBUST PRESSURE CONTROL SYSTEM

To overcome pressure fluctuation problem there is a need for fast responsive HP control. This is only possible by adopting the derivative control in the PID (Proportional-Integral-Derivative) loop of HP control, but it generates pressure transients resulting in tripping of the turbo expanders. This problem was rectified by introducing a novel control scheme based on the forced opening of the unloading valve by changing the offset of Proportional-Integral (PI) control as a function of Buffer Tank pressure (BP). A better way is proposed by applying a linear shift on offset of the PI controller used for the opening of the unloading valve, i.e. HP to buffer, while HP is 2% more than setpoint. The more opening of that valve is proposed for the higher buffer tank pressure (BP) as per Equation (1), as the flow rate through a control valve is a function of the differential pressure between HP and BP as followed by Equation (2). It is also to be noted that valve opening is increased for lower value of offset.

$$Offset = (HP - BP) \times (Range) / (HP _ SP) \quad (1)$$

where, *Range* is the maximum range of the opening of the control value and HP_SP is the set point of the delivery pressure of the compressor.

Flow rate for compressible gas through a control valve

$$Q = C_{\nu} \cdot 2.80 \cdot \frac{\Delta x}{x} \cdot \sqrt{\frac{\Delta P(2P_1 - \Delta P)}{SG} \cdot \frac{273}{(273 + T_2)}}$$
(2)

where, C_v is the flow coefficient of the control valve, $\Delta x/x$ is the fractional opening of the control valve, ΔP is the differential pressure across the control valve, SG is the specific gravity of helium with respect to air at 20°C and 1 atm,

 T_2 is the downstream temperature of the control valve.

A simulation using Matlab Simulink has been performed initially to check the performance of pressure control loop. The same is implemented by two PI control loops - one for loading and unloading valves and other for the bypass valve. At start three compressors started together and after stabilization one of the compressors is stopped at 10000 s. The quantity of helium gas inside the compressor stopped is fed to the suction side creating an instant increase in LP pressure and decrease in HP pressure, as shown in Figure 3.

In Figure 3 normal PID control loops are used without the proposed modification. The pressure fluctuation in HP and LP is higher than the safe limit of compressor and cold box operation. Compressor suction pressure reaches to sub-atmospheric value and compressor delivery pressure jumps about ± 3 bar from the HP setpoint.

Figure 4 shows the same simulation result with the modification of the control loops in which an instant opening of the unloading valve transfers excess amount of gas to the buffer tank. It reduces pressure fluctuations both in HP and LP and an increase in BP.



Figure 2: Pressure control of compressor suction and delivery for constant pressure operation of helium liquefier.



Figure 3: Simulation result (LP, HP and BP pressures) of warm helium process of the cryogenic system; only two normal PI loops are operational without any modification.



Figure 4: Simulation result (LP, HP and BP pressures) of warm helium process of the cryogenic system; with implementation of new unloading valve PID loop.



Figure 5: The fluctuation in opening of PCV289 to accommodate excess gas due to stopping of one compressor and that of PCV280 at the time of its start.

Figure 5 shows the fluctuation in opening of the loading and unloading valves (PCV 280 and PCV 289 respectively as shown in Figure 1 for Helial 2000) and bypass valve (PCV 275) before and after implementation of the new control scheme. The opening of PCV 289 is higher after modification because of the offset but the fluctuations in PCV 280 and PCV 275 get reduced after modification. Similarly, the fluctuations in HP and LP get reduced after implementation of new scheme as shown in Figure 6.

The same has been tried again by stopping the third compressor forcefully and displayed in Figure 7. Just after the compressor stopped, LP pressure increased and HP pressure decreased because of sudden flow rate reduction for the opening of the bypass valve. As the bypass valve took over the control and the suction side excess gas fed to the delivery side, LP pressure decreased and HP pressure increased to 14.3 bara. At that instant of time the unloading valve opened to some extent and it resulted in pushing the excess cycle helium gas from process to buffer tank. Ultimately, entire system stabilizes without causing any trip or alarm.



Figure 6: Buffer, LP and HP pressure curves before and after implementation of the proposed control scheme based on HP and buffer pressure.



Figure 7: Buffer, LP and HP pressure curves at start and stop of a compressor with other two compressors running after implementation of the proposed control scheme.

CONCLUSION

This novel control scheme shows a way of smooth transfer of compressors in case of parallel run for liquefiers. This scheme has been running for more than two years without a single failure. Due to physical location of the running compressors, there is increase in pressure at the compressor outlet due to pressure drop in the process line between HP sensor and compressor delivery end. This problem may be subdued by proper setting of the compressor control. While a variable frequency drive (VFD) is put for flow rate control of the compressors, the new control scheme would be modified to take care of that situation. The VFD setpoint should be set in coordination with the bypass valve opening (in order to save electrical energy).

REFERENCES

- R Dey et. al., "Operational Experience of Helium Liquefier-Refrigerator after interfacing with K-500 Superconducting Cyclotron", Proc. of Indian Particle Accelerator Conference, VECC, Kolkata, March 1-5, 2005.
- [2] Saha, Subimal; Choudhury, Jayanta; Pal, Gautam; Hajra, D. P.; Dey, R.; Sur, Amitava; Bhandari, R. K., Superconducting magnet for K-500 cyclotron at VECC, Kolkata, Cryogenics, v. 49(6), p. 235-242.
- [3] Sandip Pal et. al., "Experiences in the commissioning of new helium liquefier and the respective modifications for improvement at VECC", Proc. of 23rd National Symposium on Cryogenics, October 28-30, 2010.
- [4] Umashankar Panda, Sandip Pal, Ranadhir Dey, Tanushyam Bhattacharjee, and Aditya Mandal, "Control system of cryogenic plant for superconducting cyclotron at VECC," Proceedings of CYCLOTRONS 2010, Langzhou, China, MOPCP008, pp 57-59.

List of Authors

Bold papercodes indicate primary authors

— A —		Chachondia, A.S.	THCA05
Abe. T.	THIA02	Chaddha, N.	THPD10
Acharva, S.	THPD03 THPD04	Chakrabarti, A.	THPD46
Aghababyan, A.	THIB04	Chakravarthy, D.P.	THCA05, THPD43
Agrawal, R.K.	WEPD01 THPD06 THPD17	Chandan, S.	THPD45
Amselem, A.	THTA02	Chatterjee, A.	WEPD16, FRCC02
Ananthkrishnan, T.	THPD26, FRCB03	Chatterjee, M.	WEPD39
Ansari M A	THPD27	Chatterjee, S.	FRCC04
Antony J	THPD15. FRCA02	Chattopadhyay, P.K.	WEPD11
Arredondo I	THCB01, THPD47	Chattopadhyay, S.	THPD36
		Chaudhari, S.N.	THCD06
D		Chauhan, A.	WEPD09
-B-		Chavan, R.B.	THPD45
Bacher, R.	WEIC01	Chavda, C.K.	WEPD13
Badillo, I.	THPD47	Cho, YS.	FRCB02
Bär, R.	WEPD48,FRIA01	Chouhan, H.S.	THCD06
Bagduwal, P.S.	THPD16 , THPD35	Cilingaryan, S.	THCA06
Balalykin, N.	THPD32	Clausen, M.R.	WEKA01, THPD02
Balasubramanian, A.	THPD49	Claustre, L.	THIC01
Balewski, K.	WEPD22	Cleva, S.	WEPD34
Balzer, B.M.	THCA06	Coutinho, T.M.	THIC01
Barpande, K.G.	THPD06, THPD17		
Barua, L.	THPD36	D	
Baruah, U.K.	THPD40	—U—	
Basak, R.	FRCD02	Das, M.K.	THPD36
Basak, S.	THPD20, THPD46	Das, T.	THPD19
Basu, A.	THPD26, THPD33	Datta, K.	THPD11
Beck, D.H.	WEPD48,THPD44, FRIA01	Datta, T.S.	FRCA02
Beckmann, A.	THCD05	Dave, H.D.	THCC03
Behere, A.	WEPD10	Dave, R.	THPD40
Belsare, S.M.	WEPD18	De, A.	THPD36
Berkaev, D.E.	WEPD14, FRCB04	Deb, S.K.	THCD04
Berlik, S.	THPD18	del Campo, M.	THCB01, THPD47
Bhagwat, P. V.	WEPD27, FRCC02	Dey, M.S.	THPD50
Bhanage, V.P.	THPD27	Dhara, P.	FRCC01, FRCC04
Bharade, S.K.	THPD26 , FRCB03	Dhola, H.A.	THPD40
Bhaskar, P.	THPD36	Dhongde, J.A.	WEPD12
Bhattacharjee, T.	THPD11 , FRCC03	Di Calafiori, D.R.S.	FRCB01
Bhattacharya, S.	THPD19	Di Giovenale, D.	FRID01
Bhattacharyya, T.	THPD19	Di Pirro, G.	THCB03, FRID01
Bhaumik, T.K.	THPD12	Dissertori, G.	FRCB01
Bhole, R.B.	WEPD25, THCB02, THPD10, THP	D1 Diwakar, M.P.	WEPD10,WEPD28
Bisegni, C.	FRID01	Dixit, K.	THPD45
Biswas, B.B.	WEPD33, THCA05	Doshi, K.J.	WEPD12
Bogani, A.I.	WEPD34	Dubey, V.K.	THCD04
Bonnes, U.	THPD13, THPD14	Dutt, R.N.	WEPD38
Brand, H.	THPD44	Dutta, D.P.	THPD20, THPD49
Burandt, C.	WEPD03, THPD13, THPD14	Dutta Roy, A.	THPD36
Buteau, A.	THIC01	Duval, P.	THIB04

- C		— E —	
Catani, L.	FRID01	Edappala, P.V.	THPD30
Cecilia, A.	THCA06	Eguiraun, M.	THPD47

Ehrlichmann, H. Enders, J.	THPD18 WEPD03,THPD13,THPD14	Jain, S.K. Jang, JH.	WEPD33 FRCB02
		Jha, K.	WEPD10
— F —		Jha, R.	WEPD11, THPD30
Forneworth R I	WEDD44	Joshi, G.	THPD26, FRCB03
Fatnani P	WETBAS WEPDA1 WEPDAQ THPDA	Joshi, K.D.	THPD05
Fernandes R N	WECC03	Joshi, S.	WEPD23
Forgetta I G	FRID01	Joti, Y.	THIA02
Fukui T	THPD09	Journeaux, J.Y.	THPD40
Furukawa, K.	WECC02.WEPD26	Jugo, J.	THCB01, THPD47
Furukawa, Y.	THIA02		
· •· •· •· •· •· •· •· •· •· •· •· •· •·		— K —	
G		Kailas, S.	WEPD27, FRCC02
-u-	TUPP 40	Kalra, M.	WEPD33
Gajjar, S.		Kameshima, T.K.	THIA02
Galodiya, K.D.		Kamikubota, N.	WEIB02,WEPD47
Ganesh, G.		Kanjilal, D.	WEPD38
Gangull, I.		Karabekyan, S.	THCD05
		Karna, G.	THPD49
Garai, M.		Kasaev, A.S.	FRCB04
Gaul, S. Charat S		Kasliwal, A.	THPD28
Gharat, S.		Kewlani, H.M.	THPD43
Giacchini M.G		Khan, M.S.	WEPD18
Giovannini I G	тнсвот	Khare, V.K.	THPD36
Gobel N.C.	WEPD28	Khirwadkar, S.S.	WEPD18
Gond S		Khristi, Y.S.	WEPD12
Gore JA	WEPD27 FRCA04 FRCC02	Kiyomichi, A.	TH1A02
Gothwal P	WEPD09	Kizhupadath, K.K.	WEPD23
Götz A	THICOL	Kling, A. Kabawashi M	WEPD22
Guha, S.	THPD12	Kobayashi, Y.	
Gupta, A.K.	WEPD27	Koley, D.	
Gupta, R.P.	ТНСС03	Koon J	EDCR04
Gupta, S.K.	THPD33	Koomann A	
Gupta, V.	THPD40	Kosugo T	WEDD24
		Kozak VB	
H		Kreider M	WEPD48 FRTA01
		Kulkarni S	WEPD27, FRCA04, FRCC02
Haas, D.	WEPD19, IHCAUG	Kumar A	WEPD16
Hannurkar, P.R.		Kumar, A.	THIA03
Halje, J.		Kumar, A.	THPD20
Hansler O		Kumar, R.	THPD15
Horfurth E		Kumar, R.A.V.	WEPD23
Heron MT	WEDD52	Kumar, U.	THPD36
Higurashi V	WECC02	Kumari, P.	WEPD11
Hirono T		Kumari, S.	THPD36
Holme O	FRCBA1	Kwon, HJ.	FRCB02
Hua F	THPD14	Kyrpotin, A.N.	FRCB04
109,11			
1		-L-	
		Lad. M.	THPD16.THPD35
lida, N.	WEPD26	Lakshmy. P.S.	WEPD38
ISNII, M.	14209	Lathi, D.	THPD40
		Lustermann, W.	FRCB01
— J —		Lysenko, A.P.	FRCB04
Jain, L.	THPD27		

— M —		Panda, U.	THPD21, FRCD02
Mahata, K.	WEPD16.FRCC02	Pandey, H.K.	THPD20, THPD46, THPD49
Mahavar, K.	WEPD23	Pandit, S.K.	FRCC02
Maity. P.	FRCC01	Pandit, T.G.	THPD28
Mallik. C.	THPD12	Pandit, V.S.	WEPD43
Mandal, A.	THPD21_THPD50	Panschow, W.	WEPD48
Mandaliva, H.D.	THPD30	Paresh, P. M.	FRCB03
Mandi, T.K.	THPD20	Parghi, B.A.	WEPD12
Mandi, T.K.M.	THPD49	Parkar, V.V.	FRCC02
Masand, H.A.	WEPD12	Parmar, D.C.	THPD40
Masuda, T.	THPD09	Pasic, H.	THCA06
Mathur, Y.	WEPD38	Patel, A.M.	THPD40
Mathuria. D.S.	FRCA02	Patel, D.A.	WEPD12
Maurva, V.K.	THCD06	Patel, J.J.	WEPD11
Mavva, Y.S.	WEKA02	Patel, K.G.	WEPD13
Mazzitelli. G.	FRID01	Patel, N.C.	WEPD13
Merh. B.N.	WEPD01 THPD17	Patel, T.H.	WEPD18
Mexner, W.	WEPD19 THCA06	Patil, M.B.	THCA05
Meyer, J.M.	THIC01	Patil, R.K.	THCA05
Minashkin. V.	THPD32	Penilop, P.S.P.	THPD49
Mishra. L.	THPD43	Penning, J.	WEKA01
Mishra, R.	WEPD01	Pepler, D.A.	THPD22
Mittal, K.C.	THCA05, THPD03, THPD04, THPD4	Betitdemange, S.	THIC01
Möller, M.	THPD02	Pflüger, J.	THCD05
Motiwala, P.D.	THPD26	Pietralla, N.	WEPD03,THPD13,THPD14
Mourougayane, K.	THPD49	Piso, D.	THPD47
Mukesh Kumar, M.K.	THCA05	Pithawa, C.K.	WEPD10,WEPD28,THPD26,FRCB03
Mukherjee, S.	THPD48	Pivetta, L.	WEPD34
		Pradhan, S.	WEPD12
— N —		Prados, C.	WEPD48, FRIA01
Nabhirai BV		Prasad, U.A.	WEPD12
Naominaj, F. I. Naomin S. B.V			
Nagraju, J.D.V. Nair PM	WEDD10 WEDD28	— R —	
Nandi C		Rajan, R.N.	THPD03
Nandy PP		Rajpal, R.	WEPD11, THPD30
Naravana C	THERE		
Navatha, C.P.	THCD04	Ramachandran, K.	WEPD16
		Ramachandran, K. Rapubhy	WEPD16 WEPD38
Navallie, C.F.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27	Ramachandran, K. Raphbok7, Rauch, S.	WEPD16 WEPD38 WEPD48,FRIA01
Neidherr D	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44	Ramachandran, K. Raphby, 7 Rauch, S. Raval, B.M.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40
Neidherr, D.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04	Ramachandran, K. Rauch, S. Raval, B.M. Rawat, A.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06
Neidherr, D. Nielsen, J.S. Nozdrin, M.A.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32	Ramachandran, K. Rauch, S. Raval, B.M. Rawat, A. Ray, K. P.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20
Neidherr, D. Nielsen, J.S. Nozdrin, M.A.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32	Ramachandran, K. Raphbb K7, Rauch, S. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04
Neidherr, D. Nielsen, J.S. Nozdrin, M.A.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32	Ramachandran, K. Rauch, S. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05
Neidherr, D. Nielsen, J.S. Nozdrin, M.A.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32	Ramachandran, K. Rauch, S. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD02
Neidherr, D. Nielsen, J.S. Nozdrin, M.A. — O — Ohata, T.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32 THIA02	Ramachandran, K. Raphob K7, Rauch, S. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S. Rhyder, A.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD02 WECC03
Neidherr, D. Nielsen, J.S. Nozdrin, M.A. — O — Ohata, T. Okazaki, T.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32 THIA02 WEPD26	Ramachandran, K. Rauch, S. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S. Rhyder, A. Rodrigues, G.O.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD02 WECC03 WEPD38
Neidherr, D. Nielsen, J.S. Nozdrin, M.A. — O — Ohata, T. Okazaki, T. Okumua, R.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32 THIA02 WEPD26 FRCA03	Ramachandran, K. Raphbo K7, Rauch, S. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S. Rhyder, A. Rodrigues, G.O. Rogovsky, Yu. A.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD02 WECC03 WEPD38 FRCB04
Neidherr, D. Nielsen, J.S. Nozdrin, M.A. — O — Ohata, T. Okazaki, T. Okumua, R. Ounsy, M.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32 THIA02 WEPD26 FRCA03 THIC01	Ramachandran, K. Raphbb K7, Rauch, S. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S. Rhyder, A. Rodrigues, G.O. Rogovsky, Yu. A. Romanov, A.L.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD05 THPD02 WECC03 WEPD38 FRCB04 FRCB04
Neidherr, D. Nielsen, J.S. Nozdrin, M.A. — O — Ohata, T. Okazaki, T. Okumua, R. Ounsy, M.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32 THIA02 WEPD26 FRCA03 THIC01	Ramachandran, K. Ray B.M. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S. Rhyder, A. Rodrigues, G.O. Rogovsky, Yu. A. Romanov, A.L. Roy, A.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD05 THPD02 WECC03 WEPD38 FRCB04 FRCB04 FRCB04 WEPD25,WEPD26,THCB02,THPD11
Neidherr, D. Nielsen, J.S. Nozdrin, M.A. — O — Ohata, T. Okazaki, T. Okumua, R. Ounsy, M. — P —	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32 THIA02 WEPD26 FRCA03 THIC01	Ramachandran, K. Rauch, S. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S. Rhyder, A. Rodrigues, G.O. Rogovsky, Yu. A. Romanov, A.L. Roy, A. Roy, A.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD05 THPD02 WECC03 WEPD38 FRCB04 FRCB04 FRCB04 WEPD25,WEPD26,THCB02,THPD11 FRCC01,FRCC04
Neidher, D. Nielsen, J.S. Nozdrin, M.A. — O — Ohata, T. Okazaki, T. Okumua, R. Ounsy, M. — P — Pace, E.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32 THIA02 WEPD26 FRCA03 THIC01 THCB03	Ramachandran, K. Ray, B.M. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S. Rhyder, A. Rodrigues, G.O. Rogovsky, Yu. A. Romanov, A.L. Roy, A. Roy, A. Roy, D.A.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD05 WECC03 WEPD38 FRCB04 FRCB04 FRCB04 FRCB04 WEPD25,WEPD26,THCB02,THPD11 FRCC01,FRCC04 WEPD33
Neidherr, D. Nielsen, J.S. Nozdrin, M.A. — O — Ohata, T. Okazaki, T. Okumua, R. Ounsy, M. — P — Pace, E. Pal, G.P.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32 THIA02 WEPD26 FRCA03 THIC01 THCB03 THPD19	Ramachandran, K. Raval, B.M. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S. Rhyder, A. Rodrigues, G.O. Rogovsky, Yu. A. Romanov, A.L. Roy, A. Roy, A. Roy, D.A. Roy, P.S.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD05 THPD02 WECC03 WEPD38 FRCB04 FRCB04 FRCB04 FRCB04 FRCC01,FRCC04 WEPD33 FRCC01
Neidherr, D. Nielsen, J.S. Nozdrin, M.A. — O — Ohata, T. Okazaki, T. Okumua, R. Ounsy, M. — P — Pace, E. Pal, G.P. Pal, S.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32 THIA02 WEPD26 FRCA03 THIC01 THCB03 THPD19 WEIB01,WEPD01,WEPD25,THCB0	Ramachandran, K. Ray, B.M. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S. Rhyder, A. Rodrigues, G.O. Rogovsky, Yu. A. Romanov, A.L. Roy, A. Roy, A. Roy, D.A. Roy, P.S. Roychowdhury, P.	WEPD16 WEPD38 WEPD48,FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD05 THPD02 WECC03 WEPD38 FRCB04 FRCB04 FRCB04 FRCB04 FRCB04 FRCC01,FRCC04 WEPD33 FRCC01 THPD43
Neidherr, D. Nielsen, J.S. Nozdrin, M.A. — O — Ohata, T. Okazaki, T. Okumua, R. Ounsy, M. — P — Pace, E. Pal, G.P. Pal, S.	THCD04 WEPD01, WEPD09, THCD04, THPD0 THPD27 THPD44 THCA04 THPD32 THIA02 WEPD26 FRCA03 THIC01 THCB03 THPD19 WEIB01, WEPD01, WEPD25, THCB0 THPD11, FRCC03	Ramachandran, K. Ray, B.M. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S. Rhyder, A. Rodrigues, G.O. Rogovsky, Yu. A. Romanov, A.L. Roy, A. Roy, A. Roy, D.A. Roy, P.S. Roy, P.S. Roy, P.S.	WEPD16 WEPD38 WEPD48, FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD05 THPD02 WECC03 WEPD38 FRCB04 FRCB04 FRCB04 WEPD25, WEPD26, THCB02, THPD11 FRCC01, FRCC04 WEPD33 FRCC01 THPD43
Neidherr, D. Nielsen, J.S. Nozdrin, M.A. — O — Ohata, T. Okazaki, T. Okumua, R. Ounsy, M. — P — Pace, E. Pal, G.P. Pal, S.	THCD04 WEPD01,WEPD09,THCD04,THPD0 THPD27 THPD44 THCA04 THPD32 THIA02 WEPD26 FRCA03 THIC01 THCB03 THPD19 WEIB01,WEPD01,WEPD25,THCB0 THPD11,FRCC03 FRCD02	Ramachandran, K. Rauch, S. Raval, B.M. Rawat, A. Ray, K. P. Rehlich, K. Reju, K. Rettig-Labusga, S. Rhyder, A. Rodrigues, G.O. Rogovsky, Yu. A. Romanov, A.L. Roy, A. Roy, A. Roy, D.A. Roy, P.S. Roy, P.S. Roy, P.S.	WEPD16 WEPD38 WEPD48, FRIA01 THPD40 THCD06 THPD20 THIB04 THPD05 THPD02 WECC03 WEPD38 FRCB04 FRCB04 FRCB04 WEPD25, WEPD26, THCB02, THPD11 FRCC01, FRCC04 WEPD33 FRCC01 THPD43

Saha Das. S. Sahoo, G.K. Sahoo, S. Sahu, B.K. Saifee, K. Samanta, T. Sarkar, D. Sarkar, D. Sato, N. Saxena. Mr. Saxena, P. Schegolev, V.Y. Schoeneburg, B. Schunke, B. Senchenko, A.I. Seth, S. Shah, M. Sharma, A. Sharma, A.N. Sharma, D. Sharma, V. Shatunov, P.Yu. Shatunov, Y.M. Shirkov, G. Shrivastava, A. Singh, A. Singh, I. Singh, K. Singh, M.N. Singh, N.P. Singh, P. Singh, S. Singhai, P. Singleton, S.J. Sinha, A.K. Som, S.S. Song, Y.-G. Spangenberg, T. Sridharan, P. Srivastava, B.S.K. Srivastava, S. Stankevich, A.S. Starritt, A. C. Stecchi, A. Sugimoto, T. Sujo, C.I. Svensson, O. Svensson, S.

THPD36 WEPD22 THPD10, FRCC03 THIA03 WEPD09 THPD11, THPD48 THPD48 THCB02, THPD11 FRCA03 THPD11 THCD04 THPD32 THPD02 THPD40 WEPD14, FRCB04 THPD21, THPD50 THPD30 WEPD23 WEPD12 THPD16, THPD35 THPD03, THPD04 FRCB04 FRCB04 THPD32 FRCC02 THPD50 THCD04 THIA03 THCD04 THPD40 THPD26, THPD33, FRCA04 THPD26, THPD33, FRCA04 FRCC01, FRCC04 WEPD52 THCD04 THPD21, THPD50 FRCB02 WEPD19, THCA06 WEPD10,WEPD28 THPD06 WEPD43 FRCB04 WECC03 FRID01 THIA02 FRCB03 THIC01

THPD40

—T—		
Takamiya, K. Tanaka, R. Tanigaki, M. Terpstra, W.W. Thakar, A.M. Thakur, S.K. Thieme, M. Tillu, A.R. Tiwari, N. Tokuhisa, A. Tomar, S.S. Trubnikov, G.V.	FRCA03 THIA02 FRCA03 WEPD48, FRIA01 THPD40 THPD36 WEPD48 THPD45 THPD45 THPD16, THPD35 THIA02 THCD06 THPD32	
— U —		
Uchiyama, A. Ueda, S. Upadhyay, A.	WECC02 THPD09 THCD04	
_ V _		
Vagovic, P. Vaidya, U.W. Varmora, P.A. Vasserman, I. Verma, G. Vijayan, K. Vogelgesang, M. Vora, H.S.	THCA06 THIA01 WEPD12 WEPD44 WEPD33 WEPD52 THCA06 THCD04	
— W —		
Worm, T.	THCA04	
— X —		
Xu, J.Z.	WEPD44	
— Y —		
Yadav, V. Yamaga, M. Yamamoto, N. Yoshida, S.Y. Yoshinaga, H. Yoshino, H.	THPD45 THIA02 WEPD47 WEPD47 FRCA03 FRCA03	
Z		
Zani, F. Zelepoukine, S. Zweig, M.	FRID01 FRCB01 WEPD48,FRIA01	

Institutes List

ANL

Argonne, USA

- Farnsworth, R.I.
- Vasserman, I.
- Xu, J.Z.

ASCo

Clayton, Victoria, Australia

- Fernandes, R.N.
- Rhyder, A.
- Starritt, A. C.

BARC

Mumbai, India

- Acharya, S.
- Ananthkrishnan, T.
- Basu, A.
- Behere, A.
- Bhagwat, P. V.
- Bharade, S.K.
- Biswas, B.B.
- Chachondia, A.S.
- Chakravarthy, D.P.
- Chandan, S.
- Chatterjee, A.
- Chavan, R.B.
- Diwakar, M.P.
- Dixit, K.
- Ganesh, G.
- Gantayet, L.M.
- Gaur, S.
- Gharat, S.
- Gond, S.
- Gore, J.A.
- Gupta, A.K.
- Gupta, S.K.Jain, S.K.
- Jain, S.i
 Jha, K.
- Joshi, G.
- Joshi, K.D.
- Kailas, S.
- Kallas, S. • Kalra, M.
- Kalla, IVI.
- Kewlani, H.M.
- Kulkarni, S.
- Kumar, A.
- Mahata, K.
- Mayya, Y.S.Mishra, L.
- Misrira, L.
 Mittal, K.C.
- Motiwala, P.D.
- Mukesh Kumar, M.K.
- Nagraju, S.B.V.
- Nair, P.M.
- Pandit, S.K.
- Paresh, P. M.
- Parkar, V.V.
- Patil, M.B.

Institutes List

- Patil, R.K.
- Pithawa, C.K.
- Rajan, R.N.
- Ramachandran, K.
- Reju, K.
- Roy, D.A.
- Roychowdhury, P.
- Sharma, V.
- Shrivastava, A.
- Singh, P.
- Singh, S.
- Sridharan, P.
- Sujo, C.I.
- Tillu, A.R.
- Vaidya, U.W.
- Verma, G.
- Yadav, V.

Bhabha Atomic Research Centre

- Mumbai, India
 - Gohel, N.C.

BINP SB RAS

Novosibirsk, Russia

- Berkaev, D.E.
- Kasaev, A.S.Koop, I.
- Kozak, V.R.
- Kyrpotin, A.N.
- Lysenko, A.P.
- Rogovsky, Yu. A.
- Romanov, A.L.
- Senchenko, A.I.
- Shatunov, P.Yu.
- Shatunov, Y.M.
- Stankevich, A.S.

BRIT

Bidhan Nagar, India

- Barua, L.
- Chattopadhyay, S.
- Das, M.K.
- Kumar, U.
- Saha Das, S.

CELLS-ALBA Synchrotron

Cerdanyola del Vallès, Spain

• Coutinho, T.M.

• Aghababyan, A.

Hamburg, Germany

• Bacher, R.

Balewski, K.

Clausen, M.R.

293

DESY

- Duval, P.
- Ehrlichmann, H.
- Hatje, J.
- Hensler, O.
- Kling, A.
- Möller, M.
- Penning, J.
- Rehlich, K.
- Rettig-Labusga, S.
- Sahoo, G.K.
- Schoeneburg, B.

Diamond

Oxfordshire, United Kingdom

- Heron, M.T.
- Singleton, S.J.
- Vijayan, K.

EJIT

Hitachi, Ibaraki, Japan

• Okazaki, T.

ELETTRA

- Basovizza, Italy
 - Bogani, A.I.
 - Cleva, S.
 - Pivetta, L.

ESRF

Grenoble, France

- Claustre, L.
- Götz, A.
- Meyer, J.M.
- Petitdemange, S.
- Svensson, O.

ESS Bilbao

LEIOA, Spain

- Arredondo, I.
- Badillo, I.
- del Campo, M.
- Eguiraun, M.
- Piso, D.

ETH

Zurich, Switzerland

- Di Calafiori, D.R.S.
- Dissertori, G.
- Holme, O.
- Lustermann, W.

GSI

294

Darmstadt, Germany

- Bär, R.
- Beck, D.H.
- Brand, H.

- Herfurth, F.
- Kreider, M.
- Neidherr, D.
- Panschow, W.
- Prados, C.
- Rauch, S.
- Terpstra, W.W.
- Thieme, M.
- Zweig, M.

HITK

Kolkata, India • Chatterjee, S.

IITKGP

West Bengal, India

Sarkar, D.

INFN-Roma II

Roma, Italy

- Catani, L.
- Zani, F.

INFN/LNF

Frascati (Roma), Italy

- Bisegni, C.
- Di Giovenale, D.Di Pirro, G.
- Di Fillo, G.
- Foggetta, L.G.Mazzitelli, G.
- Mazzitelli, G
 Pace, E.
- Stecchi, A.

INFN/LNL

Legnaro (PD), Italy

- Giacchini, M.G.
- Giovannini, L.G.

IPR

Bhat, Gandhinagar, India

- Baruah, U.K.
- Belsare, S.M.
- Chattopadhyay, P.K.
- Chavda, C.K.
- Dave, H.D.
- Dave, R.
- Dhola, H.A.
- Dhongde, J.A.Doshi, K.J.
- Edappala, P.V.
- Gajjar, S.
- Galodiya, K.D.
- Gupta, R.P.
- Gupta, N.
 Gupta, V.
- Jha, R.
- Khan, M.S.
- Khirwadkar, S.S.

Kizhupadath, K.K.

Institutes List

• Khristi, Y.S.

- Kumar, R.A.V.
- Kumari, P.
- Mahavar, K.
- Mandaliya, H.D.
- Masand, H.A.
- Parghi, B.A.
- Parmar, D.C.
- Patel, A.M.
- Patel, D.A.
- Patel, J.J.
- Patel, K.G.
- Patel, N.C.
- Patel, T.H.
- Pradhan, S.
- Prasad, U.A.
- Rajpal, R.
- Raval, B.M.
- Shah, M.
- Sharma, A.N.
- Singh, N.P.
- Thakar, A.M.
- Varmora, P.A.

ISA

Aarhus, Denmark

- Nielsen, J.S.
 - Worm, T.

ITER Organization

St. Paul lez Durance, France

- Journeaux, J.Y.
- Lathi, D.
- Schunke, B.
- Svensson, S.

IUAC

New Delhi, India

- Antony, J.
- Datta, T.S.
- Dutt, R.N.
- Kanjilal, D.
- Kumar, A.
- Kumar, R.
- Lakshmy, P.S.
- Mathur, Y.
- Mathuria, D.S.
- Rao, U.K.
- Rodrigues, G.O.
- Sahu, B.K.
- Singh, K.

J-PARC, KEK & JAEA

- Ibaraki-ken, Japan
 - Yamamoto, N.

JASRI/SPring-8

Institutes List

Hyogo-ken, Japan

- Amselem, A.
- Furukawa, Y.
- Hirono, T.
- Ishii, M.
- Joti, Y.
- Kameshima, T.K.
- Kiyomichi, A.
- Masuda, T.
- Ohata, T.
- Sugimoto, T.
- Ueda, S.
- Yamaga, M.

JINR

Dubna, Moscow Region, Russia

- Balalykin, N.
- Minashkin, V.
- Nozdrin, M.A.
- Schegolev, V.Y.
- Shirkov, G.
- Trubnikov, G.V.

JNCASR

Bangalore, India

• Narayana, C.

KAERI

Daejon, Republic of Korea

- Cho, Y.-S.
- Jang, J.-H.
- Kwon, H.-J.
- Song, Y.-G.

Kanto Information Service (KIS), Accelerator Group

- Ibaraki, Japan
 - Yoshida, S.Y.

Karlsruhe Institute of Technology

- Eggenstein-Leopoldshafen, Germany
 - Balzer, B.M.
 - Cilingaryan, S.

KEK

Ibaraki, Japan

- Furukawa, K.
- lida, N.
- Kamikubota, N.
- Kosuge, T.

KIT

Eggenstein-Leopoldshafen, Germany

295

- Cecilia, A.
- Haas, D.

• Pasic, H.

Kopmann, A.Mexner, W.

- Spangenberg, T.
- Vagovic, P.
- Vogelgesang, M.

Kyoto University, Research Reactor Institute

Osaka, Japan

- Kobayashi, Y.
- Okumua, R.
- Sato, N.
- Takamiya, K.
- Tanigaki, M.
- Yoshinaga, H.
- Yoshino, H.

Nirma University

- Ahmedabad, India
 - Joshi, S.
 - Sharma, A.

RIKEN Nishina Center

Wako, Japan

• Higurashi, Y.

RIKEN SPring-8 Center, Innovative Light Sources Division Hyogo, Japan

- Abe, T.
- Tanaka, R.

RIKEN SPring-8 Center

Sayo-cho, Sayo-gun, Hyogo, Japan

- Hatsui, T.
- Tokuhisa, A.

RIKEN/SPring-8

- Hyogo, Japan
 - Fukui, T.

RRCAT

Indore (M.P.), India

- Agrawal, R.K.
- Ansari, M.A.
- Bagduwal, P.S.
- Barpande, K.G.
- Bhanage, V.P.
- Chaudhari, S.N.
- Chauhan, A.
- Chouhan, H.S.
- Deb, S.K.
- Dubey, V.K.
- Fatnani, P.
- Ganguli, T.
- Gothwal, P.
- Hannurkar, P.R.
- Jain, L.
- Kasliwal, A.
- Lad, M.

- Maurya, V.K.
- Merh, B.N.
- Mishra, R.
- Navathe, C.P.Pandit, T.G.
- Pandit, T.G.
 Rawat, A.
- Saifee, K.
- Sallee, R.
 Saxena, P.
- Saxena, T.
 Sharma, D.
- Singh, I.
- Singh, M.N.
- Sinha, A.K.
- Srivastava, B.S.K.
- Tiwari, N.
- Tomar, S.S.
- Upadhyay, A.
- Vora, H.S.

SAMEER

Chennai, India

- Balasubramanian, A.
 - Karna, G.
 - Kumar, A.
 - Mourougayane, K.
 - Penilop, P.S.P.
 - Ray, K. P.

Sokendai

- Ibaraki, Japan
 - Uchiyama, A.

SOLEIL

- Gif-sur-Yvette, France
 - Buteau, A.
 - Ounsy, M.

STFC/RAL

Chilton, Didcot, Oxon, United Kingdom • Pepler, D.A.

TU Darmstadt

Darmstadt, Germany

- Bonnes, U.
- Burandt, C.
- Enders, J.
- Hug, F.
- Konrad, M.
- Pietralla, N.

University of Siegen

Siegen, Germany • Berlik, S.

University of the Basque Country, Faculty of Science and Technology Bilbao, Spain

Jugo, J.

UW-Madison/PD

Madison, Wisconsin, USA

Zelepoukine, S.

VECC

Kolkata, India

- Basak, S.
- Bhaskar, P.
- Bhattacharjee, T.
- Bhattacharya, S.
- Bhattacharyya, T.
- Bhaumik, T.K.
- Bhole, R.B.
- Chaddha, N.
- Chakrabarti, A.
- Chatterjee, M.
- Das, T.
- Datta, K.
- De, A.
- Dey, M.S.
- Dhara, P.
- Dutta Roy, A.
- Dutta, D.P.
- Garai, M.
- Ghosh, S.
- Guha, S.
- Khare, V.K.
- Koley, D.
- Kumari, S.
- Maity, P.
- Mallik, C.
- Mandal, A.
- Mandi, T.K.M.
- Mandi, T.K.
- Mukherjee, S.
- Nabhiraj, P.Y.
- Nandi, C.
- Nandy, P.P.
- Pal, G.P.
- Pal, S.S.
- Pal, S.
- Panda, U.
- Pandey, H.K.
- Pandit, V.S.
- Roy, A.
- Roy, P.S.
- Saha, S.
- Sahoo, S.
- Samanta, T.
- Sarkar, D.
- Saxena, Mr.
- Seth, S.
- Singh, A.
- Singhai, P.
- Som, S.S.
- Srivastava, S.
- Thakur, S.K.

WBSEDCL

India

• Basak, R.

XFEL. EU

- Hamburg, Germany
 - Beckmann, A.
 - Karabekyan, S.
 - Pflüger, J.