

PC/104 ASYN DRIVERS AT JEFFERSON LAB

J. Yan, T. Allison, S. Witherspoon
Jefferson Lab, Newport News, VA 23606, U.S.A.

Abstract

Asyn Driver was applied for PC/104 IOC serial communication systems at Jefferson Lab. We chose the iGPIB-PC/104-XL as the GPIB interface module and developed a low level device driver that is compatible with the asynDriver. Instrument device support was created to provide access to the operating parameters of GPIB devices. A Low level device driver for the serial communication board Model 104-COM-8SM was also developed to run under asynDriver. This serial interface board contains eight independent ports and provides effective RS-485, RS-422 and RS-232 multipoint communication. StreamDevice protocols were applied for the serial communications. The asynDriver in PC/104 IOC applications provides a standard interface between the high level device support and hardware level device drivers. This makes it easy to develop the GPIB and serial communication applications for PC/104 IOCs.

applications, one PC/104 IOC was required to provide multiple-port serial communications. Therefore, a serial communication board, Model 104-com-8SM, was chosen for this purpose. This serial interface board contains eight independent ports and provides effective RS-485, RS-422 and RS-232 multipoint communication. By setting jumpers on the board, each channel can be configured to any of these modes. A type of XR16L788 octal Universal Asynchronous Receiver and Transmitter (UART) on the board is used as the Asynchronous Communications Element (ACE). A XILINX XCR3256XL Complex Programmable Logic Device (CPLD) is applied to control the XR16L788 chip and communicate with PC/104 ISA bus. Two 40-pin connectors are used for interfacing to communication lines for 8 channels. Figure 1 shows the eight-channel serial communication chassis, where the 104-com-8SM board is stacked on a PC/104 processor module.

INTRODUCTION

PC/104 embedded IOCs that run RTEMS and EPICS have been applied in many new projects to control all kinds of different devices in Accelerators at Jefferson Lab [1]. Different commercial PC/104 I/O modules on the market such as digital I/O, data acquisition, and communication modules are integrated in our control system. Many devices and instruments are controlled via serial and GPIB communications. With the availability of low cost PC/104 I/O modules, it would be easy to configure these systems. However, device drivers and device support for these new PC/104 I/O modules have to be written under RTEMS and integrated in EPICS system. Solutions that apply EPICS tools and develop the generic drivers for these new modules were researched. asynDriver[2][3] is a general purpose facility for interfacing device specific code to low level communication drivers. It provides a structured environment for developing support for hardware devices both asynchronous and synchronous communications. Since asynDriver provides the EPICS IOC device support, only low-level device drivers for the new I/O modules need to be programmed. This paper presents the selection of PC/104 I/O modules for serial and GPIB communications, the programming of low-level device drivers for these modules, and some samples of applications.

HARDWARE CONFIGURATIONS

The Kontron PC/104 processor board provides two ports, Com1 and Com2, for serial communication. When RTEMS is running, Com1 is configured as the console and Com2 is disabled. Some applications may need RS-232 serial communication, so Com2 can be enabled and configured to provide this function. For some other

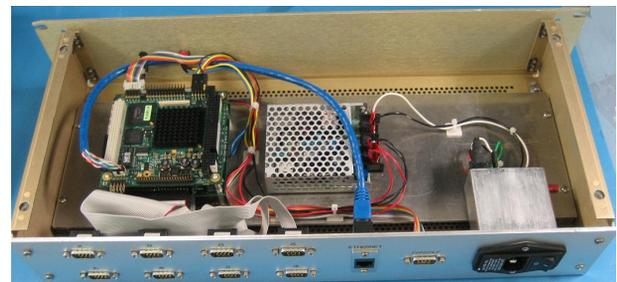


Figure 1: 8-Channel Serial Communication Chassis.

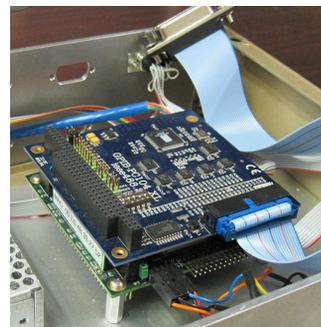


Figure 2: GPIB-PC/104-XL Module Stacked on the PC/104 IOC.

The iGPIB-PC/104-XL was chosen as the GPIB interface control module for the GPIB communication system. This module has an iGPIB 72110 chip, which provides an interface between a microprocessor system and the GPIB specified in the IEEE Std. 488.1-1087 and 499.2-1987. The iGPIB 72110, a 100-pin TQFP package, is register compatible with NEC uPD7210 in GPIB

Talker/Listener applications. Figure 2 shows the GPIB-PC/104-XL module stacked on a PC/104 processor module through the ISA bus. A 24-pin ribbon cable provides the GPIB interconnects between the interface module and up to 15 GPIB devices.

SOFTWARE DRIVERS

Since the asynDriver provides all components of device support, only the low-level driver needed to be programmed for the new I/O hardware. We wrote drivers for the ines GPIB-PC/104-XL module, 104-com-8SM model, and on-board COM2 port.

On the 104-com-8SM serial module, XR16L788 integrates functions of 8 enhanced 16550 UARTs, a general purpose 16-bit timer/counter and an on-chip oscillator. Device configuration registers include a set of four consecutive interrupt source registers that provides interrupts-status for all 8 UARTs, timer/counter and a sleep wake up indicator. Each UART channel has its own 16550 UART compatible configuration register set for individual channel control, status, and data transfer. The driver was written based on the source code for `drvAsynSerialPort.c` in the asyn package. However, the register access and interrupt handler are very different between the PC-104/RTEMS and the targeted VME/VxWorks. The structure of the driver can be described as the following:

- Create the structure for ports hardware-specific information.
- Initialize each port.
- Create the IRQ handler for each port
- Create and register asyn port name for each port.
- Set baud rate, parity, bits, stop bit for each serial port.
- Link with higher level routines of asynDriver.

The iGPIB 72110 chip on the GPIB interface module is register-compatible with NEC uPD7210, so the GPIB control interface driver was developed under the specification of uPD7210 registers. The uPD7210 is an intelligent control designed to provide high-level protocol management of the GPIB communication. Control of the uPD7210 is accomplished via 16 internal registers. The driver was programmed based on the file of `drvNi1014.c` in the asyn package. Here is the structure of the driver:

- Create the port name.
- Set the base I/O address.
- Set the interrupt vector and level.
- Create RTEMS IRQ handler.
- Register the asyn port and connect it with asyn.

All the low-level drivers were compiled as shared library files, so any application can call them.

APPLICATIONS

Most function generators and measurement instrument devices have a GPIB connector. In our new RF control system test stand we have three devices that need to be

controlled via GPIB. These devices are CG635 Synthesized clock generator, Agilent E4428C Signal Generator, and Giga-tronics 8540C Universal Power Meter. Figure 3 shows the schematic of the RF control system test stand. For each GPIB device we developed device support according the protocol of asynGpiB device support. First of all, we have to determine the set of operations that each device will have to perform. Then we create the device support file for each device and declare the command array. Each command array entry describes the details of a single I/O operation type. Finally, the application database uses the index of the entry in the command array of each file to provide the link between the process variable and the device I/O operation to read or write the value. In this application, CG635 and E4428C mostly require write operation, while 8540C power meters employ reads.

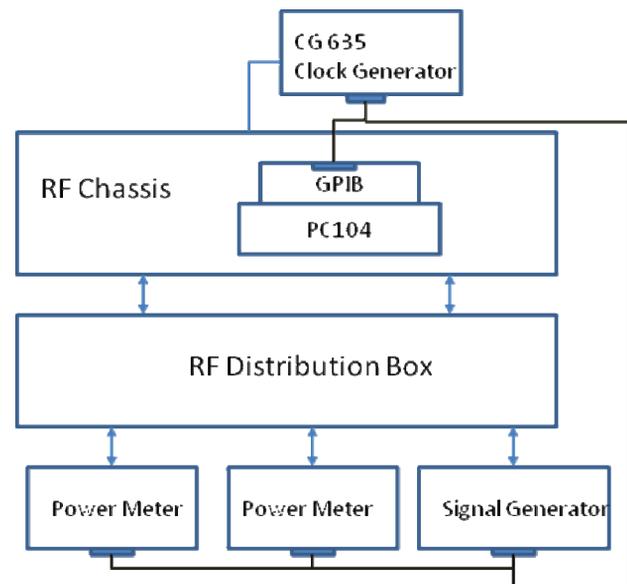


Figure 3: GPIB Communication for RF Control System Test Stand.

Beam Position Monitor (BPM) test stand is a system that calibrates 4-wire BPM cans. This system consists of a 4-channel BPM data acquisition board [4], PC/104 IOC and serial module, two Applied Motion STAC6-Si stepper drives, and a BPM can attached with two HT23-549 stepper motors. The data acquisition board reads the electrical signals from the four (x+, x-, y+, y-) electrodes inside the BPM can. Each stepper motor controls the movement of the BPM can in horizontal (X) or vertical (Y) directions. When the BPM can is moved to a specific position, two serial ports read back the value of the position and the data acquisition board would simultaneously sample the electrical signals from four antennas inside the BPM can. Stepper motors are controlled by writing and reading a set of commands to the stepper drivers. *StreamDevice* [5] is applied as the device support for controlling stepper drivers.

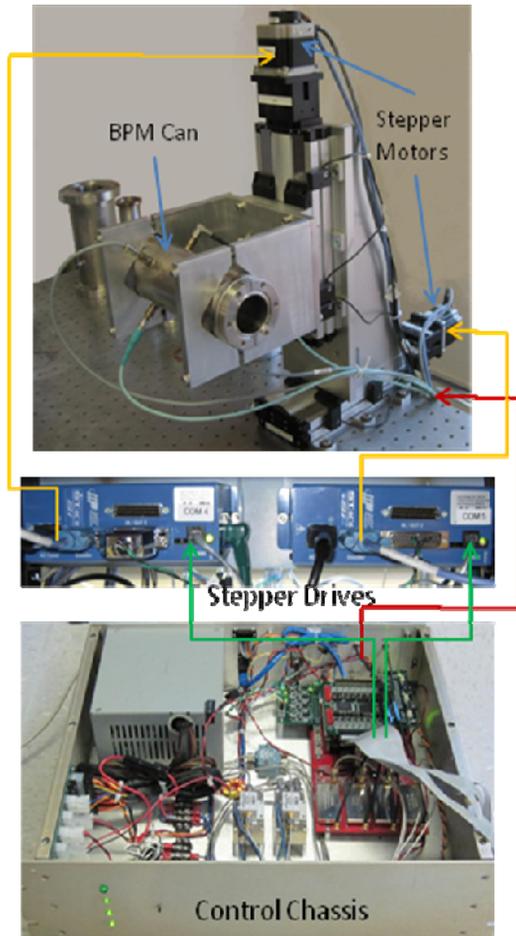


Figure 4: BPM Test Stand Stepper Motor Control System.

StreamDevice is a generic EPICS device support for devices with a “byte stream” based communication interface, such as RS-232, RS-485, GPIB, and telnet-like TCP/IP. It can be configured for any device type with protocol files in plain ASCII text which describes the commands a device understands and the replies it sends. In this application, each stepper drive has one specific protocol file that defines the commands to communicate between the device and database records. Each record with *StreamDevice* support runs a command from the protocol file to read or write its value. The advantages of *StreamDevice* are the ease to configure commands to run

devices and the support for all standard record types in EPICS base which have device support.

CONCLUSIONS

Asyn driver has been applied as generic device support for the embedded PC/104 IOCs that run RTEMs and EPICS. The low level drivers for a GPIB interface module and 8-channel serial communication module were developed to be compatible with the asyn toolkit. A number of applications have been written to use the PC/104 asyn driver in our control system at Jefferson Lab. The asyn driver provided an easy integrated solution for PC/104 serial communication applications.

ACKNOWLEDGEMENT

Thanks to the Instrument and Control group and Software group in Jefferson Lab for their technical support.

Notice: Authored by Jefferson Science Associates, LLC under U.S. DOE Contract No. DE-AC05-06OR23177. The U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce this manuscript for U.S. Government purposes.

REFERENCES

- [1] J. Yan, etc. “PC/104 Embedded IOCS at Jefferson Lab”, Proceedings of ICALEPCS2009, Kobe, Japan, P. 230-233.
- [2] M. R. Kraimer, etc. “EPICS: Asynchronous Driver Support”, 10th ICALEPCS Int. Conf. on Accelerator & Large Expt. Physics Control Systems, Geneva, 10-14 Oct 2005 PO2.074-5.
- [3] <http://www.aps.anl.gov/epics/modules/soft/asyn/R4-16/asynDriver.pdf>.
- [4] J. Musson, etc. “Reduction of Systematic Errors in Diagnostic Receivers Through the Use of Balanced Dicke-Switching and Y-Factor Noise Calibrations”, Proceedings of PAC09, Vancouver, BC, Canada, P.4057-4059.
- [5] <http://www-group.slac.stanford.edu/cdsoft/icalepcs01/StreamDevice.pdf>