ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUAPL02

PORTING VME-BASED OPTICAL-LINK REMOTE I/O MODULE TO A PLC PLATFORM - AN APPROACH TO MAXIMIZE CROSS-PLATFORM PORTABILITY USING SoC

T. Masuda†, A. Kiyomichi,
Japan Synchrotron Radiation Research Institute (JASRI), Sayo, Hyogo, Japan

Abstract

The optical-link remote I/O system called "OPT-VME system" that consists of a VME master and several kinds of slave boards is widely used in SPring-8 and SACLA. As the next generation low-end platform instead of the outdated VMEbus, a Linux PLC such as Yokogawa e-RT3 has been considered. We have developed an e-RT3-based master module OPT-PLC to fully utilize a large number of existing remote boards.

In the original system, low-level communication is performed by FPGA and high-level communication procedures are handled in the Solaris device driver on a VME CPU board. This driver becomes a barrier to port the system to the e-RT3 platform. OPT-PLC should be handled by the e-RT3 standard driver in the same manner as other e-RT3 I/O modules. To resolve the difficulty, OPT-PLC was equipped with Xilinx SoC and the high-level communication procedures were implemented as application software on ARM Linux in the SoC. As a result, OPT-PLC can be controlled through the standard e-RT3 driver. Furthermore, the system will be ported to other platform like PCI Express by replacing the bus interface block in the PL part.

This paper reports on our development as an approach to maximize cross-platform portability using SoC.

INTRODUCTION

The control system of the SPring-8 accelerator complex employs a large amount of optical-linked remote I/O systems that consist of master and slave boards to efficiently control the widely distributed accelerator equipment in the large site from the front-end computers. Originally, four different kinds of optical-linked remote I/O systems were used. We have consolidated them into two through the update of the control system. One is an "RIO system" developed by Mitsubishi Electric Co. and over 1,400 RIO slave boards are employed since 1997. The other is an "OPT-VME system" [1] developed by SPring-8 and over four hundred slave boards are used since 2001. Since the RIO system has been discontinued already, we have completed the preparation to replace it with the OPT-VME system. The master boards using the RIO system and the OPT-VME system are based on VMEbus.

Meanwhile, VME is already out-of-date and we are considering the next-generation alternative platforms. As a high-end platform, MicroTCA.4 [2] is the best candidate at present. On the other hand, a Linux-based PLC such as e-RT3 [3] by Yokogawa Electric Co. is considered as a low-end platform. Several e-RT3s are applied as the front-end computer in both SPring-8 and SACLA.

In order to effectively utilize the resources of the large amount of OPT-VME slave boards, we have developed the e-RT3 based new master module named OPT-PLC for slow control applications such as magnet power supplies. In the development, we have considered maximizing the portability of the developed FPGA logic to other platforms such as the PCI Express based platform, for example, MicroTCA.4 as the next-generation high-end platform.

OPT-VME SYSTEM

The OPT-VME system consists of two types of master boards, "OPT-VME" (Fig.1) and "OPT-CC" (Fig. 2), and ten types of slave boards (Fig. 3). OPT-VME is a single-slot 6U VME board with four optical channels and OPT-CC is a dual-slot 6U VME board with twelve optical channels. TOSLINK [4] is used for all the optical channel connectors of the OPT-VME system. The communication protocol between a master board and a slave board is "OPT-Protocol 2006" [5] also developed by SPring-8 and one slave board can be connected to one optical channel of the master board. The main specification of the OPT-VME system is summarized in Table 1.

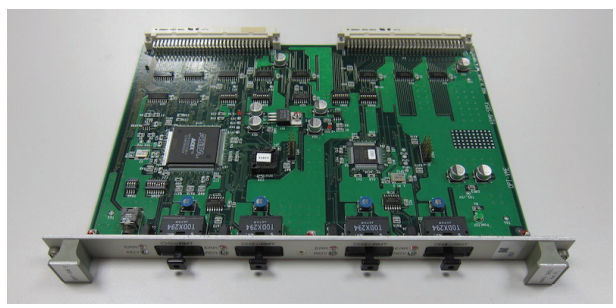

Figure 1: Photo of the OPT-VME board.

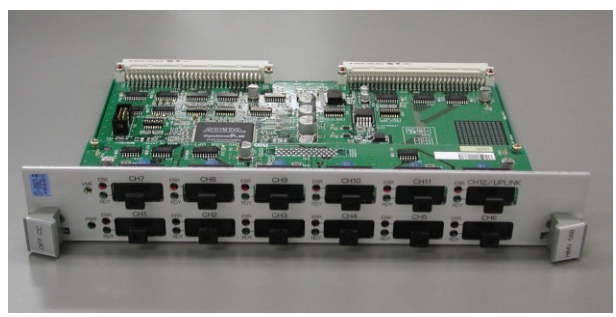

Figure 2: Photo of the OPT-CC board.

† masuda@spring8.or.jp

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUAPL02

Figure 3: Photo of a kind of slave board called OPT-RMT COMBOdao used for controlling steering magnet power supplies at the booster ring.

Table 1: Main Specification of the OPT-VME System

Transmission speed	10Mbps
Protocol	OPT-Protocol 2006
Transmission distance	300m
System configuration	1:1 communication per optical channel
Response time	50μs@direct connection / Max. 260μs@remote connection via OPT-CC
# of slaves per master	Max. 132 (12 x 11)
Optical fiber cable	Two core HPCF (200mm core/230mm clad) JIS C5976 F07 connector

Meanwhile the OPT-CC board can be used in the relay mode in addition to the master mode by changing a DIP switch on the board. It also works as a 1:11 multiplexer by switching the control interface from VMEbus to the specific optical channel (CH 12) of the board as shown in Fig. 4. That is, the remote slave board can be controlled by writing and reading data in the register for the slave board on the relay-mode OPT-CC through the communication from the master board.

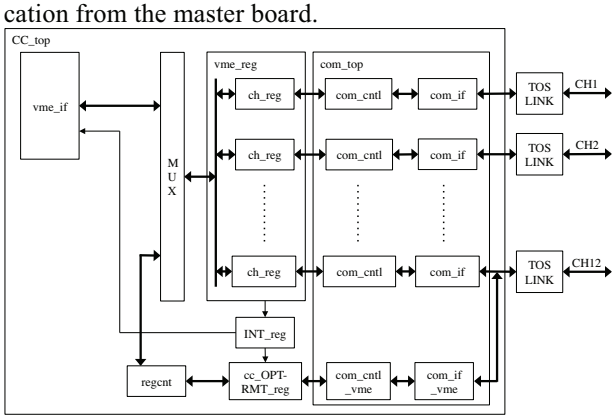

Figure 4: Block diagram of the FPGA control logic of the OPT-CC board.

Since the OPT-Protocol 2006 cannot cross a relay-mode OPT-CC, communication procedures with a remote slave board are implemented in a Solaris device driver for the VME master board. The difference in the communication procedures between a local slave board and a remote slave board are hidden in the device driver. Therefore, an application programmer does not need to change the software depending on whether the target slave board is connected in a local or remote configuration.

DEVELOPMENT OF THE OPT-PLC MODULE

Design Policy

In developing the OPT-PLC, an e-RT3 based new master module of the OPT-VME system, we have established the following policies.

1. Equip the module with as many optical channels as possible.
2. Separate an I/O unit from a logic control unit to provide enough versatility.
3. Control the module using the e-RT3 general-purpose device driver provided by Yokogawa Electric Co.
4. Control the module from a sequence CPU in addition to a Linux CPU.

As described in the previous section, the number of slave boards to be controlled from one master module is determined by the number of optical channels of the master module. Therefore, the module should be implemented with as many optical channels as possible (policy 1).

When policy 2 is achieved, the control logic unit is reusable for other purposes. Adoption of a standard specification such as FPGA Mezzanine Card (FMC) [6] is preferable for the separation. The versatility of the module is improved if the standard interfaces such as Ethernet, USB and SD card are prepared. On the other hand, preparing these interfaces is not always consistent with policy 1.

If policy 3 is realized, we do not need to spend much time and effort to produce and maintain a special device driver for this module and we can handle this module in exactly the same manner as the other e-RT3 I/O modules. Since the e-RT3 general-purpose device driver provides very primitive functions (specifying unit number, slot number, start address of the register, data count and pointer to the data) the challenge comes in implementing the communication procedure written in the device driver for the VME master board.

In order to accomplish policy 4, the minimum communication procedures including those for the remote slave boards are implemented in the FPGA control logic without using software.

Hardware Implementation

As a result of in-depth consideration, the OPT-PLC (Fig. 5) was developed as a dual slot module with five optical channels, which consisted of three printed circuit boards (PCBs). The first PCB is a control logic board equipped with Xilinx Zynq-7000 series System-on-a-Chip (SoC) [7], as shown in the left side of Fig. 6. The second and third PCBs are I/O boards with two and three optical channels, respectively, as shown in the right side of Fig. 6.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUAPL02

The second PCB is connected with the first PCB by a 70 pin stacking connector. The control logic PCB is equipped with a Gigabit Ethernet interface for ARM Linux running on a dual core Coretex-A9 ARM processor in the Zynq-7000 series, and a MicroSD interface for OS boot. The main specification of the control logic PCB is summarized in Table 2.

Figure 5: Photo of the developed OPT-PLC module.

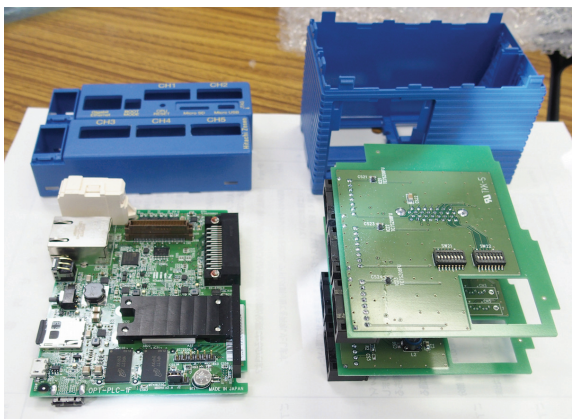

Figure 6: Photo of the logic control PCB on the left side and the I/O PCB with optical channels on the right side.

Table 2: Main Specification of the Logic Control PCB

SoC	Xilinx Zynq ; XC7Z015-1CLG485C
Memory	DDR3-SDRAM: 1GB QSPI Flash : 128MB
LAN	1 port (RJ-45 connector)
MicroSD	1 port (Micro-SD socket)
UART	1 port (Micro-USB connector)
High-speed serial I/F	4 pairs x 6.25Gbps in a 70pins stacking connector (Molex 53627-0774)
JTAG	1 port
Dimension	H100 x W58.11 x D83 [mm]
Power	+5V±5%

Thus, we have successfully established policy 2 that ensures high versatility of the logic control PCB while separating from the I/O PCBs with the optical channels. At first, we considered adopting FMC to stack the I/O PCB, but the PCB size of e-RT3 module was too small to mount FMC. For future extension, four pairs of high-speed serial links (6.25 Gbps) are assigned in the Molex 53627-0774 70 pins stacking connector. While keeping the versatility and expandability of the logic control PCB, five optical channels are successfully implemented on such a small module to realize policy 1.

Implementation of FPGA Logic

As an approach to achieve policy 3, the control logic PCB is equipped with Xilinx Zynq 7000 as the SoC device and the high-level communication procedures implemented in the Solaris device driver are built in application processes running on ARM Linux in Zynq. The communication procedure process prepared for each optical channel receives the control command and the additional parameters written in the dedicated DPRAM area by an e-RT3 Linux CPU module through the general-purpose device driver and operates the OPT-PLC control register. Figure 7 shows the block diagram of our developed FPGA logic.

The main part of the FPGA logic is built based on the Advanced eXtensible Interface 4 (AXI4) open bus [8] defined in the Advanced Microcontroller Bus Architecture (AMBA) standard to which the ARM processor is connected. At first, we planned to develop the logic modules such as the control registers for optical channels and DPRAM in the Programmable Logic (PL) block to be connected to the AXI4 bus. However, we found that this architecture became very complicated and required newly developed IP cores, so we have introduced the AXI4-bram-controller to convert AXI4 bus to RAM single access bus signals as arm-local bus.

The interface with the PLC bus is achieved by using PLC gate array and PLC-AVALON interface IP supplied by Yokogawa Electric Co. In order to realize policy 4, the control registers of the optical channels can be directly controlled from the PLC bus without the help of the ARM processor. We therefore prepare another path to access all the registers in the PL block separately from the arm-local bus by converting the AVALON bus to plc-local bus.

In addition to the above, the difference in the communication procedures between a local slave board and a remote slave board should also be incorporated in the FPGA logic for the achievement of policy 4. As a result of detailed investigation, we found that all differences could be included into the FPGA logic by improving the communication logic.

The control modules of the OPT-Protocol 2006 (com_if, cc_com_cnt) embedded in the OPT-CC FPGA logic are directly utilized in the OPT-PLC FPGA logic. These modules should be connected to the arm-local bus and the plc-local bus through the synchronization module that synchronizes the 80MHz clock of the control module of the OPT-Protocol 2006 with the 100MHz clock of the FPGA.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUAPL02

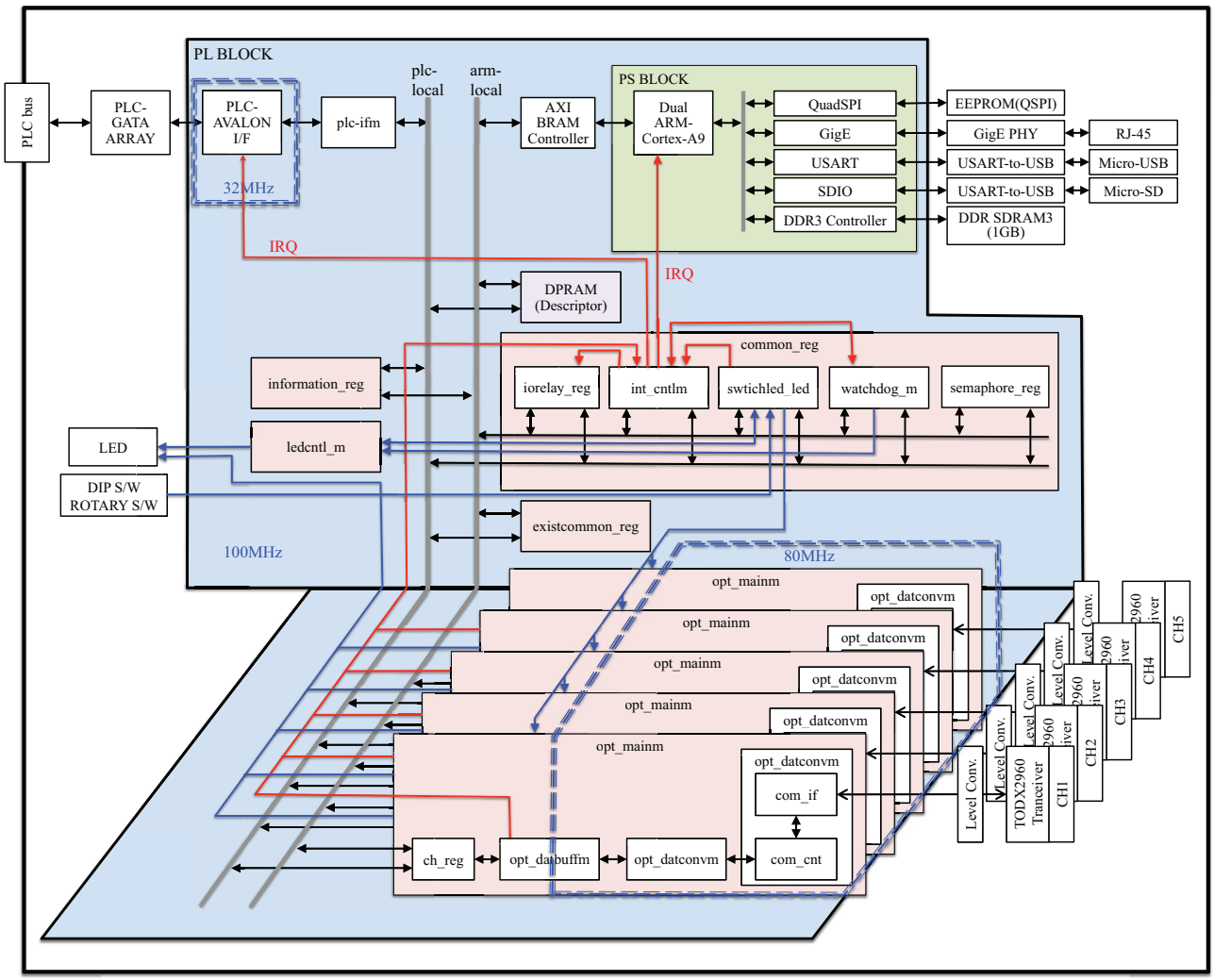

Figure 7: Block diagram of the OPT-PLC FPGA logic.

Register Structure

The address map of the OPT-PLC module from the view of the PLC bus is shown in Fig. 8. The map is classified into four areas.

1. Module management information area (MMIA), which is required for OPT-PLC to work as an e-RT3 module.
2. Common register area (CRA), which is used to control the peripheral devices of the OPT-PLC module.
3. I/O register area (IORA), which is required to control communication of the OPT-VME system.
4. Descriptor area (DA) as shown in Fig. 9, which is used to exchange the required information with the communication procedure processes.

Furthermore, the DA is divided into a common DA and five DAs for optical channels.

In normal operation, an e-RT3 Linux CPU module controls the OPT-PLC module by directing the communication procedure process on the ARM processor via the DA for each channel. Assuming operations from multiple Linux CPU modules, the exclusive control registers are implemented on the DA to realize exclusive controls between any processes.

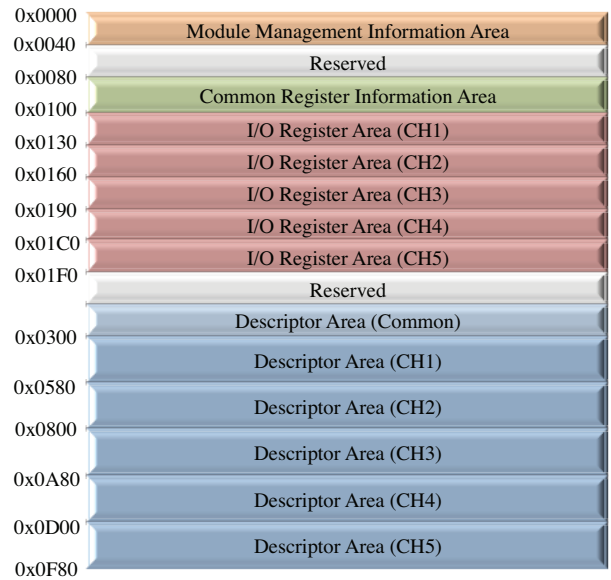

Figure 8: OPT-PLC register map from the view of the PLC bus.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUAPL02

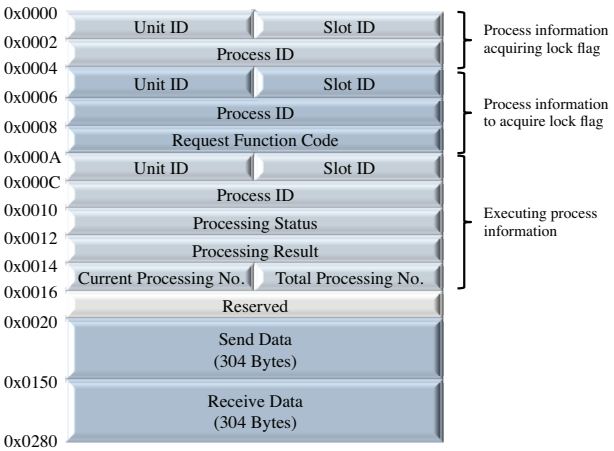

Figure 9: Register map of the descriptor area (DA).

On the other hand, the IORA can be directly accessed from the Linux CPU module, mainly for the debugging of the communication procedure process.

In addition, when installing a sequence CPU module into the PLC system, the module can control the communication of the OPT-VME system by operating the IORA. However, the sequence CPU does not use the exclusive control registers in DA. In the event that the Linux CPU is also installed in the PLC bus and controls the same OPT-PLC module with the sequence CPU, the Linux CPU should access the IORA via the sequence CPU.

Software Structure

The software structure of the OPT-PLC module is illustrated in Fig. 10. As described above, we have implemented the hardware and the FPGA logic to enable us to utilize the e-RT3 general-purpose device driver. In that implementation, we have successfully incorporated a part of the communication procedures of the OPT-VME system in the Solaris device driver into the FPGA logic. We have implemented the rest of the procedure in the Solaris device driver as an application running on the ARM Linux in the PS part of Zynq. It is very important to implement it as the application software and consequently, we can easily port the software when a SoC device is changed or when the ARM Linux version is updated. So the device driver for the ARM Linux to access the control registers is a simple driver considering the portability.

The control command to the communication procedure process is written in “Request Function Code” at the address of 0x0008. This code is exactly the same as the 2nd argument (request) of the ioctl system call. The process carries out the procedure according to the request function code in the same way as the procedure in the ioctl entry point in the device driver. Parameters passed as the third argument of the ioctl system call are set in the “Send Data” area with the address from 0x0020 to 0x0150. The process sets the execution result in “Processing Result” at the address of 0x0012 and the data is set in the “Receive Data” area with the address from 0x0150 to 0x0280. This process is prepared for each optical channel. We employ an entire-system control process to control the watchdog timers for the communication procedure process and the LEDs on the front-panel of the module.

On the Linux PLC side, in addition to the constraint of using the e-RT3 general-purpose device driver, we imposed the design conditions to maintain compatibility of API functions of the Solaris device driver supplied to the application programmers, as much as possible. Accordingly, we can port the API functions mechanically. However, the e-RT3 general-purpose device driver requires a unit number and a slot number that do not exist in the VME system, so we have added these parameters to the OPT-PLC API functions.

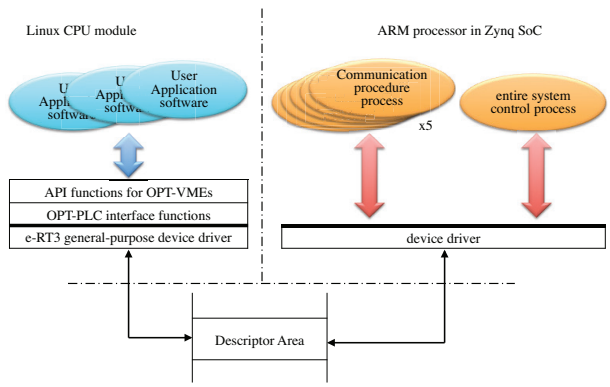

Figure 10: Software structure of the OPT-PLC module.

STATUS AND FUTURE PLAN

Implementation of the OPT-PLC hardware and firmware built on ARM Linux has been accomplished, and the API functions for two kinds of slave boards have been ported, with positive results.

As the practical test of the OPT-PLC module, we will apply the module to the control of steering magnet power supplies of the booster ring. The system consists of three VME-based master boards and 80 slave boards called OPT-RMT COMBOdao (Fig. 3) and several numbers of the relay-mode OPT-CC [9]. We will replace the VME system with an e-RT3 equipped with OPT-PLC modules.

SUMMARY

We have successfully ported the VME-based optical-link remote I/O module called OPT-VME system to the e-RT3 platform OPT-PLC module. The developed module OPT-PLC is equipped with Xilinx Zynq 7000 SoC in order to build the communication procedures mainly into the PS part as the application software running on the ARM Linux. As a result, the interface with the PLC bus is simplified and the e-RT3 general-purpose device driver is available. This means that we do not need a special device driver for the module to realize high-level communication between the master and slaves; we only need a simple device driver to manage memory access and interrupt. In addition, we can port the developed FPGA logic to an alternative bus such as the PCI express by replacing the PLC bus interface block in the PL part. The interface simplification enhances portability.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUAPL02

The hardware, firmware on the ARM processor, and software on the Linux CPU module are working well. The module will be tested for the control of the steering magnet power supplies of the booster ring.

REFERENCES

[1] T. Masuda, *et al.*,"Upgrade of the SPring-8 Linac Control by Re-engineering the VME Systems for Maximizing Availability", paper TU617, Proc. of ICALEPCS'03, Gyeongju, Korea, 2003.
[2] MicroTCA.4 , http://mtca.desy.de
[3] RTOS e-RT3 Plus, https://www.yokogawa.co.jp/rtos/rtos-index-ja.htm (in Japanese)
[4] Toshiba Optical Semiconductor Devices, http://toshiba.semicon-storage.com/ap-en/product/opto.html
[5] T. Masuda *et al.*, "Development of OPT-VME Board towards the Unification of Optical-Linked Remote I/O System at SPring-8", Proc. of 7th Annual Meeting of Particle Accelerator Society of Japan, Higashi-Hiroshima, Japan, 2008, p.377. (in Japanese)
[6] FMC Marketing Alliance, http://www.vita.com/fmc
[7] Zynq-7000 All Programmable SoC, https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html
[8] Architecture | AMBA 4 – Arm Developer, https://developer.arm.com/products/architecture/amba-protocol/amba-4
[9] S. Ueda *et al.*, "Upgrade of the Control System for Steering Magnet Power Supplies at SPring-8 Booster Synchrotron", Proc. of 9th Annual Meeting of Particle Accelerator Society of Japan, Osaka, Japan, 2012, p.695. (in Japanese)