DISTRIBUTED SIGNAL MULTIPLEXOR at FERMILAB

Charlie Briegel, Dean Still
Fermilab
Accelerator Division Controls Department
MS 347
P.O. Box 500
Batavia, IL. 60510 USA

A set of 1 GHz multiplexors is distributed at remote locations which enable signals to be routed to a variety of instrumentation equipment. The commercial multiplexor provides up to 20 1X4 switches per location and can be controlled via GPIB.

Control of the multiplexors utilizes MOOC (Minimal Object Oriented Communications) which provides the skeleton software for communication to an application program. The control of a route across distributed multiplexors can be effected by turning ON a device. The route has various mechanisms for reserving or usurping the ownership of the multiplexors which make up the signal's route. The application and front-end software implementation is described.

The hardware consists of a VME crate, a 68040 processor running vxWorks OS, and GPIB interfaces including transparent support of a remote GPIB connected via Ethernet to the front-end computer.

1. OVERVIEW

A reoccurring goal in the control room is to measure a signal on the appropriate scope for a test. A distribution panel with a maze of wires is commonly used to acquire this signal. The cable is manually connected to the scope from the source which may span multiple distribution panels from remote locations. While this method works for stable connections or short-term tests, it does not adequately address today's dynamic needs for acquiring various signals to a shared set of instrumentation. In this case, depending on the sequence of events, a different set of signals need to piped to a set of instruments as the sequence evolves and then data is acquired. A reliable and efficient method for distributing these signals is required.

A commercial multiplexor was purchased which provided a variety of options. Fermilab purchased only one type of multiplexor card; a 1 GHz 1X4 multiplexor with 4 sets of multiplexors per card, 5 card slots per box, a total of 20 1X4 1 GHz multiplexors per box. The control system interface is via a GPIB connection and utilized existing software and hardware infrastructure to manipulate the switches. Four boxes were acquired; two were located in the Main Control Room (MCR), one was located in Booster in the RF area, and one was located in the Main Ring in the RF area.

Minimal Object Oriented Communications (MOOC) is the underlying software used to implement support for GPIB communications. MOOC hides the layers of software required for communications and invokes methods to fulfill data acquisition requests and settings from the control system.

2. MOOC

This infrastructure enables classes to be individually loaded and initialized so a system can be individually tailored. Although the classes start independently, the GPIB class is inherited, so backward references will be made to this base class. Classes which inherit the GPIB class can provide hidden functionality to the user, or derived calculations from GPIB devices. MOOC and the pre-existing GPIB class minimizes the effort needed to create the MULTIPLEXOR class and enables the user to concentrate on the problems of interfacing to a particular GPIB device. The following is a portion of the startup script for the vxWorks node.

```
shellPromptSet ("GPIB->")
# a bunch of loads were deleted for this document...
# specify the controller type per 0-31 gpib address range
# and IP host for ENET boards
GPIB_obj("NI_1014","")
GPIB_obj("ENET_GPIB","host_name")
#Create the gpib class
GPIB_obj_init()
# specify the gpib address of the MUXes
MULTIPLEXOR_obj(6)
MULTIPLEXOR_obj(7)
MULTIPLEXOR_obj(44)
#Create the multiplexor class
MULTIPLEXOR_obj_init()
#  Slam:  wake up VAX task for download of MUX settings
MiscBoot()
```

3. GPIB Class

A class was written to support up to eight separate GPIB controllers from a single VME processor. The supported controllers reside on the VME bus, an Industry Pack (IP) interface, or Ethernet through an Ethernet to GPIB interface. The Ethernet to GPIB controller enables distributed connectivity from a central location via a TCP/IP connection. This connection is treated as another GPIB bus controller and the network connection is maintained inside the driver interface.

The GPIB class supports transparent access to the device by enabling the console user to send and receive strings of data which is normally ASCII text. The class provides semaphore protection so only one user can access a GPIB controller at a given time. The semaphore provides hardware encapsulation of a single command. The sequence of commands to the device are not protected by the class and no logical ownership of the device is provided. When protection is required, the application is set to single-user execution.

The GPIB class provides control so an application can reset the controller, clear the GPIB device, trigger the device, set the REN line for the GPIB bus, or put the device in LOCAL. Digital status is returned to reflect the serial poll of the device.

4. MUTIPLEXOR Class

The multiplexor class effectively inherits the GPIB class and adds functionality. While the GPIB class is still available to the user, this new class and its corresponding devices provide a higher level of functionality. The user is unaware of the GPIB commands required for reading, setting, control, or status of the device.

There are three types of devices associated with this class.

1. The 1X4 multiplexor contains 4 devices which can be closed (ON) or opened((OFF). The multiplexor provides BBM (break before make), so closing a device will automatically open all the other three devices associated with the multiplexor. The status will return the state of the device. Currently, these devices can be manipulated without any constraints. All changes to the device are saved internal to the box and are recalled at initialization time of the class. This device type is intended for diagnostics.

2. The status of all the switches can be read by a single device which uses a nibble to represent the status of the four switches. So five 16-bit words are returned, one for each card in the box. This device cannot be set and is intended for status display of the box. The device can be reset, which places all the switches into a previously saved state.

3. The set of multiplexors can be manipulated by setting a device to ON. This device has a data structure associated with it which specifies a route for a signal to get to an instrument. The route can be read or set as an array of four byte structures. There can be up to 512 such devices, consisting of up to 20 multiplexors per route specification. Further, the device can return the status and ownership of each multiplexor specified in the route in an array of structures.

When a device is turned ON, the route is searched to determine that no other device of this type is currently using any of the multiplexors. If all the multiplexors are available, then the path is created by sequentially closing the multiplexors which make up the route. At the same time, the user can request ownership of the device for 5 minutes, 15 minutes, 1 hour, 8 hours, forever, or go to war with other users (LIFO). If the device is not available, the application can raise its priority by sending a request to turn ON the device as a super-user and over-ride any previous ownership. When the device is turned OFF, the path remains closed until someone requests one of the multiplexors The OFF command simply implies the multiplexors in this device's route are available. A reset to the device opens all multiplexors along the route.

This class provides ownership and easy manipulation of a set of multiplexors. A software program such as a sequencer can simply turn ON this type of device, collect the data for the signal on the instrument, then move on to the next measurement.

5. USER INTERFACE

A console user interface layer provides access to the GPIB class device by specifying only the node and address, where the address specifies both the controller (0-7) and the GPIB device address (0-30). The console user interface finds the corresponding device name and utilizes generic communication mechanisms to manipulate the device. The user needs to know where the device is located. Existing applications for GPIB devices allow commands to pass to the device without any programming and can easily be adapted to provide specific support for a device.

The MULTIPLEXOR class supports generic data acquisition services in the Fermilab control system. This enables these devices to appear as generic parameters for pre-existng applications. The underlying GPIB commands to manipulate the multiplexor are hidden from the user.

A specific application was written to specify and display the multiplexors associated with the route for a given device. The device name for each multiplexor was used to specify the route. When the device's route is set, the information is automatically forwarded by MOOC to the central database. When the processor is booted, the data base is downloaded automatically. The following shows the application displaying a route specification.

6. COMMENTS

The implementation currently consists of approximately 40 signals and approximately 12 scopes. The Main Ring RF box is not connected to the control room, due to a limited number of wires between RF and MCR. The system should expand when more operational experience is acquired.

The physical routing of signals to scopes and maintaining these paths is not an easy task. The maze of wires is **not** significantly reduced and the wires must still be connected appropriately to the multiplexors. This multiplexor does not provide any testing mechanism to understand the connectivity. The visualization of the connections will be enhanced by using a graphical interface for documenting the routes.

The Ethernet to GPIB controller enables distributed boxes to be centrally controlled. This is an essential part in providing ownership and enabling the routes to be accurately controlled.

The protection of the system could be improved The multiplexors can be manipulated directly without any protection thorough the GPIB class or the first device type of the MULTIPLEXOR class. The intended method to manipulate the multiplexors is through the last device type of the MULTIPLEXOR class which is protected. If necessary, all other mechanisms could be blocked or provide feedback to the existing protection mechanism.

There is no mechanism to prevent an inappropriate signal (i.e. high voltage) to be routed to an instrument. The ease by which the signal can be routed increases the possibility of causing damage to an instrument. A digital volt meter (DVM) accessible within the multiplexor could characterize the signal and perhaps prevent inappropriate connections. It might be interesting to use the multiplexor to mulitplex a DVM to test signals before they are switched onto equipment.

References

[1] RACAL-DANA Series 1250 Universal Switch Controller, May 1993, RACAL-DANA Publication No. 980609.
[2] ESP-488 Software Reference Manual for the GPIB-ENET, Feb., 1995, National Instruments Part Number 320910A-01.
[3] User Manual IP-488, Green Spring Manual Revision ii.
[4] Boris Lublinsky, MOOC, Fermilab internal document.
[4] Jeff Utterback, Making Data Accessible to ACNET from a VxWorks Front Entd, Fermilab internal document.
[5] B. Lublinsky, Nuclear Instruments & Methods in Physics Research A 352 (1994) 403-406 North-Holland.