Smart Rack Monitor for the Linac Control System

S. Shtirbu, R. W. Goodwin, E. S. McCrory, M. F. Shea
Fermi National Accelerator Laboratory
P. O. Box 500
Batavia, IL 60510

Abstract

The Smart Rack Monitor (SRM) is a low-cost, one board, data-acquisition module for the upgraded Linac control system at Fermilab. The SRM is based on the Motorola MC68332 microcontroller in the Business Card Computer (BCC) configuration. It is connected to the Linac local control station (to be referred to as Local Station) by an Arcnet LAN, and can be located close to the controlled hardware. Each Local Station is connected to several SRMs. The SRM has 64 A-D channels, sixteen D-A channels, and eight bytes of digital I/O on the mother board. Software Components Group's pSOS is the SRM's kernel. The SRM's software is cross-developed on VAX/VMS in C. The SRM does not have an attached console and is fully controlled by the Local Station. It performs data acquisition and settings, as directed by the Local Station. Its existence is transparent to the rest of the control system. The SRM supports code updates downloading from the VAX, through the Local Station.

I. INTRODUCTION

The 400MeV Linac upgrade project at Fermilab has spawned an upgrade of the Linac local control stations. The major change is the upgrade from a Multibus MC68000 system to a VME MC68020 system. The Local Stations are connected to each other and to the rest of the accelerator control system by a Token-Ring LAN. Since the new Local Stations are more powerful than the previous ones and can control more devices, fewer of them are needed. Reducing the number of Local Stations requires extensive wiring to reach one Local Station from several RF stations.

The Smart Rack Monitor is based on the Rack Monitor designed for DZero [1]. It has its own processor and controls the local devices. It is connected to Local Stations by an Arcnet LAN, which eliminates the need for the extensive wiring. Since such a design also frees the Local Station's CPU from data acquisition tasks that are performed in parallel, more devices can be controlled by one Local Station.

II. GENERAL DESCRIPTION [2]

A. Business Card Computer

The SRM is based on the Motorola MC68332 microcontroller (MCU). This new processor is available in a small subsystem called Business Card Computer (BCC), a 2.25 by 3.5 inch board that has 128 kbytes of PROM, 64

kbytes of RAM, and an RS-232 serial interface. (The SRM is not normally connected to a terminal). The BCC attaches to the motherboard by two 64-socket DIN connectors.

B. Local Area Network Interface

Standard Microsystems Corporation's COM90C65 Arcnet adapter is integrated into the SRM's motherboard.

C. Memory

In addition to the memory on the BCC, the SRM has 64 kbytes of non-volatile RAM. This memory is used for the SRM's code and tables.

D. Packaging

The SRM is a self-powered, 2U (3.5 inch) rack mounted chassis. It has analog and digital I/O connectors on the lower 1U of the rear panel, with additional connector space available on the upper part for connectors used by daughterboards.

III. ON BOARD I/O CAPABILITY [2]

All I/O devices on the SRM board are memory mapped, to allow easy access from the controller.

A. Analog Input

There are 64 channels of differential A-D input. Four connectors on the rear panel of the SRM are used to connect the analog inputs to eight 8-channel differential input multiplexers. Outputs from these multiplexers are combined by a ninth multiplexer that drives an instrumentation amplifier followed by the digitizer. The family of Crystal Semiconductor converters used is pin-compatible with twelve, fourteen or sixteen bit resolutions. The Crystal Semiconductor A-Ds are CMOS devices and include an onboard microcontroller to recalibrate the device at power-on, or by an external command. The converters include an input sample-and-hold function.

B. Analog Output

Sixteen D-A output channels are provided using four Analog Devices' AD664 quad D-A circuits. The D-A registers are accessible for both read and write. Output ranges of these converters are programmable.

C. Digital I/O

Eight bytes of digital I/O are on the SRM board. Normally, the 74ALS652 chips are used. These chips offer non-inverting bidirectional octal latch-buffer circuit with high active totem pole outputs. All digital I/O lines are pulled to +5V with 1 kohm pullup resistors. The data direction of the digital I/O is determined for each byte by a front panel piano DIP switch.

Photograph of a Smart Rack Monitor

IV. DAUGHTERBOARDS [2]

One of the major advantages of the SRM design is the ease of adding custom interfaces where needed. These interfaces are implemented as daughterboards with the same footprint and pinout as the Motorola BCC module, so that all the facilities of the MCU are available. Such boards can be stacked with virtually no limit (there are three sockets for BCC-pinout boards on the motherboard). Several such boards are already implemented.

A. Interface to old Linac hardware

Two daughterboards are needed to interface with the existing Linac hardware. One of them handles nine bytes of digital I/O, and the other interfaces to multiplexed D-A, A-D chassis and three digital I/O bytes. These daughterboards have the same interface as the Multibus boards used in the original Local Stations.

B. Four bytes of digital I/O

This board is an extension to the digital I/O capabilities on the SRM motherboard (eight bytes). These daughterboards can be plugged-in and used for input/output digital I/O when more devices need to be connected.

C. Four Channel Predet Timer

This board uses the Actel-1010A field programmable gate array. It features four run-time programmable 16-bit delay registers (interfaced as D-A channels). It is driven by an external free running clock and a trigger pulse. After a trigger is received, it counts clock pulses. When the count matches the value in one of the delay registers, a one-clock-wide pulse is produced.

V. LINAC LOCAL STATION SUPPORT [4]

SRMs are fully controlled from the Local Station. The Local Station code now supports Arcnet, while using Token-Ring to communicate with the rest of the control system.

The Local Station makes the existence of SRMs transparent to the outside world. When a setting is desired, the Local Station sends the request to the SRM and forwards the reply back to the requestor, as if the devices were connected to the Local Station itself. The Local Station responds to data requests with data from its own data pool.

Every Linac cycle (15 Hz) the Local Station broadcasts a cycle message to all the SRMs in its network. Each SRM acquiresthe current readings and sends a reply message back. The Local Station then analyzes the reply, updates its data pool, and scans for alarms. It takes five to twenty milliseconds for the data-acquisition reply to come back from an SRM, depending on the number of devices connected to it (including devices connected via daughterboards).
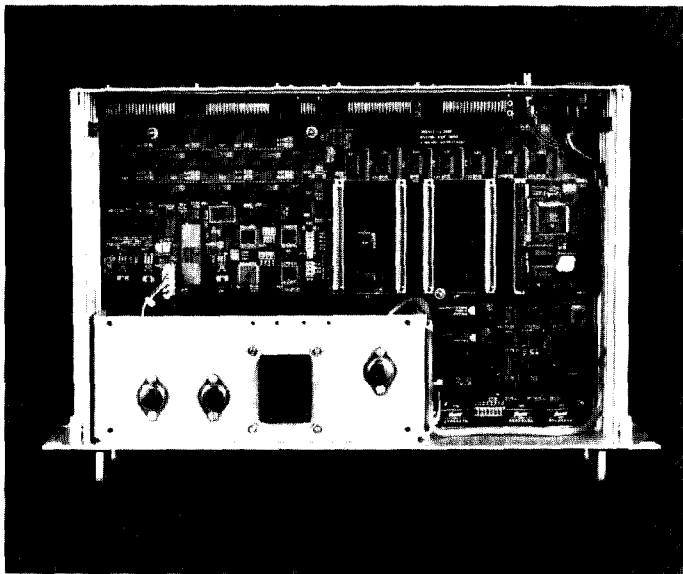
New versions of the SRM code are downloaded from a Local Station via Arcnet (instead of downloading the code through the serial port in each SRM, using the M68332BUG interface). The new version is first downloaded once from the host to a Local Station (via a serial port); then it can be sent to all the SRMs that are connected to any Local Station.

Since SRMs do not have a monitor connected to them, the Local Station console can be used instead. A special application program monitors an SRM's memory and enables changes to its contents from a Local Station. This feature is useful for setting up the SRM, telling it how many devices to access, and of what types, and for debugging.

VI. NETWORK INTERFACE [5]

The SRM is connected to the Local Station by an Arcnet local area network. Each Linac Local Station has a separate network with three or more SRMs. The physical media is coaxial cable in most cases. Fiber optics is used to connect to SRMs located in the Cockroft-Walton high voltage domes of the pre-accelerator. Arcnet was chosen because of its simplicity, low cost and the industrial support available. The limited message size (508 bytes) is sufficient in this implementation.

SRMs talk only with their Local Station and no messages are exchanged among them. SRMs do not initiate any message to the Local Station. The Local Station polls its SRMs for data acquisition buffers at the beginning of every 15 Hz Linac cycle. The SRMs poll the hardware and send back the up-to-

date values in a single Arcnet frame. SRMs also respond to multiple-reply requests, that require a reply every cycle. One shot requests and settings are also supported. A subset of the Acnet protocol [6] (developed in Fermilab) and a special "number 4"* protocol are used. The Acnet protocol is designed for data acquisition tasks and handles requests, replies and multiple-reply requests. The #4 protocol is an application layer that specifies the action to be taken by the SRM. The general-purpose Acnet protocol fits so well that no #4 protocol header is needed for replies from SRMs to Local Stations.

VII. SRM SOFTWARE [5]

The software for the SRM is written in C and cross developed on a VAX/VMS environment. The Microtec cross development tools are used.

A. Kernel

The kernel is Software Components Group's pSOS. pSOS is used in the Local Stations and therefore is the natural choice.

B. M68332BUG

The Motorola M68332BUG is an evaluation and debug tool for the MC68332. It comes with the BCC. It uses the BCC's serial interface, and allows the user to download and debug code for the MC68332. The M68332BUG was integrated into the SRM code (it is kept in PROM). The bug program has no knowledge of the kernel and the SRM code. At reset, it tests the MCU and the memory and transfers control to a pSOS initialization routine. One can easily switch to the bug program (when an SRM is running) by toggling a DIP switch which is checked every cycle. When in the bug program, one can download a new version of the SRM code, check and modify tables in the non-volatile memory, and even set chip selects and timers. When done, one can give the bug program a *go* command, and the SRM resumes running its code from the exact point where it was when the bug program took control (unless new code was downloaded). The ability to switch back and forth from the SRM code to the bug program is extremely useful for debug purposes.

The interface to the serial port provided by the bug program is also used to write values of selected channels to the serial port. This is useful in SRMs that are not close to a Local Station. Obviously, one has to connect a terminal or a portable computer to the serial port to see those readings.

C. Local Cycle

Normally the SRM polls the hardware for current values every time it receives a cycle message from the Local Station (at 15 Hz). In order to support data acquisition even when the connection to the Local Station is broken, the SRM has an internal 12.5 Hz cycle that is activated if the external cycle

message fails to arrive. The acquired readings can be seen by connecting a terminal to the serial port (see above).

D. Readings/Settings

The SRM code supports A-D readings, D-A settings, and digital I/O reading, settings, pulses, and motor control of the various A-D, D-A and digital I/O devices on the SRM's motherboard and on the various daughterboards.

VIII. COMMISSIONING

Sixty Smart Rack Monitors are on order. The first group is to be used with the upgraded Linac control system, expected later this year. The rest of the SRM modules are for the 400MeV Linac upgrade (for which the same Local Stations are used), several of which are to be used during testing and commissioning of the new accelerator.

IX. CONCLUSION

The Smart Rack Monitor designed in Fermilab is an example of a low-cost (both in hardware cost, and in engineering time for design and development) data acquisition module. Nevertheless, it is a very flexible device and can serve as platform for a variety of applications.

X. REFERENCES

[1] R. Goodwin, et al, "Initial Operation and Current Status of the Fermilab D0 VME-Based Hardware Control and Monitor System" in *Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems*, Vancouver, Canada, October 1989, pp. 125-129.
[2] M. Shea, "Smart Rack Monitor", internal note, 1990.
[3] E. McCrory, et al, "Upgrading the Fermilab Linac Control System", in *Proceedings of the 1990 Linear Accelerator Conference*, Albuquerque, New Mexico, September 1990, pp. 474-477.
[4] R. Goodwin "SRM Message Protocols (from the local station)", internal note, 1991.
[5] S. Shtirbu "Smart Rack Monitor Software", internal note, 1991.
[6] C. Briegel, et al, "The Fermilab Acnet Upgrade", in *Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems*, Vancouver, Canada, October 1989, pp. 235-238.

---

* This is the fouth data communication protocol currently supported by the Linac local control stations.

1486