CONTROL OF THE LOW LEVEL RF SYSTEM OF THE LARGE HADRON COLLIDER

A. Butterworth[1], J. Molendijk[1], R. Sorokoletov[2], F. Weierud[1]

[1]CERN, Geneva, Switzerland, [2]JINR, Dubna, Russia

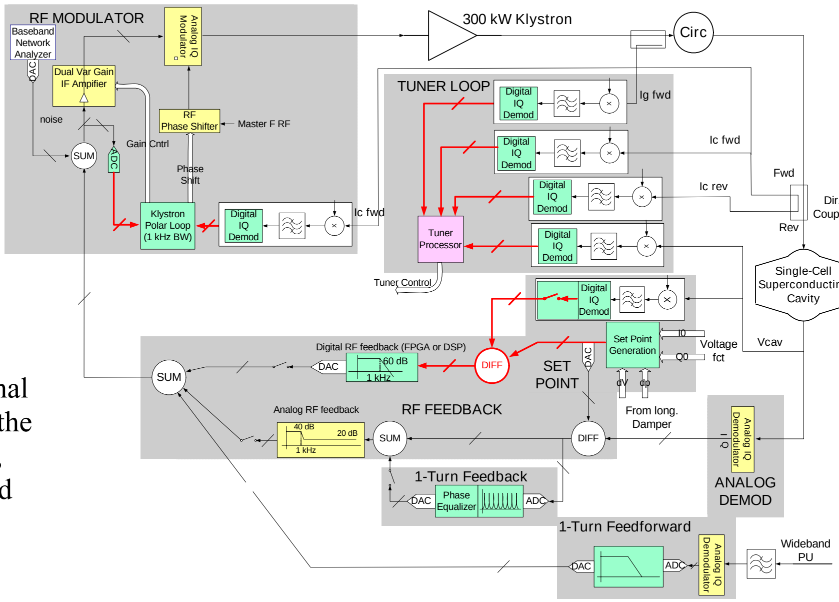
ABSTRACT

The radiofrequency (RF) acceleration system of the Large Hadron Collider (LHC) comprises 8 superconducting 400MHz cavities in each of the two LHC rings, with each cavity driven by a klystron amplifier. The phase and radial position of the beam and the accelerating voltage and phase in the cavities are controlled by a complex system of feedbacks collectively known as the low-level RF (LLRF) system. The LLRF system is implemented in mixed analogue and digital hardware modules using a custom form factor based on the VMEbus platform. The control system, from the crate controller CPU and timing modules, through the front-end software and communications middleware to the application software interface, is implemented using tools and components provided by the CERN accelerator controls group, such as the new Front-End Software Architecture (FESA). This paper describes the controls architecture of the LLRF system and discusses the hardware and software choices made in its implementation.

OVERVIEW OF THE LLRF SYSTEM

Introduction

The LLRF system comprises a number of feedback loops around the accelerating cavities and the beam [1]. The *cavity controller* (Figure 1) controls the phase and amplitude of the cavity voltage, minimising disturbances coming from high-voltage supply droop and ripples, and from transient beam loading. It also includes a tuner feedback loop to maintain the cavity in tune. The *beam control* system controls the phase and radial position of the circulating beam and the RF synchronisation with the other accelerators in the injector chain.



Analogue 🟨
FPGA 🟩
DSP 🟪

Figure 1: Functional block diagram of the Cavity Controller, showing the mixed analogue/digital implementation.

Hardware implementation

The LLRF electronics is implemented as a hybrid analogue/digital system composed of high-frequency analogue components and digital signal processing performed in Field Programmable Gate

Arrays (FPGA) and Digital Signal Processors (DSP). The 400 MHz RF signals are down-converted in an analogue RF front-end to an intermediate frequency of 20 MHz before being digitised by ADCs running at a sample rate of 80 MS/s. The data streams are then split into in-phase and quadrature signals (I/Q demodulation) at 40 MS/s for processing in the FPGAs.

The extensive use of digital processing reflects the current trend in LLRF system design and represents a departure from the traditional all-analogue approach used until recently at CERN.

LLRF CRATE AND MODULES

After considering several other options, it was decided to base the LLRF hardware modules and crates on the VMEbus standard. However, the 160mm depth of a standard VME board is too small to accommodate the analogue RF electronics with their associated shielding, and there are too few user-defined pins, so a special form factor was defined with 6U height and a depth of 220mm. The modules have two 96-pin backplane connectors, with the upper (J1) connector carrying the VME A24/D16 interface, and the lower (J2) connector used for routing of high-speed serial data links, clocks and trigger lines over a private backplane. The high-stability linear power supplies required for the analogue RF electronics are also routed over the J2 connector.

The LLRF-specific crate is shown in Figure 2. The right-hand 15 slots are reserved for custom modules with a custom P2 backplane, while the left-hand 5 slots are equipped with a short VME P2 backplane, enabling it to accept standard VME32 modules. Slot 0 houses the PowerPC CPU board, and the remaining 4 slots are available for timing modules or other standard boards.

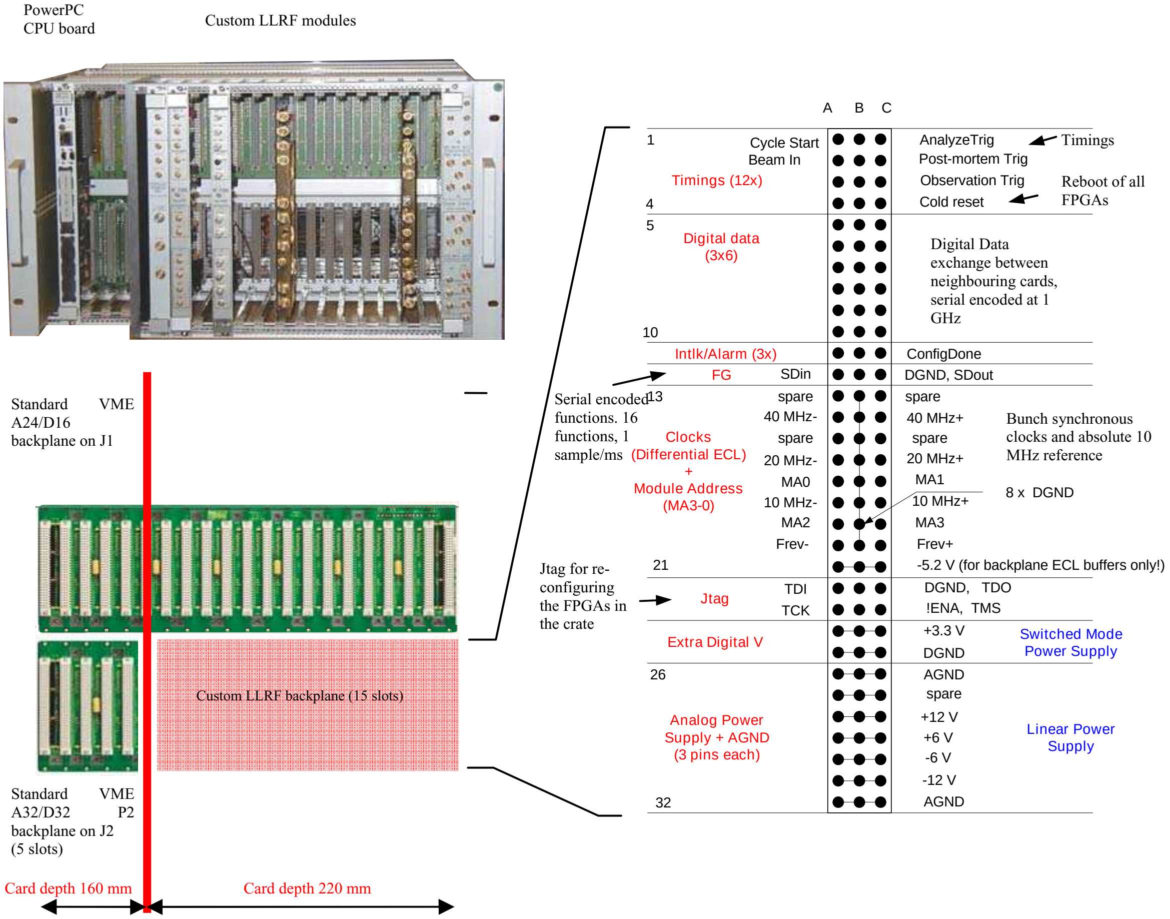

PowerPC CPU board
Custom LLRF modules
Standard VME A24/D16 backplane on J1
Custom LLRF backplane (15 slots)
Standard VME A32/D32 P2 backplane on J2 (5 slots)
Card depth 160 mm
Card depth 220 mm
A B C
1
4
Timings (12x)
Cycle Start
Beam In
AnalyzeTrig
Post-mortem Trig
Observation Trig
Cold reset
Timings
Reboot of all FPGAs
5
10
Digital data (3x6)
Digital Data exchange between neighbouring cards, serial encoded at 1 GHz
Intlk/Alarm (3x)
ConfigDone
FG
SDin
DGND, SDout
Serial encoded functions. 16 functions, 1 sample/ms
13
21
Clocks (Differential ECL) + Module Address (MA3-0)
spare
40 MHz-
spare
20 MHz-
MA0
10 MHz-
MA2
Frev-
spare
40 MHz+
spare
20 MHz+
MA1
10 MHz+
MA3
Frev+
-5.2 V (for backplane ECL buffers only!)
Bunch synchronous clocks and absolute 10 MHz reference
8 x DGND
Jtag for re-configuring the FPGAs in the crate
Jtag
TDI
TCK
DGND, TDO
!ENA, TMS
Extra Digital V
+3.3 V
DGND
Switched Mode Power Supply
26
32
Analog Power Supply + AGND (3 pins each)
AGND
spare
+12 V
+6 V
-6 V
-12 V
AGND
Linear Power Supply

Figure 2: The LLRF-specific VME crate and custom backplane.

ON-BOARD DIAGNOSTICS

Since a large part of the LLRF system is now implemented in digital hardware, the traditional analogue test points are no longer available. It is therefore essential to design embedded diagnostics into the LLRF hardware [2]. Data from "virtual test points" in the digital signal processing chain are acquired into on-board circular memory buffers at up to 40 MS/s. The buffers can be frozen by a hardware or software trigger and read out over the VME bus.

The example shown in Figure 3 was acquired during LLRF tests in August 2005 on a production LHC cavity. The plots were taken from the on-board diagnostic buffers of the tuner control board, and show the operation of the mechanical cavity tuner. When the tuner phase error signal strays outside the dead-band, the tuner stepper motor moves one step, which corresponds to a resonant frequency adjustment of about 25 Hz. The second plot shows on an expanded scale the oscillating error signal due to the mechanical resonance of the cavity at around 143 Hz.

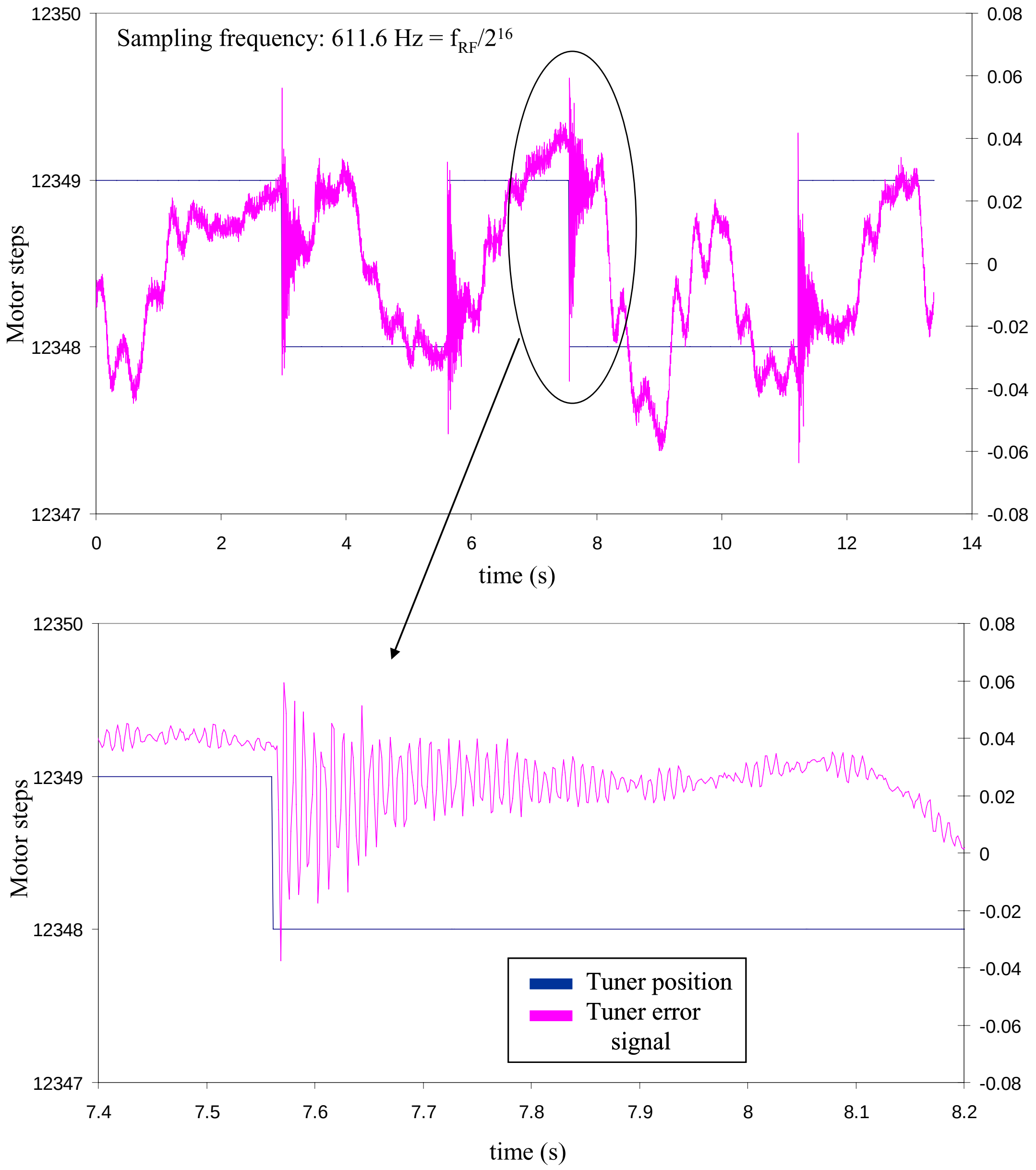


Figure 3: Example of data acquisition using on-board diagnostic memory buffers.

SOFTWARE ARCHITECTURE

“Standard” architecture for VME systems

The LLRF control system interface (Figure 4) follows the “standard” software architecture defined by the Accelerator & Beams controls group, and uses its associated tools and components.

- *DriverGen* is a utility for generating LynxOS device drivers for VME hardware access [3]. It uses a hardware description (memory map) entered in the controls hardware configuration database.
- *FESA* (Front End Software Architecture) is a framework for Front-End equipment control software [4]. It schedules real-time software actions according to machine timing events from the accelerator timing system, and handles data persistence and cycle-to-cycle multiplexing. It is also integrated with the controls configuration database to handle automatic deployment of software on the front-end computers.
- The *Controls Middleware* (CMW) is a “software data bus” allowing remote access to equipment through a device/property model [5]. Java client applications are the “standard”, but a C++ client package is also available.

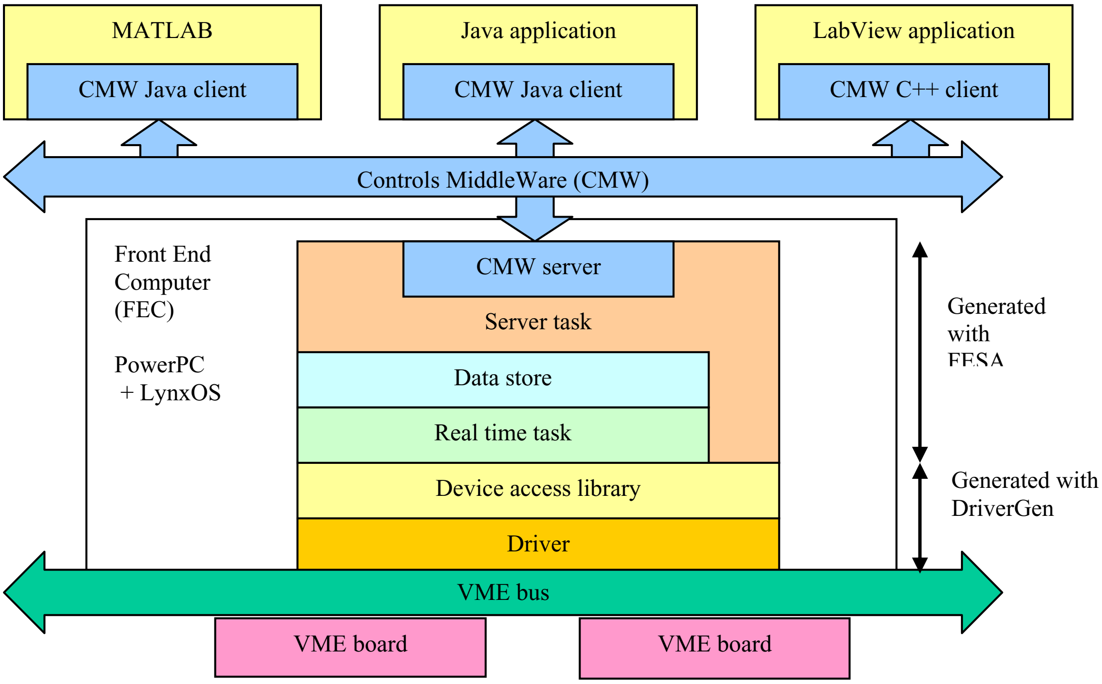
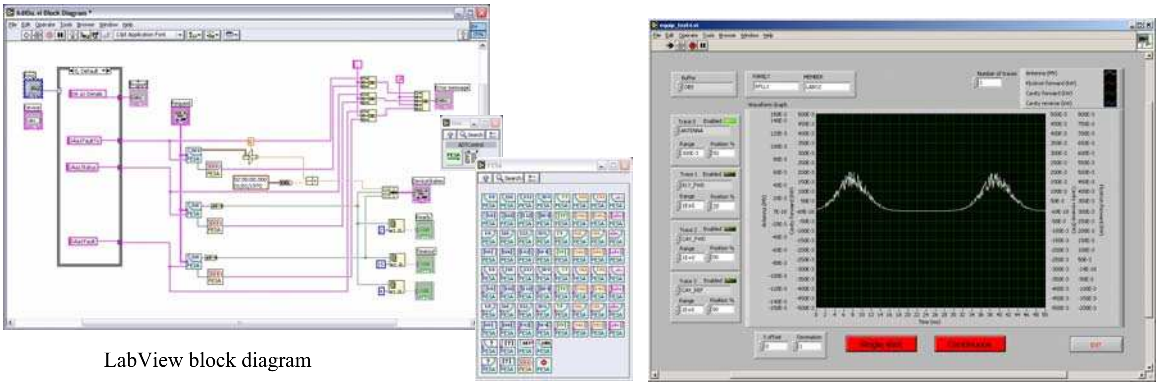


Figure 4: “Standard” software architecture for VME systems.

Application software interface

The LLRF system is accessible via the CMW middleware from all standard Java control room applications, and from any external software package which supports an interface to Java, such as MATLAB and Mathematica.

A wrapper library [6] has also been written for the CMW C++ client package permitting access to CMW-enabled devices from LabView via a palette of custom Virtual Instrument components (Figure 5).



LabView block diagram

CMW/FESA palette

LabView applitn n

Figure 5: Use of CMW client interface package for LabView.

CONCLUSION

The LHC LLRF system is a highly complex system, which makes extensive use of digital processing performed in powerful programmable logic devices. This has allowed the inclusion of sophisticated on-board diagnostics, essential for initial commissioning and for monitoring of the system performance. The choices of controls hardware and software architectures have been made and implementation is underway. The use throughout of standard tools provided by the Accelerators and Beams controls group allows remote monitoring and control through the control system via standard operations software and also the use of powerful specialist tools such as LabView and MATLAB.

REFERENCES

[1] P. Baudrenghien, “The LHC Low Level RF”, LLRF05, CERN, Geneva, 2005

[2] J. Molendijk, “Complex digital circuit design for LHC Low Level RF”, LLRF05, CERN, Geneva, 2005

[3] A. Gagnaire, Y. Georgievskiy, “Driver Generation Tools”, http://ab-co-fc.web.cern.ch/ab-co-fc/drivers/Presentation-MoulinetteDEvelopement-26092003.ppt, CERN, Geneva, 2003

[4] M. Arruat, S. Jackson, J-L. Nougaret, M. Peryt, “Equipment Software Modeling for Accelerator Controls”, ICALEPCS2005, Geneva, 2005

[5] K. Kostro, J. Andersson, F. Di Maio, S. Jensen, N. Trofimov, “The Controls Middleware (CMW) at CERN, Status and Usage”, ICALEPCS2003, Gyeongju, Korea, 2003

[6] R. Sorokoletov, http://ab-rf-cs-cmwwrapper.web.cern.ch