

ARCNET as a Field Bus in the Fermilab Linac Control System

M.F. Shea, R.W. Goodwin, M.J. Kucera and S. Shtirbu
 Fermilab[†], P.O. Box 500
 Batavia, Illinois 60510

Abstract

Data acquisition hardware in accelerator control systems is connected by a field bus to networked computers that supply data to consoles. Industry attempts to standardize on a low level field bus have not succeeded in providing a single well-supported bus. This paper describes a data acquisition chassis that connects to VMEbus computers using ARCNET, a full featured token-passing local area network, as the field bus. The performance of this technique as implemented in the control system for the Fermilab Linac is given.

I. INTRODUCTION

A field bus is the connection between local computers and the interface hardware that controls accelerator equipment. Various field buses have been used by control system designers, but no single bus has emerged as a widely used standard. Example field buses are MIL-1553, Bitbus, and CAMAC.

This paper describes the use of ARCNET as a field bus in the Fermilab Linac Upgrade project. The output energy of the Linac injector is being increased from 200 MeV to 400 MeV by replacing the last four of the nine 200 MHz Alvarez tanks with 800 MHz side-coupled structures. This upgrade doubles the beam energy in the same accelerator enclosure. The control system being designed as part of the upgrade, will be installed for the preaccelerators and the remaining five 200 MHz tanks as well as the new 800 MHz sections.

II. ARCNET AS A FIELD BUS

If the interface hardware that connects to accelerator equipment is dumb, then the field bus must be register-based. Such buses have limited addressing capacity and the amount of data transferred per transmission is small. Nodes on register-based field buses usually operate as slaves that must be polled by a master computer to collect their data.

If the interface hardware contains a processor, then a local area network can be supported. Benefits of using a local area network include longer messages, higher data transmission rates, longer distance connections, and peer protocol communications that allow a remote node to source a message

without being polled. A single message may contain all the analog and digital data collected by a node.

Characteristics of ARCNET include:

- 2.5 MHz serial bit rate
- 508 byte maximum message length
- Bus, multiple star and daisy-chain topology
- Coax, twisted pair and fiber optic transmission media
- Maximum distance without repeaters >600 m. (coax)
- Transformer isolated cable drivers
- Deterministic token passing access protocol
- Built-in acknowledgment of successful message reception
- Network configuration and timeouts implemented in controller chips

Compared with the characteristics of ARCNET, typical field buses are much lower in performance: MIL-1553 operates at one MHz, has a maximum message length of 32 words and can address 32 locations per node; Bitbus is not transformer coupled and is limited to 375 kHz for reasonable distances; normal CAMAC commands access only three bytes. All of these buses use master-slave communication protocols.

III. ARCHITECTURE OF THE LINAC SYSTEM

The overall design of the Linac control system follows the standard architecture of local VMEbus 68020 computers networked to each other and to console workstations by an IEEE-802.5 Token Ring. A separate ring is used for the Linac VMEbus stations and the connection to the central control system is made by a commercial multiport Token Ring-to-Token Ring bridge. This organization isolates the traffic on the Linac ring from all other network traffic.

A. System Hardware

Individual VMEbus computers, called Local Stations, contain a core set of five cards; a 68020 processor, a 1MByte non-volatile RAM card, the Token Ring adapter, a crate utility card and an ARCNET adapter. All but the crate utility card are commercially available. Each VMEbus station acts as a stand-alone control system complete with its own database and software resident in the non-volatile RAM. Following a power outage, these systems reset, reconnect to the network, and send the last known settings to the hardware. Because they have non-volatile RAM, Local Stations do not require routine downloading. Local Stations run synchronously with the 15

[†] Work supported by the U.S. Department of Energy under contract No. DE-AC02-76CHO3000

Hz cycle rate of the Linac. Each cycle they read data from the hardware according to entries stored in a data access table, update a data pool of current readings, respond to requests for data by returning answers over the network, scan the newly acquired data for alarm conditions and send alarm messages to a multicast address on the network.

As with most control systems, the least standard part is the interface to the accelerator components. Accelerator equipment is usually sparsely scattered over a large area. The cabling length should be minimized for reasons of signal quality and cost, but the signal density is not high enough to justify the installation of VMEbus crates at all the locations where data is to be acquired. The solution is to collect equipment signals locally with cost effective interface hardware and to transmit this data to the VMEbus crates over a field bus. For the Linac Upgrade, a data acquisition chassis called the Smart Rack Monitor (SRM) collects data from the hardware and transmits it to the VMEbus Local Stations over the ARCNET field bus.

B. System Software

Network software was designed to support the Token Ring chip set. Network frames are received into a large circular buffer via DMA, and a pointer to each message contained within each frame is passed to the designated task using a standard message queue supported by the pSOS operating system kernel. Network messages to be transmitted are passed by reference through another message queue to the network transmitting logic. When the queue is flushed, consecutive messages destined for the same node are combined into a common frame if possible. This can greatly reduce network software overhead for both sender and receiver.

ARCNET support was added to the system software retaining the overall plan described above. Separate I/O buffers are used for each network. The ARCNET interrupt routine transfers the frame between the hardware buffer and system memory analogous to the Token Ring's DMA transfer. Emulation of the Token Ring software data structures permitted much of the network code to be common. A range of 16-bit node numbers is mapped onto ARCNET nodes so that user code sees no difference between the two networks. The task-task protocol widely used in Fermilab accelerator control systems allows for future protocol additions. A simple data acquisition request protocol was designed for use with the SRMs.

To support SRM data acquisition, three new entry types were designed for use in data access table entries. The first one results in sending a cycle request message, normally broadcast, to the SRMs. Each SRM that receives this message reads its own hardware data according to entries in its own data access table and replies with a single frame that includes all types of data. The second entry type waits for a given SRM to reply, and includes a deadline time within the 15 Hz cycle. One can

optimize data collection by ordering the wait-type entries so that lightly-loaded SRMs appear early in the table. The third entry type is used to map the various data types included in the SRM's response into the local station's data pool. Status bits are set or cleared according to whether a given SRM replied in that cycle. These status bits can be enabled for alarm reporting if an SRM stops sending replies.

IV. SMART RACK MONITOR

The SRM is an extension of the MIL-1553 Rack Monitor (RM) used in the DZero colliding beams detector. In that case, the RM was dumb, so the register-based field bus protocol MIL-1553 was used. For the Linac Upgrade, it was decided to use the existing digitizers and D-A converters. Operating these devices requires that a computer be located close to the equipment thus requiring a processor in the rack monitor.

A. SRM Hardware

The SRM, shown in Figure 1, is a 2U chassis that contains 64 16-bit differential A-D channels, 16 12-bit D-A channels, 8 bytes of digital I/O, 256K bytes of nonvolatile RAM, the ARCNET interface and a processor. The processor is on a Motorola *Business Card Computer* (BCC) that includes an MC68332 microcomputer, 64k bytes of RAM, 128k bytes of ROM, and a serial port. The BCC is marketed by Motorola as part of an MC68332 evaluation kit. Three BCC sockets were included on the SRM motherboard: two for expansion daughterboards and one for the BCC. Daughterboards used in the Linac system will be for special equipment interfaces and for the timing system that is based on a 10 MHz serial event clock. A BCC and a 9-byte digital I/O daughterboard are shown in Figure 2.

ARCNET was chosen for the SRM network for several reasons: its cost is low, its 2.5 MHz serial bit rate is adequate, the physical bus/logical ring architecture makes it easy to connect and the token passing protocol eliminates collisions. The automatic network configuration minimizes processor overhead, the *free buffer enquiry* and *message received* features of the protocol makes the communication reliable. Silicon support for ARCNET is excellent.

B. SRM Software

At this time, the software in the SRM is limited to data acquisition and debugging, although future applications may include additional tasks such as closed loop regulation. Software in the SRM is written in C and uses the pSOS real time operating system kernel. Motorola's 332Bug program that is bundled with the BCC is retained for low level debugging.

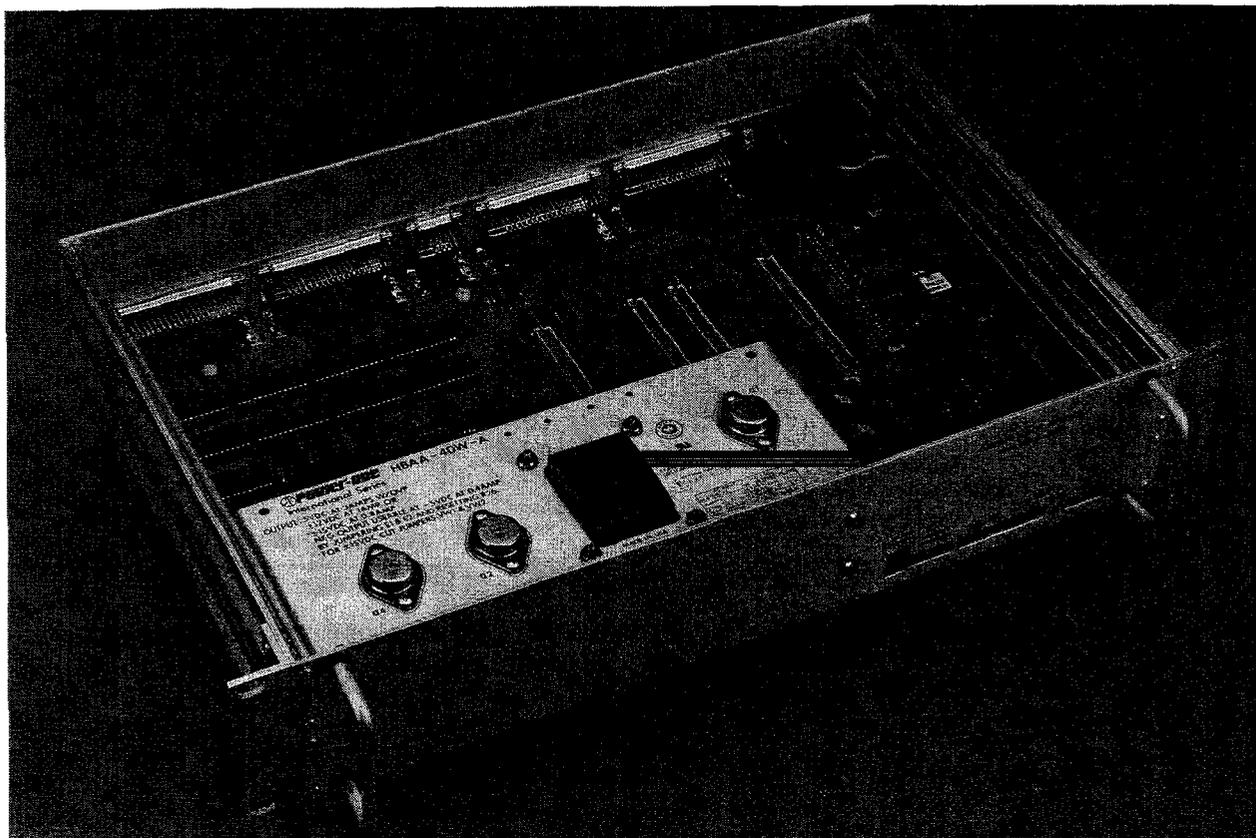


Figure 1. Photograph of a Smart Rack Monitor

When the SRM is reset, 332Bug initializes the MC68332 chip selects and then enters the SRM system code where it executes a repetitive cycle that collects data and prepares a message to send to the VMEbus Local Station over ARCNET. Normally, the cycle is initiated by the ARCNET broadcast message sent by the Local Station, but an internal M68332 timer is used to trigger a cycle, if no ARCNET message is

received, to permit stand alone testing.

A copy of the system code resides in the 128 kbyte PROM on the BCC, and another copy may be stored in the non-volatile RAM of the SRM motherboard. During initialization, if a RAM version is found, that copy is used. If no system is stored in RAM, the PROM version is copied to RAM. The MC68332 contains a hardware watchdog timer that requires refreshing at regular intervals. If a timeout occurs, the PROM version of the SRM code is copied to RAM and operation resumes. New versions of SRM code can be downloaded over ARCNET. Because some version of the SRM program is available in PROM, the SRM will function at least well enough to allow downloading the current version of the software. In normal operation, the SRM will find the current version in RAM.

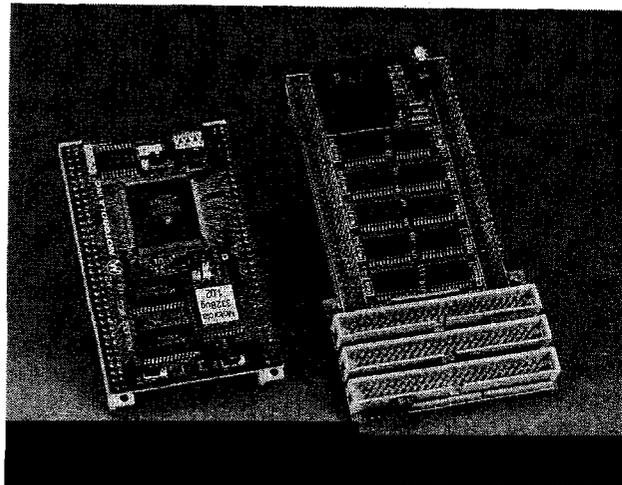


Figure 2. Photograph of a Business Card Computer and a 9-byte Digital I/O Daughterboard

The SRM determines what data to collect by referring to a data access table containing 16-byte entries that describe the type and amount of data to read. Separate data types are defined for digital I/O, A-Ds, D-A readback and so on. After all data is read, the SRM replies to the Local Station's broadcast request. Note that all SRMs connected to a Local Station, usually three or four, acquire their data in parallel and the Local Station only receives one message from each SRM.

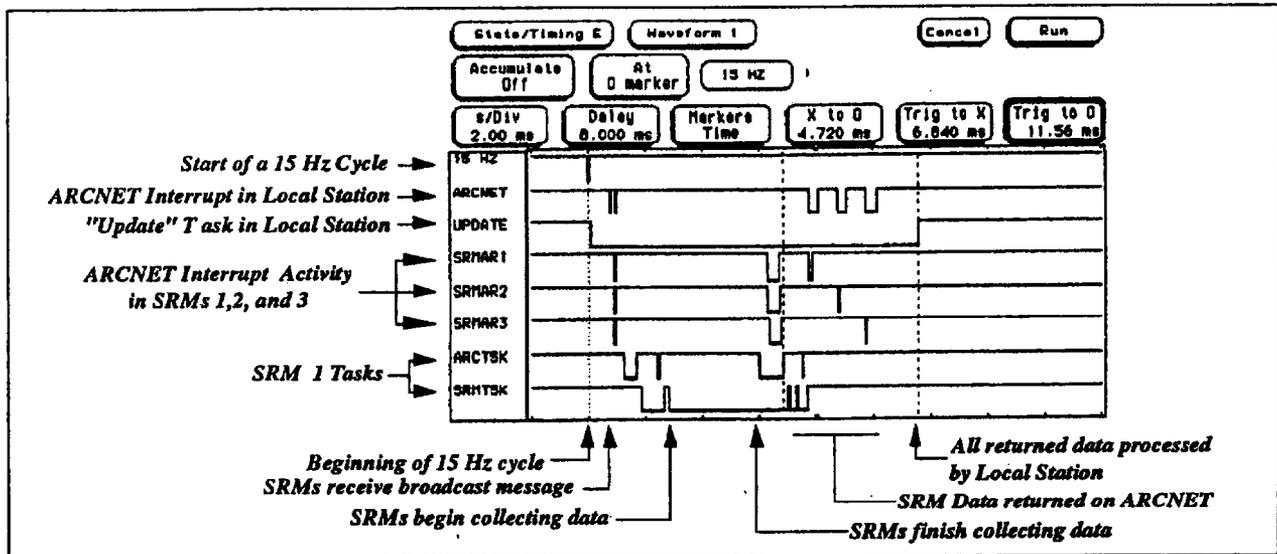


Figure 3. Logic State Analyzer Timing Display of SRM Operation

Data types are included to read back setting values of digital and analog output in the message returned to the Local Station. This is done to allow for future closed loop programs that modify output settings locally within the SRM.

V. PERFORMANCE AND TIMING

Figure 3 shows a logic analyzer display of the timing characteristics of a Local Station and its three Smart Rack Monitors. In response to the 15 Hz interrupt, the Local Station broadcasts an ARCNET message to the SRMs to initiate data collection. After 6.8 ms, the SRMs are ready to return their data. At 11.6 ms into the 66 ms cycle, data from all three SRMs has been received and processed by the Local Station. For this test, all SRMs were reading the onboard 64 A-D channels, the 8 bytes of digital I/O, and the readback of 16 D-A settings. Because the SRM activity is overlapped, an additional SRM would add only the ARCNET serial time plus the Local Station processing time, about 1.1 ms per SRM.

VI. PRESENT STATUS AND DISCUSSION

The Linac Upgrade is to be carried out in two parts; the replacement of the existing control system scheduled for January 1992, and the upgrade of the Linac itself, scheduled for late 1992. The SRM was prototyped and then fabricated in quantity by an outside vendor. Sixty-five SRMs have been delivered.

The concept of a networked data acquisition chassis that satisfies most of the interface requirements of the accelerator seems to work well. A single interface design that is duplicated in quantity results in a cost effective solution to the distributed data acquisition problem. Because of the single daisy-chained coaxial cable ARCNET connection, the SRMs can be installed wherever the data sources are located. If new

devices are installed, another SRM can be easily added to read and control the additional equipment.

Data collection by SRMs is efficient because all SRMs collect their data in parallel. The VMEbus computer broadcasts a single command over ARCNET to initiate the data acquisition cycle and then processes the SRM messages as they are received.

Because it is a widely used network, ARCNET enjoys considerable industry support. A commercial copper-to-fiber optic ARCNET bridge is used to connect the VMEbus ARCNET adapter to the SRMs in the two 750 KV equipment domes of the preaccelerators. SRMs in the domes have the ARCNET copper transceivers replaced with pin-compatible fiber optic transmitter/receivers.

The connectivity provided by ARCNET allows future tasks to be handled by the SRMs. Because of the peer protocol of the network, one SRM could ask for and receive data from other SRMs. The other SRMs can be connected to the same or different Local Stations. Using this feature, an embedded control loop could be run in one SRM with data being collected from several other SRMs.

VII. ACKNOWLEDGMENTS

Several people contributed to the hardware development of this project. In particular, Robert Florian has been in charge of the VMEbus systems, and the early SRM design was done by Alan Jones, now at the SSC laboratory.