INPUT OUTPUT CONTROLLER OF DIGITAL LOW LEVEL RF SYSTEM IN NSRR

Zong-Kai Liu †, Fu-Yu Chang, Lung-Hai Chang, Mei-Hsia Chang, Ling-Jhen Chen, Fu-Tsai Chung, Ming-Chyuan Lin, Chih-Hung Lo, Chi-Lin Tsai, Chaoen Wang, Meng-Shu Yeh andTsung-Chi Yu, NSRRC, Hsinchu, Taiwan

Abstract

Low Level Radio Frequency (LLRF) systems operating at NSRRC are based on analog technology and are used both at the Taiwan Light Source and the Taiwan Photon Source. In order to have better RF field stability, a new digital LLRF system based on Field Programmable Gate Array (FPGA) was developed. A card-sized single-board computer is used as the input/output controller of the digital LLRF system and its design and implementation with EPICS applications are reported here.

INTRODUCTION

There are two light sources located at NSRRC, one is the Taiwan Light Source (TLS) and the other is the Taiwan Photon Source (TPS) [1] both using analog LLRF systems. They are designed to have a stability of 1% in amplitude and 1° in phase for the accelerating cavity voltage [2]. In order to improve the performance of the LLRF system, a new digital LLRF (D-LLRF) system, based on Field Programmable Gate Array (FPGA), was developed in house. Figure 1 shows the schematic diagram of the D-LLRF system. Preliminary results from performance tests show that the RMS amplitude and phase errors of the accelerating voltage measured at 280 kV for the TLS booster with a D-LLRF system are about 0.18% and 0.13 degree, respectively. The 60 Hz noise sideband and its harmonics can be suppressed down to -70 dBc. Details of the design, the implementation and test results of the D-LLRF system can be found in [3].

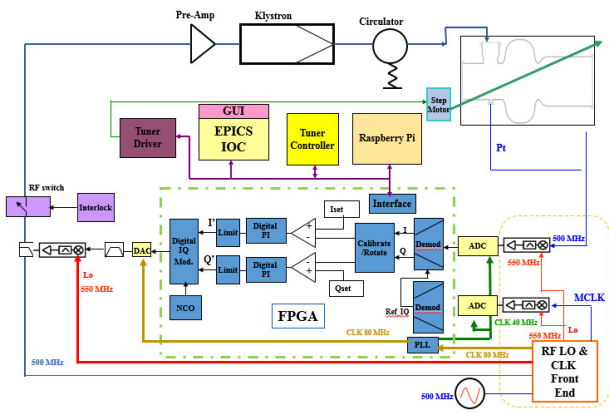

Figure 1: Schematic diagram of the D-LLRF system.

Raspberry Pi (RPi) is an educational used card-sized single-board computer [4]. Its applications cover a wide range of industrial control, telecommunication, latest laptops and mini-PCs, etc. Some hardware specifications are shown in Table 1 [4].

Table 1: Hardware specifications of the RPi 2 mode B

Raspberry Pi 2 Model B	
SoC	Broadcom BCM2836
CPU	900 MHz 32-bit quad-core ARM Cortex-A7
Memory	1 GB
Storage	SD card
I/O	GPIO, UART, I²C bus, SPI bus with two chip selects, I²S bus, +3.3 V, +5 V, GND.
OS	Raspbian, Ubuntu, Windows 10 IOT, …

The Raspberry Pi 2 model B was selected to be used as the input/output controller in the D-LLRF system, which provides communication functions with the FPGA. It also allows the controller to be hosted and operated from a local or remote PC. Figure 2 is a photo of the D-LLRF controller hardware with the Raspberry Pi inside. The design and implementations of the input/output controller with EPICS applications are discussed in this paper.

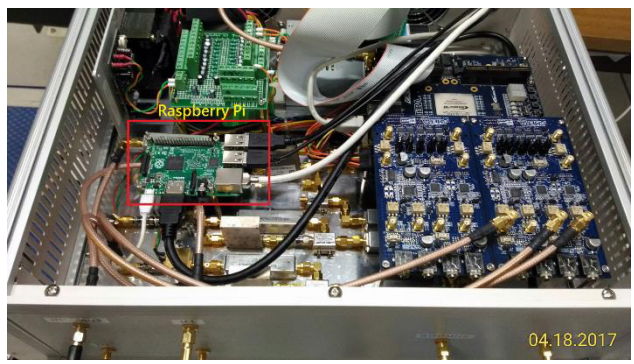

Figure 2: Photo of the D-LLRF controller hardware with the Raspberry Pi inside.

RASPBERRY PI AS AN IO CONTROLLER

The RPi has 40 GPIO pins, which are selected to be used for communications with the FPGA. Except for the ground and DC voltage output pins, there are a total of 26 pins that can be used: 2 pins are selected for read/write enable signals, 7 pins are for memory addressing and 14 pins are for data signals. In total 128 addresses with 14-bit data communications are available. Due to the limited number of

† liu.zk@nsrrc.org.tw

Copyright © 2017 CC-BY-3.0 and by the respective authors

ISBN 978-3-95450-182-3

GPIO pins, the 14 GPIO data pins need to be bi-directional so that they can be used for input while reading and can be used for output while writing. The maximum GPIO I/O speed with a python library RPi.GPIO is about 70 kHz [5]. This is enough for the basic command function of the D-LLRF controller.

Software Architecture

The software architecture for communication is shown in Fig. 3 and the application programs are developed with python. User commands can be sent to the FPGA for local operation by a local GUI or for remote operation via EPICS interface. The GPIO pins, which are controlled by read/write functions, can be set to either input or output mode depending on the digital signal being reading from the FPGA or setting to the FPGA. A bi-directional IO module is built into the FPGA to control the data flow and the direction is determined by the enable signal and the memory address. Random Access Memories (RAMs) are used for data storage for the machine status, operation parameters and command digits. All data used in the FPGA are updated from two register modules: one is for writing and the other for reading. There are some RAMs for the function of circular buffers, which are transient recorders used for analysis of trip events and transient effects. As shown in Fig. 3, there are three data streams controlled by a bi-directional IO module: one is for writing to the FPGA (red), one is for reading from the FPGA (white) and one is for reading from the circular buffers (yellow).

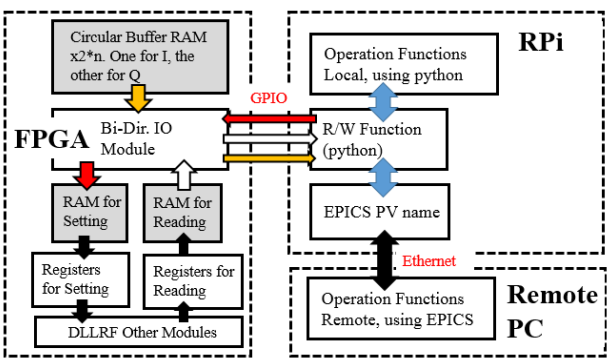

Figure 3: Architecture of the communications software.

Operating Functions

Figure 4 shows the GUI for the local operation which is developed by using the python thinker package. The operation functions are listed below:

- Mode selections: function for switching between off mode (RF off), tune mode (feed forward operation) and operation mode (feedback operation), as well as selection on whether it is in the ramping (used booster) or in DC mode of operation.
- Operation parameter setting: enable set point of amplitude and phase for the gap voltage in both tune and operation mode.
- Remote or local control selection.
- PID parameter setting.
- Gap voltage calibration: set the coefficient to convert digital values to real gap voltages.
- Auto monitoring: functions as monitor for important signals, such as transmit power, forward and reflected power, as well as a debug signal which can be selected by the debug channel selector.
- Circular buffer: transient recorders used for analysis of trip events, transient effects and RMS error estimates. Figure 5 shows the amplitude and phase RMS error estimates for the gap voltage at 280 kV for the TLS booster with an error tolerance of the D-LLRF system of about 0.18% and 0.13 degree, respectively.
- Feed forward table: provides an arbitrary waveform on top of the output signal. It can be used for tuning PID parameters or for RF conditioning.

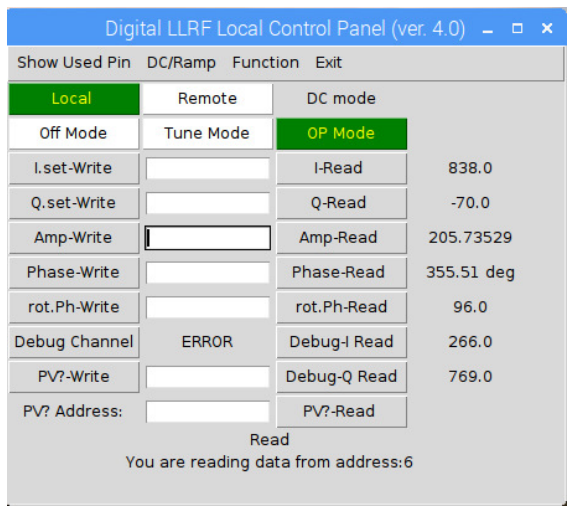

Figure 4: Beta version of the GUI for the local operation.

EPICS Applications

For remote operation, an EPICS (Experimental Physics and Industrial Control System) interface is used. EPICS is a set of open source software tools, libraries and applications developed collaboratively and used worldwide to create distributed soft real-time control systems for scientific instruments such as particle accelerators [6]. Figure 6 shows schematically the function architecture for remote operation. EPICS base and modules are necessary to set up a Linux based server to build an EPICS soft IOC. Because python is selected for application developments, PyEpics [7] is also needed to be installed on the RPi. Based on EPICS PV (Process Variable) channel access, RPi can access these PVs and get all the parameters, as well as being able to update PV values. The RPi, the EPICS soft IOC server and other EPICS IOCs are all on the same EPICS network. Other EPICS PVs from additional IOCs can be used for different applications on the RPi as well. The soft IOC server is on the public network and users from a remote control PC can access these EPICS PVs and develop applications for remote control.

Copyright © 2017 CC-BY-3.0 and by the respective authors

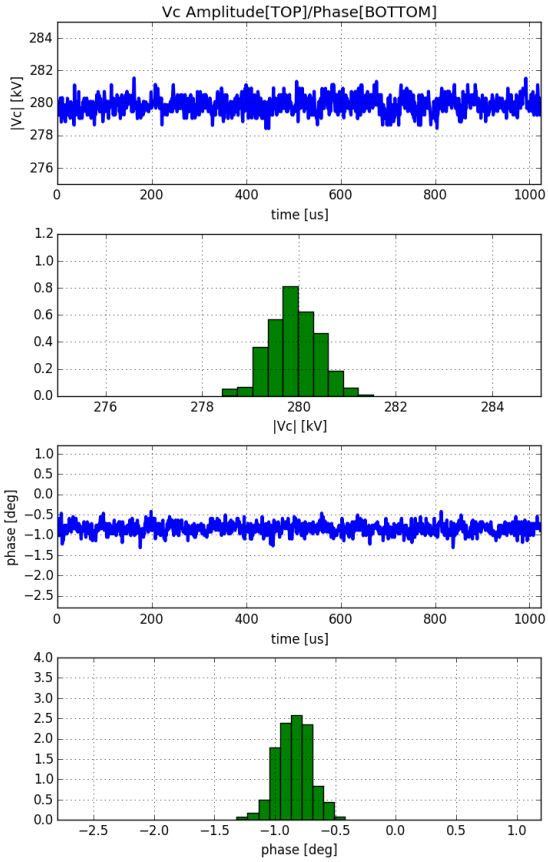

Figure 5: RMS error estimation for the gap voltage amplitude and phase at 280 kV for the TLS booster using the function of circular buffer.

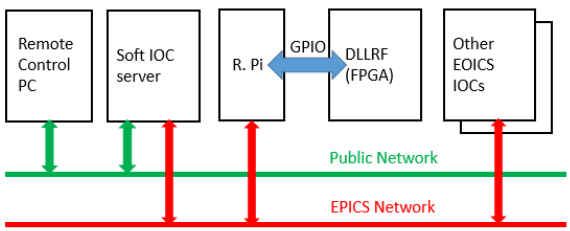

Figure 6: Function architecture for remote operation.

Other applications, utilizing the EPICS package, such as data acquisition and monitoring [8], can be implemented as well. Data acquisition servers on the EPICS network collect the PV information from the soft IOC. Users on the clients therefore can use these data to perform functions like archiving and monitoring.

CONCLUSION

A RPi is chosen for the IO controller of the D-LLRF system in NSRRC and utilizes GPIO pins for communications with the FPGA. Application programs can be developed with python and remote operation, data acquisition and monitoring can be implemented through the EPICS package. All functions were tested with the FPGA and show satisfactory functionality. More applications will be implemented in the near future.

REFERENCES

[1] C. C. Kuo *et al.*, "Commission of the Taiwan Photon Source", in Proc. IPAC'15, Richmond, USA, May 2015, paper TUXC3.

[2] M. S. Yeh *et al.*, "Low-Level RF Control System for the Taiwan Photon Source", in Proc. IPAC'11, San Sebastián, Spain, Sep. 2011, paper MOPC163, pp. 463-465.

[3] F. Y. Chang *et al.*, "Realization of Digital Low Level RF Control System for Taiwan Photo Source", presented at IPAC'17, Copenhagen, Denmark, May 2017, paper THPAB148, this conference.

[4] Raspberry Pi website, https://en.wikipedia.org/wiki/Raspberry_Pi

[5] RPi.GPIO website, https://pypi.python.org/pypi/RPi.GPIO

[6] EPICS website, http://www.aps.anl.gov/epics

[7] PyEpics website, http://cars9.uchicago.edu/software/python/pyepics3

[8] Y. H. Lin *et al.*, "Data Acquisition and Monitoring for TPS SRF Module Horizontal Tests", in Proc. IPAC'13, Shanghai, China, May 2013, paper THPEA053, pp. 3264-3266.

Copyright © 2017 CC-BY-3.0 and by the respective authors

ISBN 978-3-95450-182-3