HIGH LEVEL SOFTWARE STRUCTURE FOR THE EUROPEAN XFEL LLRF SYSTEM

Christian Schmidt*, Valeri Ayvazyan, Julien Branlard, Łukasz Butkowski, Olaf Hensler,
Martin Killenberg, Mathieu Omet, Sven Pfeiffer, Konrad Przygoda,
Holger Schlarb, DESY, Hamburg, Germany
Adam Piotrowski, FastLogic sp. z o.o., Łodz, Poland
Wojciech Cichalewski, Filip Makowski TUL-DMCS, Lodz, Poland

Abstract

The Low Level RF system for the European XFEL is controlling the accelerating RF fields in order to meet the specifications of the electron bunch parameters. A hardware platform based on the MicroTCA.4 standard has been chosen to realize a reliable, remotely maintainable and high performing integrated system. Fast data transfer and processing is done by field programmable gate arrays (FPGA) within the crate, controlled by a CPU via PCIe communication. In addition to the MicroTCA.4 system, the LLRF comprises external supporting modules also requiring control and monitoring software. In this paper the LLRF system high level software used in E-XFEL is presented. It is implemented as a semi-distributed architecture of front end server instances in combination with direct FPGA communication using fast optical links. Miscellaneous server tasks have to be executed, e.g. fast data acquisition and distribution, adaptation algorithms and updating controller parameters. Furthermore the inter-server data communication and integration within the control system environment as well as the interface to other subsystems is described.

INTRODUCTION

The Deutsches Elektronen-Synchrotron (DESY) in Hamburg is currently building the European X-ray Free Electron Laser (E-XFEL) [1]. This hard X-ray light source generates up to 27000 coherent laser pulses per second with a duration of less then 100 fs and a wavelength down to 0.05 nm. For this, electrons have to be accelerated to 17.5 GeV using a 2 km particle accelerator based on superconducting radio frequency technology. Precision regulation of the RF fields inside the accelerating cavities is essential to provide a highly reproducible and stable electron beam. RF field regulation is done by measuring the stored electromagnetic field inside the cavities. This information is further processed by the feedback controller to modulate the driving RF source, using a low level RF system. Detection and real-time processing are performed using most recent FPGA techniques. Performance increase demands a powerful and fast digital system, which was found with the Micro Telecommunications Computing Architecture (MicroTCA.4) [2]. Depending on the application either parallel processing for latency efficiency or pipelined processing of a large number of similar RF signals is aspired. In-tunnel installation with limited access requires a compact system with implied redundancy. Further remote access to status information, software upgrades and maintenance is ensured. Finally the modularity and scalability of this system guarantees to have later upgrades in order to meet demands within a lifetime of the machine and beyond. DESY currently is operating the Free Electron Laser (FLASH), which is a user facility of the same type as E-XFEL but at a significantly lower maximum electron energy of 1.2 GeV. The LLRF system for FLASH is equal to the one of E-XFEL, which allows for testing, developing and performance benchmarking in advance of the E-XFEL commissioning [3]. A picture of the FLASH installation can be found in Fig. 1.

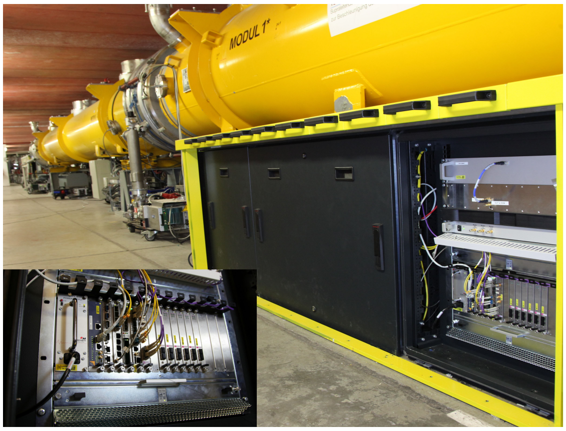

Figure 1: Installation of MicroTCA crates inside the FLASH tunnel underneath the accelerator module. The radiation shielding (yellow cabinet) protects the electronic rack inside. The insert shows the MicroTCA crate installed in this rack.

LLRF SYSTEM LAYOUT

The present LLRF system for a single E-XFEL RF station controlling 32 cavities is given in Fig. 2. LLRF systems with a different quantity of cavities are build as a subset of this configuration. Machine operation is performed in a pulsed mode. The duty cycle, i.e. the repetition rate (10 Hz) of the RF pulse to the RF pulse length (about 1 ms) is about 1 %. The impact to the RF regulation is such that within in a regular mode of unchanged operating conditions, i.e. stable working point consecutive pulses are similar, however the initial conditions vary as well as long term changes occur. In

[*] christian.schmidt@desy.de

ISBN 978-3-95450-148-9

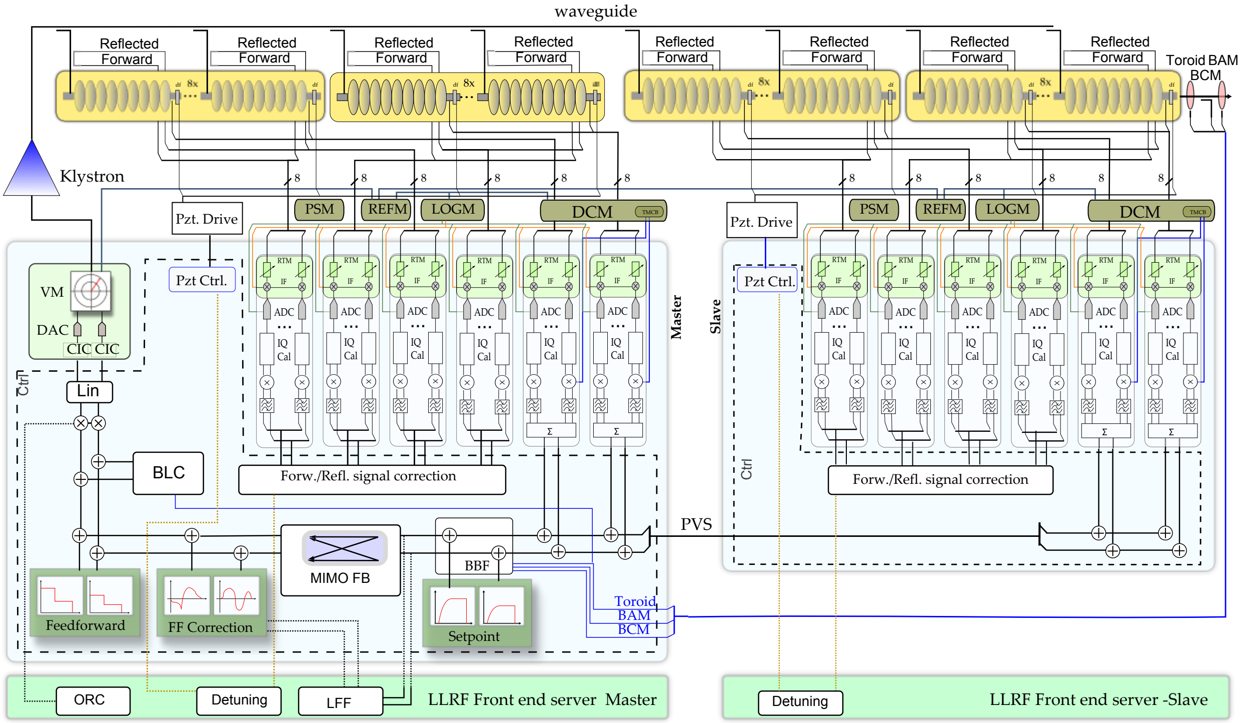

Figure 2: Basic layout of the LLRF station for the master and slave control used in E-XFEL. Signal processing path as well as communication links are presented. Communication between the master and slave system is done using optical links, as well as for connecting further subsystems to the LLRF (marked as BAM, BCM and Toroid).

order to optimize the field regulation a temporal separation of feedback is applied in the following way:

- μs scale: intra pulse feedback and digital signal processing on the FPGA
- *ms* scale: pulse to pulse adaptation, removing repetitive control errors, processed in the CPU
- *s* scale: long term drift compensation and slow feedback mechanism by external modules or feedbacks

Intra Pulse Data Processing

Fast regulation steps are processed within the FPGA. Furthermore the fast regulation loop is separated into a preprocessing step, running on a different FPGA than the main controller application, which collects data from several preprocessing boards using low latency communication links in the backplane of the MicroTCA.4 crate. The 16 cavity data acquisition and preprocessing section is identical for the master and the slave subsystem. The main controller sums up the two partial vector sums (PVS), resulting from the contribution of cryomodule (CM) 1 and 2 on the master LLRF system and CM 3 and 4 on the slave LLRF system. This total vector sum is processed within a multiple input, multiple output (MIMO) feedback controller [4] and added to the feed-forward signal in order to to generate the klystron drive. Further beam based information transferred through optical links are processed in order to improve the regulation performance (BBF) or compensate effects like beam loading (BLC).

Pulse to Pulse Adaption

Repetitive control errors such as generated by cavity detuning, are rather compensated by some additional feed-forward input instead of the feedback. An model based iterative learning feed-forward algorithm (LFF) processing data from pulse to pulse on the CPU is used to play some correction signals in order to relief the action required from the feedback. Changes in the actuator chain are visible in offsets of the correction signals and are compensated by automatic scaling and rotating the output vector (ORC).

Long Term Drift Suppression

Long term drifts, especially in the measurement chain are currently the most critical part in the regulation loop. There exist different approaches to compensate for this, e.g. reference signal tracking or passive methods like ensuring stable environmental conditions. Here a special drift compensation module (DCM) is used which measures a priori the RF pulse a reference pulse and corrects the RF signals for each channel accordingly. Further beam based measurements are used to quantify the control error with respect to the electron beam and slowly adapt the operating conditions by changing LLRF setpoints.

The combination of these feedback mechanisms is the basis to meet the given RF field regulation requirements of

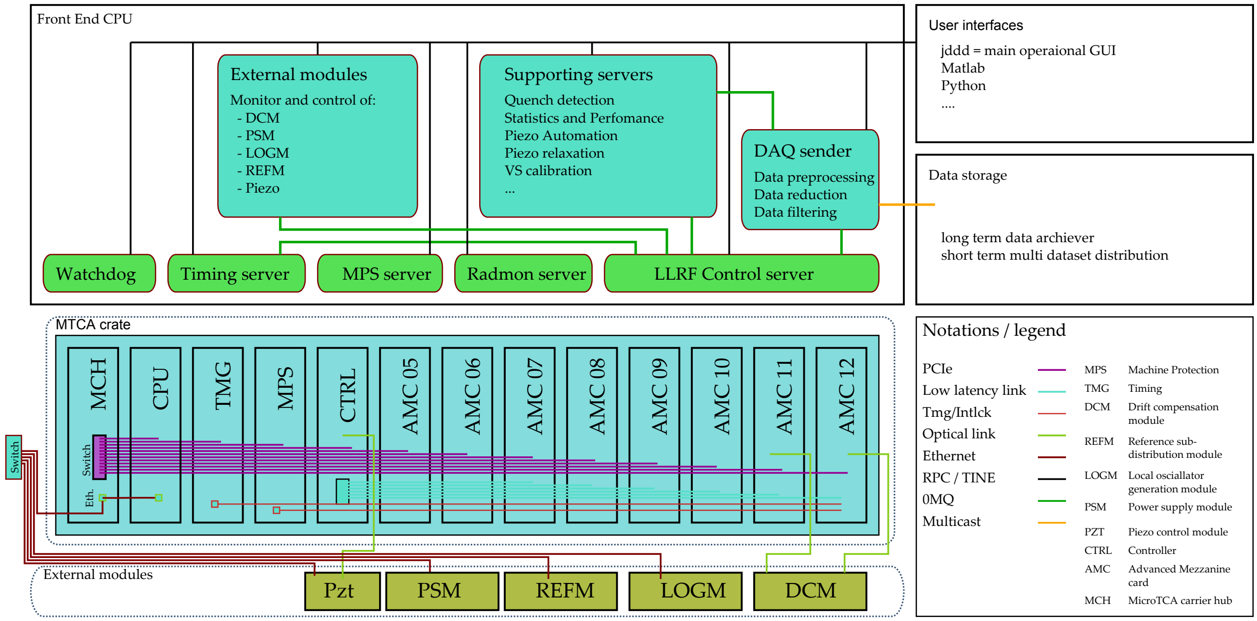

Figure 3: Front end server structure and communication links within the MicroTCA.4 LLRF system.

0.01 % and 0.01 deg in amplitude and phase. Control and communication with the firmware is based on the LLRF front end server.

ARCHITECTURE OF THE LLRF FRONT END SERVERS

Having a pulsed mode of operation requires to synchronize all subsystems to a frequent trigger. This is realized by a timing distribution system which triggers all subcomponents within the crate at a given time. Further the servers running on the front end CPU are triggered by the same mechanism. This synchronization puts the subtasks within one RF pulse in line which is: starting the RF pulse, processing data within the FPGA, transferring data to the CPU, processing data and further sub-distribute, update control variable withsin the FPGA. Communication between the FPGA and the CPU is done by direct memory access over a PCIe link [5]. Fixed point representation from the FPGA is converted to physical meaningful data before further distributing the data to graphical presentation, other processing tasks or data storage. Inter process communication on the same CPU is realized using the ZeroMQ communication protocol [6]. Java based DOOCS is in charge of the graphical user interface [7]. Data communication to the GUI is done by RPC or TINE calls, which is further extended to use third party display and processing programs like Matlab. A sketch of the inter-server communication topology, inter-crate communication links and user interfaces can be found in Fig. 3.

The LLRF control server itself can be seen as the basic interface to the MicroTCA.4 hardware such that communication to front end registers is being controlled by this particular device. Data acquisition and pre-processing is a high priority task and should be treated as quasi real time capable. Therefore other tasks displayed as supporting servers will have a lower priority since the system is still operable even without having them running. The DAQ sending device is collecting data from the LLRF control server before transmitting them further to a data storage. Here the data from all subsystems is synchronized and stored for later analysis or directly used in higher level automation routines. This data storage is extremely helpful for post mortem analysis of system malfunctions.

Decoupling of the basic LLRF control server from other supporting servers turned out to be rather helpful in terms of maintenance and further developments. Having the front end server optimized at a given state it usually requires only minor adaptations during the running process. Further changes in the functionality of the server should be ideally transparent to other clients, which can be done except of interface changes.

Extensions and additional status and processing information is likely often adapted to new requirements. Maintaining such systems does not directly influence the main controller and also bugs usually induced by new developments are transparent to third party clients. Having defined clear interfaces allows to have a modular mechanism. For example the same LLRF front end server can be used for different subsystems even if additional supporting servers are not required or wanted. This reduces the overall load on the processing unit by reducing the number of subtasks to the required minimum. The goal of having a highly modular system for the MicroTCA.4 hardware and its corresponding firmware also reflects on the software architecture.

Basic Front End Servers

The basic idea for a front end server is to have one instance capable to communicate with the hardware device installed in the system. For example comparing Fig. 3, the devices of timing control (TMG), machine protection (MPS) are single server - single device pairs. For the LLRF control server there are in total 7 devices controlled by one server which can be seen as one combined system. This comparison reflects the complexity and large amount of data processing taking place. Reducing the functionality of this server to basic functionalities is a consequence to distribute tasks in an effective way.

Supporting Servers

The subcategory of supporting servers contains several functionalities separated into frequently running instances and user triggered applications. As an example the routine of automatic piezo tuning should be described here. In order to minimize the cavity detuning, piezo actuators are used to play a mechanical stimulus to compensate to RF field induced deformations of the cavity. This is done by computation of the actual detuning within the statistics and performance server. Necessary data for this computation is send from the LLRF control server as described above. After the detuning parameters are computed, they are further send to the piezo automation server which computes the waveform to be played. This is transferred back to the LLRF front end server which sends this waveform to the firmware registers as a control command sequence for the next pulse. A different example can be found with the VS calibration instance, which is triggered unfrequent by user requests in order to recalibrate the measurement chain of the LLRF system. Permanent operation of this server is not possible since the calibration method is invasive to machine operation and usually requires a special setup.

External Device Servers

Within the LLRF system there exist next to the MicroTCA.4 crate supporting components which also require to be controlled and monitored. The setup varies with the different locations the system is implemented. Each of this system requires a server, however there are many functionalities within which can be transferred in between. All these servers share special classes which can be plugged together in a modular way. Communication is done over ethernet which would basically allow to run the instances also on a different CPU if high load on the front end CPU would require this.

CONCLUSION AND OUTLOOK

In this paper the MicroTCA.4 based LLRF control system in view of the LLRF software structure for the E-XFEL has been presented. Basic layout ideas and structuring as well as communication paths are discussed. This architecture is currently operating in the FLASH facility, meeting the required goals in terms of RF field stability. However the basic functionalities are successfully running, continues developments are ongoing which further improve the overall system performance, as well as the maintainability and reliability.

Currently all acceleration modules being installed in the E-XFEL are tested and characterized in advance. Part of this program is done by the LLRF system since gained information will be used for later automation routines. Storage of this data is done by a database which requires automatic acquisition and load to the systems being installed in the E-XFEL. Furthermore special automation routines developed for the tests are directly transferable to E-XFEL and also to FLASH.

Fast error tracking helps to minimize machine downtime. Therefore status and health information is monitored for all subsystems, however automatic processing this information is essential to support operators with the key information instead of a large amount of error notifications. Systematic failure analysis methods are currently under development.

Finally the LLRF system containing hardware, firmware and software are going to be used in other facilities which might use different control systems. To integrate this in a different environment, a control system independent server is developed, which decouples functionalities of the server from requirements of the control system [8]. All these modifications demand restructuring work, however the modularity of the systems as described in this paper helps to minimize the effort and reduces the risk of system malfunctions.

REFERENCES

[1] *The European X-Ray Free Electron Laser Technical Design Report*, `http://xfel.desy.de`

[2] MicroTCA$^{®}$ is a trademark of PICMG, MicroTCA.4 specifications: `http://www.picmg.org`

[3] J. Branlard *et al.*,*The European XFEL LLRF System*, Proceedings of the 3th International Particle Accelerator Conference 2012, New-Orleans, USA, 2012.

[4] C. Schmidt *et al.*, *Precision Regulation of RF Fields with MIMO Controllers and Cavity-based Notch Filters*, Proceedings of the XXVI International Linear Accelerator Conference 2012, Tel-Aviv, Isreal, 2012.

[5] M. Killenberg et al., “Drivers and Software for MicroTCA.4”, WEPGF015, *These Proceedings*, ICALEPCS’15, Melbourne, Australia (2015).

[6] http://zeromq.org

[7] E. Sombrowski, R. Kammering, K.R. Rehlich , *A HTML5 Web Interface for JAVA DOOCS Data Display*,*These Proceedings*, ICALEPCS’15, Melbourne, Australia (2015).

[8] M. Killenberg et al., “Integrating control applications into different control systems”, TUD3O05, *These Proceedings*, ICALEPCS’15, Melbourne, Australia (2015).