Design and configuration of VME EPICS driver for He RFQ LLRF control system

Tae-Sung Ahn*, Hae-Seong Jeong, Seong-Gu Kim, Young-Gi Song, Han-Sung Kim, Kyung-Tae Seol, Hyeok-Jung Kwon, Yong-Sub Cho
Korea Atomic Energy Research Institute, Korea Multipurpose Accelerator Complex, 181 Mirae-ro, Geoncheon-eup, Gyeongju-si, Gyeongbuk 780-904, Korea
**Corresponding author: ats4815@kaeri.re.kr*

1. Introduction

Since the end of 2014 Korea Multi-purpose Accelerator Complex (KOMAC) has developed Helium Radio-Frequency Quadrupole (He RFQ). In He RFQ development, the role of the high-power Radio-Frequency (RF) is very important because it is responsible for stable delivery and efficient acceleration of the beam. Since the amplitude control requirements of LLRF system are ±1 % (amplitude), we need a precise remote control system for this reason [1]. This system is referred to as Low-Level RF (LLRF) control system. This paper describes the basic configuration tasks performed by hardware side and the software side to build the LLRF control system, and describes the future work of the He RFQ LLRF control system based on this paper.

2. Environment Setup

Since the control requirements for He RFQ LLRF system are ±1 % (amplitude), IVME7210 baseboard and Pentek7156 FPGA PCI PMC extended board are selected. Pentek7156 PMC module provides 20 MHz to 400 MHz of sampling rate unlike 125 MHz of Pentek7142 PMC module and up to 1 GB of DDR2 SDR and dual timing buses for independent A/D and D/A clock rates. Because of He RFQ uses 200 MHz of sampling rate, Pentek7156 is useful and selected. Figure 1 shows the VME SBC baseboard and FPGA PMC extended board for the LLRF control system. LLRF control system is divided into two major areas: hardware and software. In terms of hardware, there are two parts that a Versa-Module Eurocard (VME) Single Board Computer (SBC) as the Experimental Physics and Industrial Control System (EPICS) Input Output Controller (IOC) which help operator to easily access to LLRF system through the Ethernet and Field-Programmable Gate Array (FPGA) Peripheral Component Interconnect (PCI) Mezzanine Card (PMC) extended board which processes the data through the feedback control. In terms of software, there are several layers. There is driver support layer that configures the hardware modules. Above this, device support layer makes the connection with EPICS through the database layer. EPICS is standard software platform and provides a programming environment with open source. In client PC which remotes the LLRF system, operator controls the LLRF by using Channel Access (CA) of the EPICS communication protocol through Operator Interface (OPI).

Fig. 1. IVME7210 baseboard and Pentek7156 FPGA PMC extended board.

Fig. 2. LLRF control system test environment

Figure 2 shows the test configuration environment of LLRF control system. Software and hardware configuration for VME EPICS driver are implemented. First, driver support layer is a part of responsible for configure modules (such as ADC, DAC, FPGA, CLOCK, etc.) which are the FPGA PMC extended board. In this section, set the sampling frequency, clock, memory, data path, etc. The compilation takes the default example for each module in the Windriver workbench (Board Support Package) [2]. At this time, we should set the variable (ex. 715X → 7156) and routing (ex. EPICS_BASE, complier, source code, etc.) configuration [3]. Source files compiled, the object files are generated. This object file is downloaded on vxworks operating system of VME SBC. Module is ready to working. Second, Data processed in the driver support layer is stored in memory and waiting for request. Device support layer is a part that calls stored data and connects with the upper EPICS. The Device support is configured by modifying the device example

source code that is generated when compiling EPICS BASE. Modifications are corrected by referring to the record in EPICS database. Third, records are defined at database and are fetched by using functions in start-script of EPICS IOC. Using CA of EPICS communication protocol, operator can control the data in OPI.

3. Implementation

At first, we install the operating system by downloading vxworks image file on VME SBC. The vxworks operator system provides several advantages to user: a light kernel image, a real-time operating system, an easy cross-compile environment [4]. Download method is connection with VME SBC using console program (Teraterm in test environment) through serial communication. When we install the operating system, it should be noted vxworks boot parameter settings. We set the vxworks operator system IP as 192.168.2.11 and host PC IP as 192.168.2.30. Vxworks uses the 6.9 version. Since the VME SBC selects the vxworks operator system and its CPU is pentium, cross complier of EPICS BASE is vxworks-pentium architecture. After EPICS BASE compilation, move the module source files to a specified path for compile and create an object files. Object files are loaded to FPGA PMC module and executed by using function. Functions are described in each module source code file. And then, modules operate and wait for data from other modules or external clock or EPICS software. EPICS IOC operates by running a start-script in vxworks. Current EPICS IOC has example Process Valuables (PV) and not detects to device support layer. So, it is need to modify the database layer and connect with device support layer. Also, connection between device support layer and VME operator system is needed. It would be completed by using register memory. Figure 3 show that EPICS embedded software layer and data stream. Client OPI uses QT framework provided by EPICS extension software.

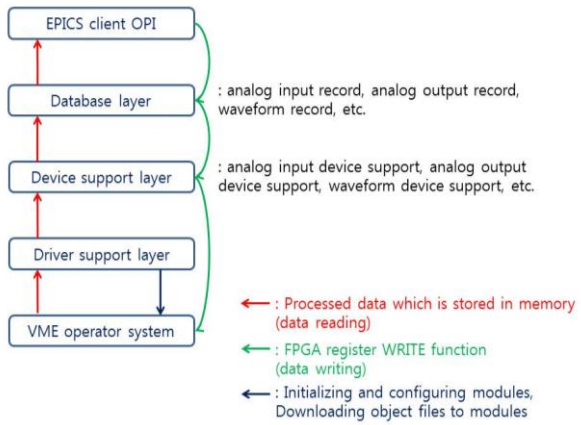

Fig. 3. Software layer of VME EPICS driver.

4. Conclusions

LLRF control system development at the He RFQ development stage is important. LLRF control system development requires the exact configuration of hardware and software. For each of the Layer configuration is completed on the software side and hardware modules: vxworks operating system installation, EPICS BASE compilation, module source code compiled, object file loading and execution on vxworks, EPICS IOC operation check, etc. Now the next step is the connection work between each layer. After you complete the actual hardware/software test, development of LLRF control system will be completed.

ACKNOWLEDGMENT

This work was supported by the Ministry of Science, ICT & Future Planning of the Korean Government.

REFERENCES

[1] Hyeok-Jung Kwon, Han-Sung Kim, Kyung-Tae Seol and Yong-Sub Cho, Control of the RF System for the Helium RFQ, 2014 Korean Nuclear Society Autumn Meeting, Pyeongchang, 2014
[2] WindRiver Systems, Inc. URL:http://www.windriver.com.
[3] Experimental Physics and Industrial Control System (EPICS). URL:http://www.aps.anl.gov/epics.
[4] Young-Gi SONG, Han-Sung KIM, Kyung-Tae SEOL, Hyeok-Jung KWON and Yong-Sub CHO, J. Korean Phys. Soc. Vol.59, No.2, p.577, 2011.