STEPPING MOTOR CONTROL FOR SEPTUM PLATE POSITIONING

K. Herlo, GSI, Darmstadt, Germany

Abstract

A new injection septum has been installed in the GSI accelerator facility. Both septum plates can be adjusted by four stepping motors, where two motors per plate have to be moved simultaneously and in parallel. Since the GSI standard stepping motor controller can handle only one motor at a time, a commercial stepping motor controller, microIOC-M-Box from Cosylab Ltd., had to be integrated, that can handle all motors simultaneously. We installed the software development system and ported the new GSI front-end control software, the device manager, to the motor controller. Afterwards, we designed and implemented a new device model that assures parallel movements and does extensive checks to prevent from too large skew angles. In the operator's interfaces, no particular care to meet the position constraints is needed.

INTRODUCTION

The heavy ion synchrotron (SIS) in the GSI accelerator facility is presently been upgraded. A part of the upgrade program is a new injection septum (see Figure 1). Each septum plate can be moved by two independent stepping motors. When moving the motors, only small skew angles are allowed. Too big differences in the positions of both motors of a plate will result in stress on the ceramic bearings which then may break.

The standard stepping motor controller in the GSI accelerator unfortunately can handle only one motor at a time. When used to drive a septum plate, both motors have to be moved alternatively stepwise to always assure small skew angles: A cumbersome and error-prone process.

Figure 1: Inside the new injection septum.

Presently, the control system's communications middleware is rebuild to overcome the limitation to the original proprietary front-end platform, which are VME CPUs with MIL-Bus device interfaces. A newly developed device manager, using CORBA as communication middleware, replaces the system software on the VME master controller [1].

The new controls middleware allowed integration of a commercial stepping motor controller for the septum, Cosylabs [2] microIOC-M-Box [3]. The microIOC-M-Box combines programmable multi-axes stepping motor controller, LCD/LEDs and an embedded system based on Intel CPU running Debian Linux (see Figure 2).

Figure 2: microIOC-M-Box

To include the microIOC-M-Box into the GSI control system, the Linux environment had to be prepared for the device manager requirements. The device manager had to be ported and missing system functionalities had to be implemented. Finally a new device model had to be designed and implemented for the septum device.

SOFTWARE DEVELOPMENT ENVIRONMENT

microIOC software development environment is a software package provided by Cosylab. The software is installed on personal computer with a RedHat Linux operating system. The development environment behaves as a Debian Linux operating system command shell with some additional commands and utilities. It enables development and building of software for the microIOC.

To build the device manager further programs are needed. In the development environment a GNU-Compiler-Collection [4] in version 2.95 was installed. We needed a newer one. Now a GNU-Compiler 3.4.6 is installed.

omniORB 4.1.2 [5] was installed. It is a high performance CORBA ORB, that is needed for the communication middleware.

All necessary new installed libraries have to be installed on the microIOC-M-Box too.

LINUX ENVIRONMENT

General

Some common services have to be set up.

- Setup of network services. microIOC-M-Box get its network address over DHCP.
- The host name has to be defined.
- The system time has to be set up.
- The GSI multicast has to be defined. It is needed for the alarm handler.

Logfiles

The device manager uses the Linux system functionality `syslog()`. `syslog()` generates a log message, which will be logged by `syslogd`. Depending on priority of the message, it will be written into an applicable logfile. The following priorities are used from device manager:

- err (error), that is the highest priority and will be written into the error_log-logfile
- debug, that is the average log message level. This message will be logged into the debug-logfile.
- info, the lowest log message level. It will be written into the messages-logfile.

The new syslog has to be defined and the new logfiles have to be made known in the `syslog.conf` file.

```
>> local0.err    /var/log/equ/error_log
>> local0.info   /var/log/equ/messages
>> local0.debug  /var/log/equ/debug
```

Service Start-stop-daemon

The start-stop-daemon is a linux service. It creates and terminates daemons or processes. For the device manger a service script is implemented, and for each linux system runlevel the appropriate action is taken. E.g. by starting the microIOC-M-Box the service start-stop-daemon starts the device manager.

MISSING SYSTEM FUNCTIONALITY

In GSI's VME front-end environment the device manager uses some services which are implemented on a separate equipment controller [1]. To provide full functionality on the microIOC-M-Box, theses services had to be added to the device manager.

- A service to update every four seconds the status of a device. The new status has to be compared with the previous one. If any bit has changed an alarm has to be sent. This data packet includes the name of the device, the old status, the new status and a time stamp.
- A functionality to "INIT" and "RESET" the device manager has to be implemented. At the moment both functionalities reload the device constants for all declared devices by the device manager.

DEVICE SOFTWARE

The injection septum has two electrode plates, cathode and anode. Each plate is mounted on two ceramic bearings which can be moved by independent stepping motors. To position a plate always two motors of the plate have to be moved simultaneously and in parallel. All needed checks will be handled in the device software, as soon as the motor control is implemented in the septum device software.

Check of Plate Distance

One condition of the control is to keep a minimum and maximum distance. The minimum and maximum distances are given. Before a movement is made the result of the new distance between the plates will be checked. The new distance has to be in its allowed limits.

Check of Skew Angles

Small skew angles are not a problem. But too big differences in the positions of both motors of a plate will result in stress on the ceramic bearings which then may break.

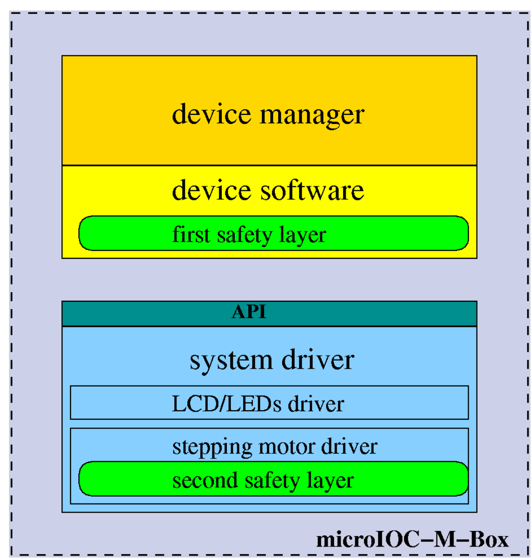

Figure 3: microIOC-M-Box system overview

It is very important to provide safety of the injection septum. The software has to make sure that the differences of both motors positions of a plate will not be too large. Therefore a double safety layer was implemented. Figure 3 shows the microIOC-M-Box system overview and the implemented safety layers.

First layer All move commands from the device software (properties) will split in several small move commands. After each move command the skew angle of the plates will be checked. If the skew angle is too large, all movements will be stopped immediately.

Second layer The driver for the stepping motors checks the skew angle before executing the move command. No movements will be made if the skew angle is too large.

Septum Control

According to GSI's representation of devices in the control system, the septum plates have to be modelled by a set of properties, implemented on the microIOC-M-Box. The above described checks are performed in the properties.

Table 1 presents the implemented properties.

Table 1: Injection Septum Properties

Property	**Meaning**
Read DISTI	The property reads the actual position of the cathode and anode plate. Then return the calculated distance.
Write DISTS	The property moves the cathode plate by the given distance.
Read DISTI	The property returns the nominal distance of the plates.
Call INITIAL POSITION	The property moves all four motors to the known initial positions. This positions are particularly suitable for the injection.
Write POSIREL	The property moves the cathode plate relatively to the given value.
Read POSITI	The property returns the actual position of the anode plate.
Write POSITS	The property moves the complete system of cathode and anode plate to the new given absolute position of anode.
Read POSITS	The property returns the nominal position of the anode.
Read POSIABSI	The property reads all actual positions.
Read POSIABSS	The property returns the nominal positions of all motors.
Call STOP	All motions stops immediately.
Read STATUS	The property reads several actual conditions, e.g. the anode motors move. The important conditions that gives an alarm are: for all motors touch of inner or outer switch, skew angles of the plates and passing the minimum or maximum plate distance.

Operating Application

Figure 4: Java Application for the Septum control

The septum motor device can be accessed in the same way as any other device in the GSI control system. However, being one of the first devices connected by the renovated network communication, it can be accessed directly from Linux and Windows computers.

An operating application to control the septum is shown in Figure 4. The application was implemented by Cosylab [2]. The application uses the above described properties. In the operator's interfaces, no particular care to meet the position constraints is needed.

CONCLUSION

The injection septum has been installed at June 2008 in the GSI accelerator facility. Since then the stepping motor control proved its suitability to adjust the septum plates. The operators are satisfied with the functionality of the device control software.

The implementation of the microIOC-M-Box shows the benefits of the renovated network communication [1] that allows to integrate a wide range of equipment hardware.

REFERENCES

[1] U. Krause, L. Hechler, K. Herlo, K. Höppner, P. Kainberger, S. Matthies, G. Schwarz, "Integration of Renovated Networking Middleware into a Running Control System Environment", `MOW03`, PCaPAC 2008

[2] Cosylab Ltd. `http://cosylab.com`

[3] microIOC `http://microioc.com`

[4] GCC, the GNU Compiler Collection `http://gcc.gnu.org`

[5] omniORB `http://omniorb.sourceforge.net`