

# REALTIME CONTROL SYSTEM FOR MICROPROBE BEAMLINE AT PLS

J. C. Yoon, J. W. Lee, K. H. Kim, and I. S. Ko  
Pohang Accelerator Laboratory, POSTECH, Pohang 790-784, Korea

## Abstract

The microprobe beamline of the Pohang Light Source (PLS) consists of main and second slits, a microprobe system, two ion chambers, a video-microscope, and a Si(Li) detector. These machine components must be controlled remotely through the computer system to make user's experiments precise and speedy. A real-time computer control system was developed to control and monitor these components. A VMEbus computer with an OS-9 real-time operating system was used for the low-level data acquisition and control. VME I/O modules were used for the step motor control and the scalar control. The software has a modular structure for the maximum performance and the easy maintenance. We developed the database, the I/O driver, and the control software. We used PC/Windows 95 for the data logging and the operator interface. Visual C++ was used for the graphical user interface programming. RS232C was used for the communication between the VME and the PC.

## 1 INTRODUCTION

The Pohang Light Source (PLS) is a 2-GeV third generation light source that has a 150-m long 2-GeV injector linac and a 280-m circumference storage ring. There are eight beamlines at the end of this reporting period; White-beam, NIM for ARUPS and GAS-phase, Photo-emission spectroscopy, EXAFS, X-ray scattering, Lithography, SAXS, and High Resolution X-ray Diffraction.

The Microprobe Beamline was designed for fluorescence microprobe applications, LIGA, TRXF, and the calibration of optical components such as monochromators, mirrors, etc. Most components of the photon transfer line of the front-end and the beamline, such as spool chambers, valves and shutters, pumps, vacuum gauges, cooling water lines, dry nitrogen lines for pneumatic actuators, etc, are assembled on supports/girders. There has been white beam available since June 1996. For this beamline, the real-time control system is divided into two layer systems, and these are linked to a serial network. A structure of the control system was finalized in March 1996, and the actual software development started in June 1996. Before

starting the major work, we made the signal list and the design manual for the microprobe beamline control system. This beamline control system was completed in October 1996. After some modifications and upgrade work during the machine commission period from November 1996 to July 1997, it now works very well to meet our requirements.

The microprobe beamline control system has two layers of the hierarchy; the user interface computer layer (UIC), and the machine interface unit layer (MIU). The UIC layer is based on PC (Intel 586) with Windows 95 and terminals. Visual C++/Borland C is used for the graphical user interface tool. The MIU layer is directly interfaced to individual machine devices such as a stepper motor, slit movers, limit switches, a Si(Li) detector and a sample stage. The MIU layer is based on the VMEbus standard with an OS-9 real-time operation system. The UIC layer and MIU layer are linked through RS232C. They communicate each other in the master-slave form.

## 2 SYSTEM CONFIGURATION AND HARDWARE

The microprobe beamline consists of various kinds of machine components. Major components include an optical bread board, 4-jaw slits which can adjust the horizontal and vertical acceptance angle from 0 to 3 and 0 to  $\pm 2$  mrad, respectively two ion chambers (one filled with  $N_2$  and the other with Ar), a Si crystal monochromator on a Huber 410 goniometer (XYZ• sample position stages), a video-microscope, and a Si(Li) detector as shown in Figure 1. The breadboard sits on a support with 6 degrees of free movements so that it can be put in a desired position and orientation. All stepping motors are controlled by a Pentium PC, which is linked to stepping motor VME I/O modules on a VME crate through the RS-232 serial communication. The output of the Si(Li) detector travels to a Multi-Channel Buffer (MCB) via a spectroscopic amplifier. The PC receives and controls the data from the MCB through an I/O card. Using a stepping motor controlled cross-slit as a pinhole, a medium-spatial-resolution X-ray fluorescence microprobe has been possible.

### 2.1 Machine Interface Unit (MIU)

The MIU is directly connected to individual devices and performs the low-level data acquisition. For the data acquisition and process in real-time, we adopt the VME system. It used Motorola's MVME147S-2 as its single board computer. MVME147S-2 has 8MB DRAM with 25 MHz CPU clock, one SCSI port, four serial ports, and one Ethernet port. It is also equipped with a floppy disk, a 1 GB hard disk, and a streaming tape backup. This system runs on the real-time operation system OS-9 V 3.0. We use a stepping motor I/O (Green Spring's IP-Stepper) for the interface to the slits of the microprobe scanning system. And two more I/O boards are used; one for the scalar module for voltage frequency counter at the ion chamber and an other for the tree industry carrier module (IP610) for IP-Stepper.

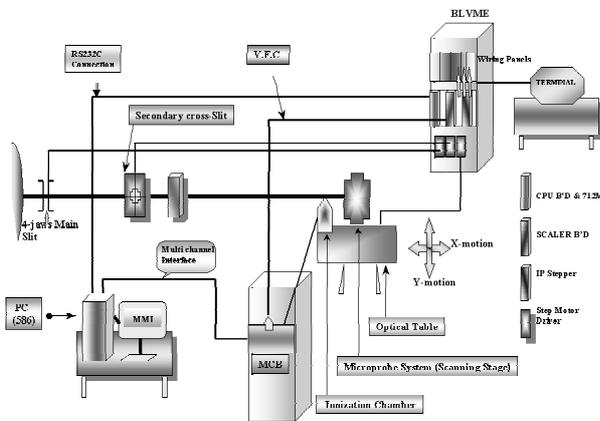


Figure 1: Structure of the control system.

## 2.2 User Interface Computer (UIC)

The UIC provides various kinds of graphical interface windows for the user to control and analysis the measured data from the microprobe beamline. In the UIC, there are one Pentium PC and one terminal, which are placed in the experimental station. A PC is set up for the beamline control and the data acquisition. A terminal is used to monitor the MIU development system and to display the local control program. Programmings for the graphic user interface have been done with Visual C++/Borland C.

## 3 SOFTWARE FEATURES

The major features that must be provided by the software of the acquisition subsystem are the real-time performance, communication facilities and convenient environments for software developments and debugging. In the field of the real-time system software, there is neither a standard nor a widely accepted even for general system architecture. A common approach is to provide the front-end processors with a real-time kernel and to

develop, with cross-development tools, all the application software on a host running a non real-time operating system.

We adopted in favor of OS-9 68K, which is a real-time operating system widely used in the industry and many HEP laboratories. OS-9, which is one of the most popular real-time systems for VMEbus, is adopted by PLS as a standard.

In parallel to the hardware structure, there are two layers in the software structure. The software structure of the Microprobe Beamline is shown in Figure 2. There are two important features in the control software; the master/slave model to exchange the data between the user interface layer and the machine interface layer. MIU software is basically divided in the following families; device access software, application user software, diagnostic monitor software, communication software, and utility software.

### 3.1 User Interface Computer Layer

The Microsoft Visual C++ development system for Windows is a very powerful workbench that provides a perfect visual interface for programmers. So the user interface programs dedicated to the system are written in Visual C++.

The major role of the user interface software in the UIC is to provide user with various tools to monitor and control the machine. Major components of the UIC software are a database, graphical user interface software, and communication software.

The database consists of the static database partition and the dynamic database partition. The static database stores all sorts of static information to manage machine control signals. The dynamic database stores set-points and read-back values for control signals.

We have adopted a master/slave model for the communication between the UIC and the MIU. Slave tasks running on the UIC send service requests to master tasks running on the MIU.

Graphical user interface software is provided to the user for easy operating environments. There are several windows for operators to control and monitor the individual system such as a main slit, a cross slit, an optical table control panel, and a MCB control panel. Each window has a value display area and a control sub-window. Selecting areas with a mouse or select items from the pull-down menu can make any control action easily.

### 3.2 Device access software

The device access software uses various I/O descriptors and drivers for driving hardware I/O ports. So, graphical user interface programmers do not need to know about hardware features, addresses, and configurations. The device access software consists of a database generation task (*gener\_tsk*), a command buffer (*cmdbuf*), and a current local database (*blmiudb*).

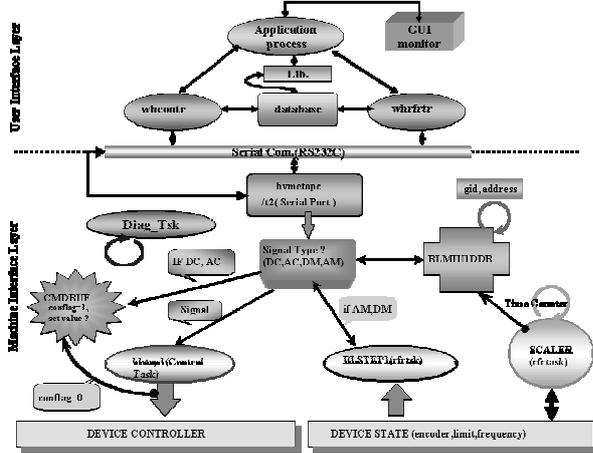


Figure 2: Software structure of the control system.

### 3.3 Application user software

The application user software allows the direct control and monitoring of individual device. This software deals with the translation between the global device signal identification number (GID) and its associated physical address of I/O point.

The application user software consists of the control software (*blstep1*) and the monitor software (*blstep1\_rfr*). The control software controls the individual device according to the device control command, which consists of a GID and a set-value. The monitor software monitors periodically the current status of devices and put monitored data into the local database.

### 3.4 Diagnostic monitor software

The diagnostic monitor software checks and gathers various system-related information such as an application task status, a device status, a network status, etc. This software stores all these data into the special local buffer called the diagnostic buffer, which can be used for the system diagnosis later.

### 3.5 Communication software

This software allows message exchanges between the UIC and the MIU. This software uses a simple command-response model implemented with a low-level packet transmission scheme. The UIC initiates the

communication session by sending a command packet to the MIU. The MIU responds according to the command packet.

In the MIU, there is one network task related to RS232C network called *bvmetopc*. This task is normally in the sleeping state and is awakened by the command packet sent by *whcontr* or *whfrfr* to the UIC. If this is the device control command packet, *bvmetopc* extracts the device set-value and the GID from the command packet. For the fast action, the network task notifies the appropriate control task of the arrival of the command through the software interrupt signal. If this is the database refresh command packet, *bvmetopc* makes a response packet consisting of all GIDs and devices values stored in the current local database, and sends it to *whfrfr* task running on the UIC.

## 4 CONCLUSION

We have described the real-time control system for the microprobe beamline at the PLS. At present, we have a reliable control system for the microprobe beamline. We are upgrading and modifying the control system to accommodate new control requirements and to apply long-term operational experiences.

From now on, it is planned to build over 20 beamlines including five insertion devices by the year 2002, and they need a control system to make user's experiments precise and speedy. We are developing shared library codes for fast developments of beamline control systems.

## 5 ACKNOWLEDGEMENTS

This work is supported by Ministry of Science and Technology, Korea and Pohang Iron & Steel Co.

## 6 REFERENCES

- [1] Pohang Accelerator Laboratory, "1996 Annual Report," 1997.
- [2] I.S.Ko, Jin W.Lee, Jong C. Yun, Eun H. Lee, and Byoung R. Park, "Computer Control System of PLS 2-GeV Storage Ring," in Proc. of IEEE Real-Time Conference 97 (to be published).
- [3] "OS-9 Operating System Technical Manual", Microware Systems Cooperation.