ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA166

CONTROL SYSTEM INTEGRATION OF A MicroTCA.4 BASED DIGITAL LLRF USING THE CHIMERATK OPC UA ADAPTER

R. Steinbrück†, M. Kuntzsch, P. Michel, Helmholtz-Zentrum Dresden-Rossendorf, Dresden
M. Killenberg, M. Hierholzer, H. Schlarb Deutsches Elektronen-Synchrotron, Hamburg (DESY)
C. Iatrou, J. Rahm, L. Urbas Professur für Prozessleittechnik, Technische Universität Dresden, Dresden (PLT-TUD)

Abstract

The superconducting linear electron accelerator ELBE at Helmholtz-Zentrum Dresden-Rossendorf is a versatile light source. It operates in continuous wave (CW) mode to provide a high average beam current. In order to meet the requirements for future high resolution experiments the analogue low level radio frequency control (LLRF) is currently replaced by a digital MicroTCA.4 [1] LLRF system based on a development at DESY, Hamburg.

Operation and parametrization is realized by a server application implemented by DESY using the ChimeraTK software framework. To interface the WinCC 7.3 based ELBE control system an OPC UA adapter for ChimeraTK has been developed in cooperation of DESY, Technische Universität Dresden (TUD) and HZDR. The contribution gives an overview of the collaborating parties, the variable mapping scheme used to represent LLRF data in the OPC UA server address space and integration experiences with different industrial OPC UA Clients like WinCC 7.3 and LabVIEW.

INTRODUCTION

The ELBE accelerator has been operated with an analogue LLRF system since more than 15 years. To meet increasing demands on beam quality (jitter and energy stability) it has been decided to migrate to a digital FPGA based LLRF system developed by DESY for the European XFEL. Thus allowing for the implementation of advanced control methods and beam based feedback loops.

The adaption of the MicroTCA.4 LLRF system required:

- hard- and software revision the for CW operation
- interfacing the ELBE control system

The latter topic is in the main scope of this paper.

SYSTEM DESCRIPTON

Digital LLRF

The MicroTCA.4 based LLRF is a multi-layer system: the FPGA-Firmware is parametrized, diagnosed and operated by a server application running on a controller board inside the MicroTCA.4 chassis. This controller is a full featured single board computer running Ubuntu server 16.04.3 LTS. It is connected via PCI express bus (PCIe) to all LLRF FPGA cards (Fig 1).

A single LLRF server application provides access to about 500 process variables (PV) over Ethernet. The variables are mainly scalars and 16384 element arrays of 32 bit integer and double data types. Dependent on the accelerator operation mode they are updated at a rate of around 10 Hz.

ELBE accelerator

The ELBE LLRF controls phase and amplitude for seven cavities in total. Two of them are normal conducting buncher cavities the other are super conducting TESLA-type cavities.

At ELBE industrial control hard- and software is used, namely Siemens S7-300/400 programmable automation controllers (PLCs) on the field level and WinCC 7.3 SCADA on the human-machine interface (HMI) level. WinCC is not especially suitable to process and display large data arrays with decent update rates. For such tasks usually LabVIEW is used at ELBE.

In summary integrating the digital LLRF into the ELBE control system requires suppling data to 3 types of clients (see Fig. 1):

- an expert GUI for system commissioning, configuration and diagnosis;
- WinCC 7.3 HMI for general operation;
- ELBE S7 PLCs for exchanging status and control data.

INTEGRATION SCHEME

The LLRF server application is developed at DESY, Hamburg using ChimeraTK [2]. A major feature of this tool kit is the abstraction of device server development from middle layer control system communication by defining an adapter mechanism [3]. A device server for a specific control system is generated by compiling it with the respective control system adapter.

A middle layer (Ethernet) protocol had to be selected, that is supported by all commercial products used at the ELBE accelerator (Siemens PLCs, WinCC SCADA, LabVIEW). At the same time an open source stack of the protocol had to be available to be license compatible with ChimeraTK. OPC UA as an open, cross-platform communication standard has been identified as the most appropriate interface solution. The chair of distributed control engineering of Technical University Dresden supported the LLRF integration by development of an according adapter using the open source OPC UA stack Open62541 [4].

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA166

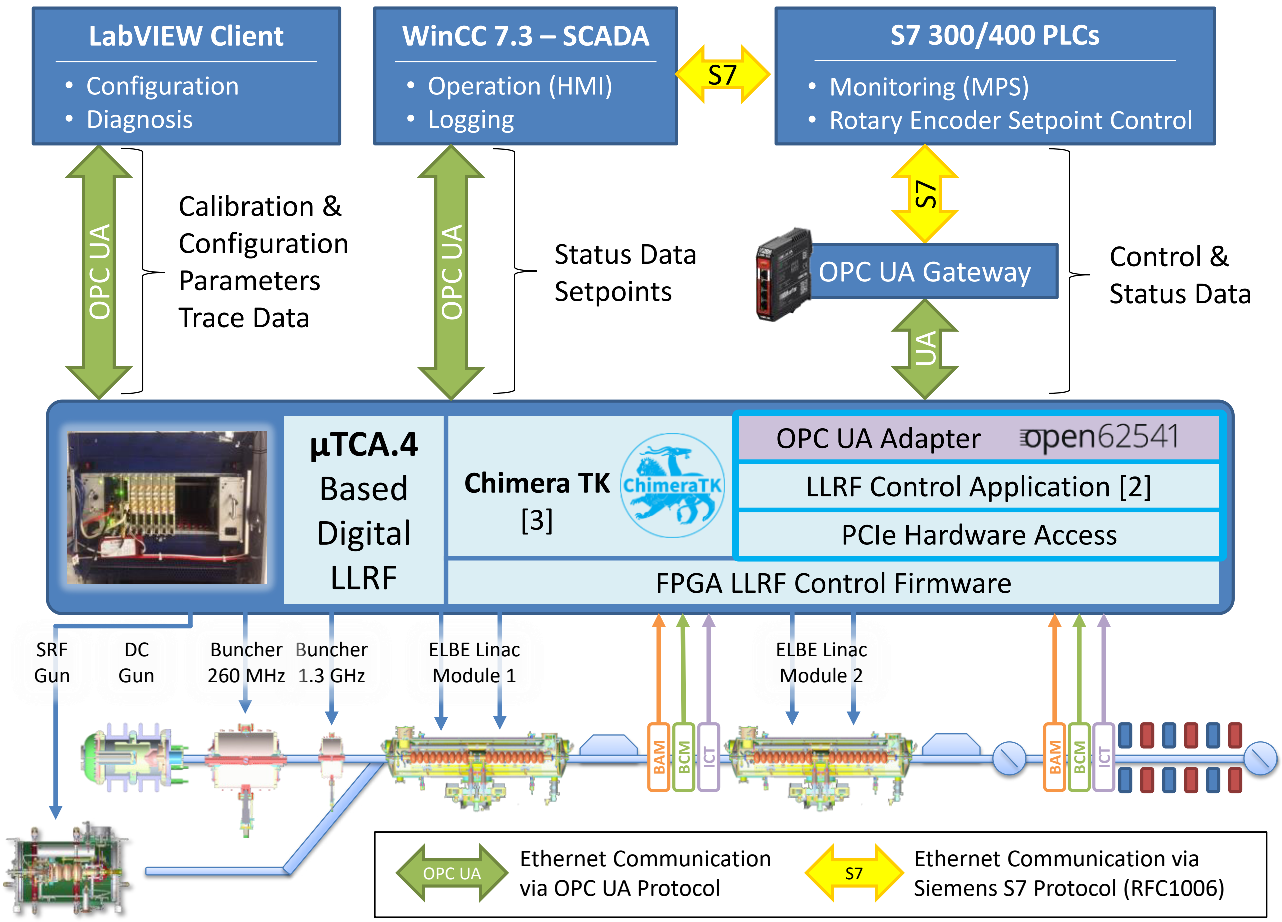

Figure 1: Digital LLRF ELBE control system integration scheme.

REALISATION

OPC UA Adapter

Part of the OPC UA Adapter is the variable mapping. It translates the process variables (PV) defined in the ChimeraTK application into the OPC UA server address space. According to the ChimeaTK data model, in addition to the value each PV holds further meta data elements: description, engineering unit, name and type.

The variable mapping, schematically shown in Figure 2, provides renaming, restructuring and the introduction of aliases to adapt the PV naming for the needs of different client applications. Additionally it is possible to add nodes with static information that is only available to system integrators. All variables are represented in a tree structure. The mapping and server configuration is controlled by an XML file that is read during server start-up. Hence there is one mapping configuration file for each server instance.

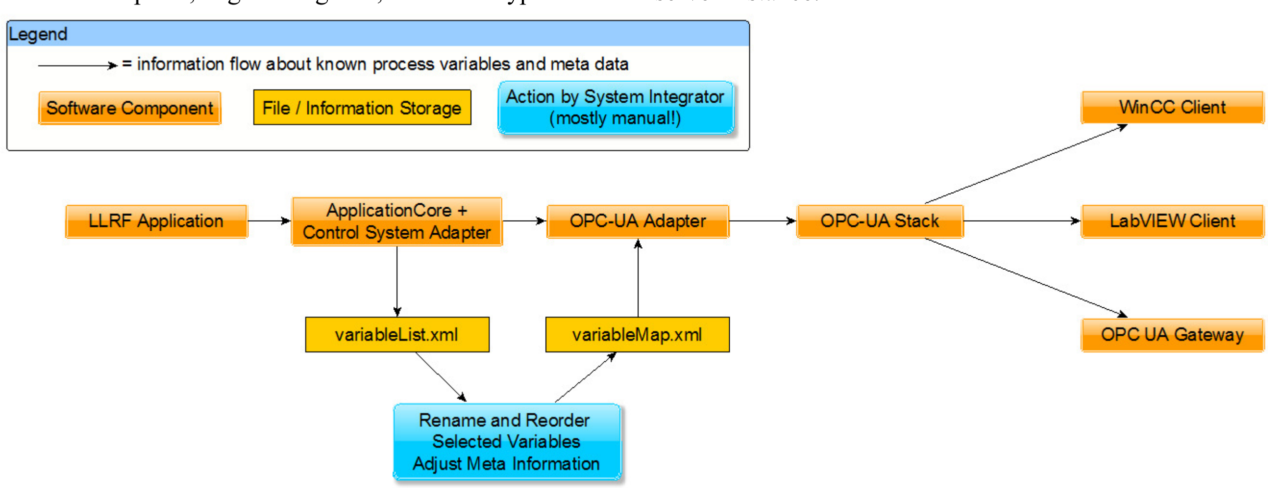

Figure 2: ChimeraTK OPC UA adapter - variable mapping scheme.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA166

LabVIEW Client Application

Whereas the Datalogging and Supervisory Control (DSC) Module of LabVIEW 2015 SP1 incorporates an OPC UA client and server API, no further functionality like an OPC UA server browser or the possibility to bind front panel controls of LabVIEW virtual instruments (VI) to OPC UA nodes is integrated in the LabVIEW IDE. Therefore creation and maintenance of HMI VIs with many I/O variables is time-consuming. Additionally the OPC UA communication leaves a big footprint in the block diagram of the VI.

Therefore a binding functionality was realized and a HMI tool was created using the LabVIEW Shortcut Menu Plug-In feature. The tool allows browsing OPC UA servers, selecting data nodes and inserting front panel controls with according OPC UA binding annotation at VI edit time (see Fig. 3). This binding information is read during panel start-up and used to establish the connection to the OPC UA variables.

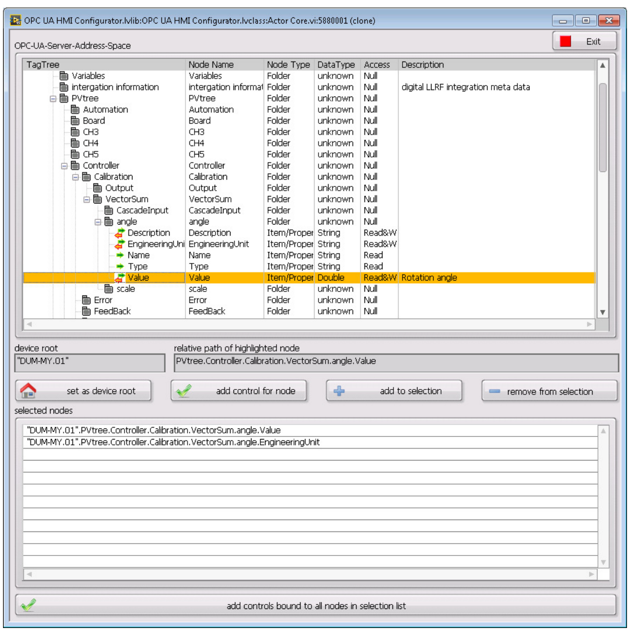

Figure 3: OPC UA HMI VI Tool.

For better code maintainability and reusability the implementation followed an object oriented design approach with the use of NI Actor Framework. All HMI panels are actors that can be called dynamically and run asynchronously as parallel processes.

Binding paths are relative to a selectable device root folder. Thus a designed panel can be used directly for all devices of the same type. The HMI actor simply has to be called with the correct server connection and device root path (see Fig 4).

Updates of indicators from OPC UA nodes and vice versa, is event triggered. The client application subscribes to the OPC UA server and registers variables as monitored items. The OPC UA server informs the client about data changes by raising data change events. These are used to write the new values into the corresponding controls.

The list of monitored items is dynamically adjusted to only receive updates of controls that are currently visible, thereby reducing network and application load.

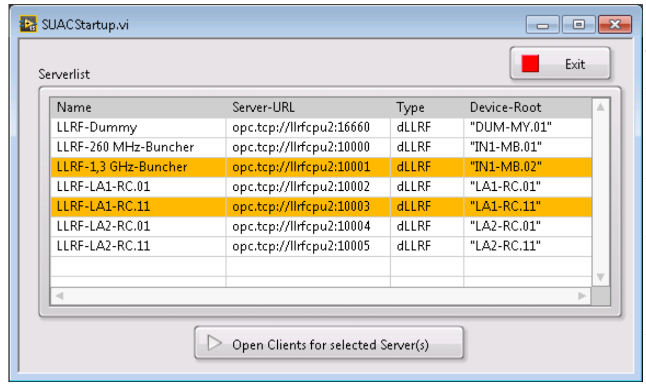

Figure 4: LLRF GUI start-up panel.

In the other direction value change events of front panel controls are used to write user input data into bound OPC UA nodes.

During commissioning of the digital LLRF with two ELBE cavities (one normal and one super conducting) in September 2017 this implementation was successfully tested. More than 10 graphs of arrays with 16384 elements each, were simultaneously updated with around 10 Hz. Figure 5 shows an example screenshot of the running application.

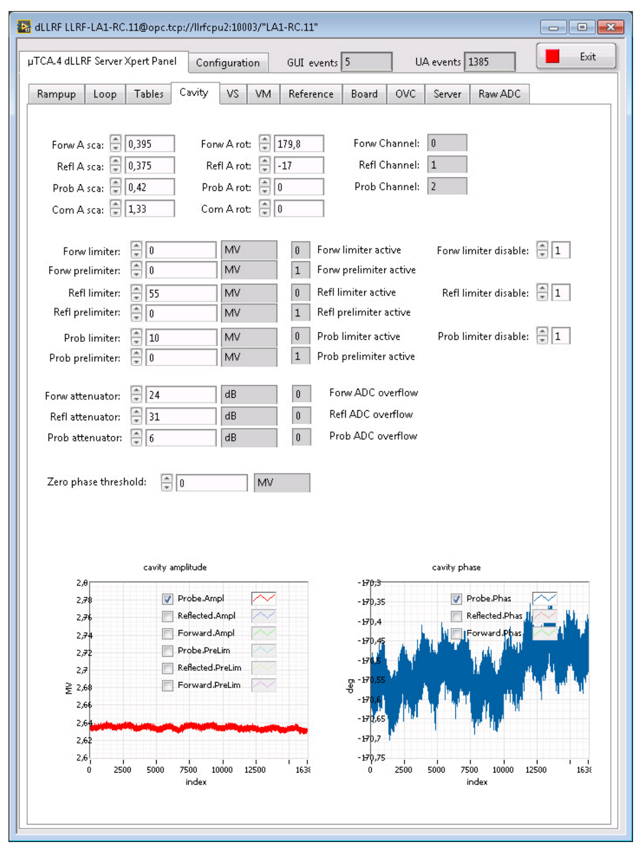

Figure 5: LLRF expert panel.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA166

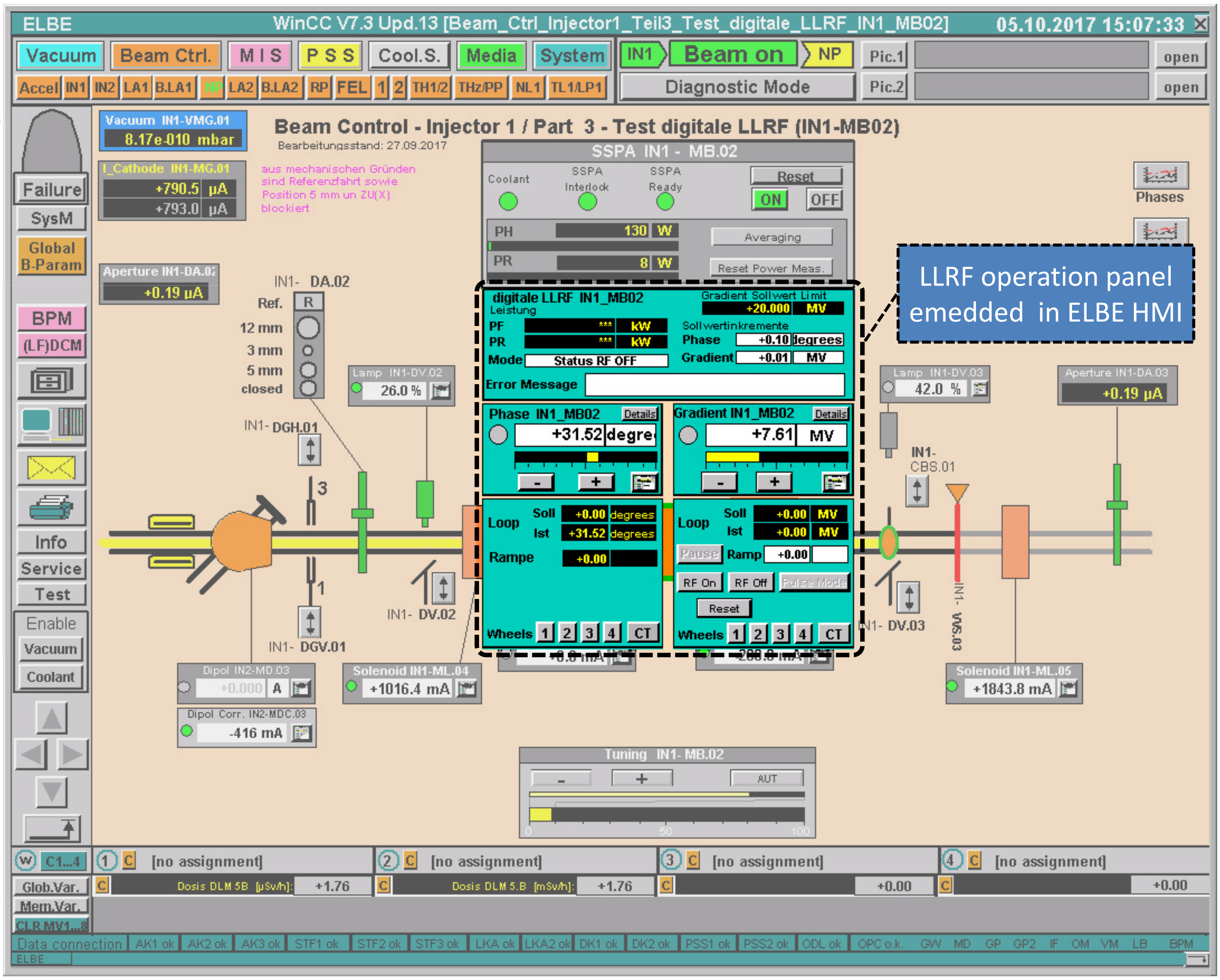

Figure 6: LLRF operation panel embedded in ELBE WinCC HMI.

PLC Data Connection

For several reasons a direct communication between the LLRF servers and the ELBE PLCs is necessary. First the machine protection system (MPS) has to be informed about the current LLRF operation mode. At the same time the MPS must be able to constrain LLRF operation mode changes. Another reason is the use of rotary controllers at ELBE. These devices allow for smooth setpoint adjustments by turning a controller wheel with a rotary encoder. They are directly connected to the PLCs.

Siemens S7-300/400 PLCs do not inherently support the OPC UA protocol. While there are OPC UA communication modules available for the S7-400 series, there are none for the S7 300 PLCs. The compact OPC UA gateway IBH Link UA by IBHsoftec [5] is a versatile solution for this problem. It is symbolically parametrized by means of the STEP7 PLC project and can act as OPC UA client and server. Once configured, it transfers data between OPC UA devices and S7 PLCs transparently.

WinCC 7.3 Operation Panel

The usual LLRF operation requires only a reduced set of variables. WinCC 7.3 natively includes an OPC UA Client channel. Therefore WinCC variables can easily be connected to OPC UA nodes. In Figure 6 the operator HMI for one of the ELBE bunchers is displayed.

CONCLUSION AND OUTLOOK

Within the scope of a commissioning in September 2017, all hard and software layers of the new digital LLRF for ELBE have been tested in conjunction for the first time. The system is now prepared for parallel operation with analogue LLRF. All components will be further refined on the basis of long-term operation experience.

REFERENCES

[1] MicroTCA.4, `http://mtca.desy.de/`
[2] M. Hierholzer *et al.*, “A facility-independent low-level RF server for MicroTCA.4-based systems”, ICALEPCS’17, Barcelona, Spain, Oct. 2017, poster THPHA136, this conference
[3] M. Killenberg *et al.*, “Abstracted Hardware and Middleware Access in Control Applications”, ICALEPCS’17, Barcelona,Spain, Oct. 2017, poster TUPHA178, this conference
[4] Open62541, `https://open62541.org/`
[5] IBH Link UA, `https://www.ibhsoftec.com`