Firewire Ready for Instrument and Control Applications

By Gary Sakmar
Keithley Instruments, Inc.

The 1394 Trade Association (1394TA) has published two new data communication protocols that facilitate the use of IEEE-1394 (Firewire) for measurement and control applications. The introduction of these protocols opens up new possibilities for system developers by providing efficient, high-speed communication between PC controllers and electronic instrumentation and control devices.

There are two major advantages of connecting measurement and control devices using industry standard I/O, such as that defined by Firewire:

1. **Cost** -- Standard I/O connectors and cables are less expensive than those using proprietary designs, which usually are produced in lower volumes.

2. **Ease of use** -- You simply plug standard, familiar cables into connectors that already exist on the PC. In Firewire's case, there is no need open up the computer to install a host bus adapter and you do not have to deal with hardware or software configurations for such an adapter.

Within the framework of the general Firewire specification, there is the opportunity for data communication rates up to 400MHz, with 1GHz planned for the near future. However, until now, system designers were in a quandary on how best to implement use of Firewire. What they needed was a lightweight protocol that would make it easy to implement these high-speed communications.

The Industrial Instrument and Control Working Group of the1394TA, which consisted of members from Keithley Instruments, 3A International, Agilent and National Instruments, solved this problem by sponsoring development of two protocols that facilitate use of IEEE-1394 in measurement and control applications. Protocol 1394TA IICP-1.0 details the basic methodology for asynchronous communications to electronic instrumentation and control devices. Protocol 1394TA IICP488-1.0 describes use of the first protocol to communicate IEEE-488.1 and -488.2 messages and command/control sequences on a Firewire bus. This facilitates the use of GPIB (SCPI) commands with Firewire.

(These protocols can be downloaded as PDF files from the 1394TA web site at www.1394TA.org/Download/Technology/Specifications/iicp1.0.pdf and www.1394TA.org/Download/Technology/Specifications/iicp4881.0.pdf.

The 1394 serial bus interface is already becoming a standard offering on some PCs, and many more manufacturers plan on providing it to meet I/O needs for consumer electronics. Instrument and control system developers are now taking greater interest in Firewire, not only because of its bandwidth, but also because of the following advantages:

- Self-configured addressing (users do not have to set address switches and there is no potential for address conflicts)
- A tiered-star topology allowing up to 63 connected devices

Implementation of IICP Communications

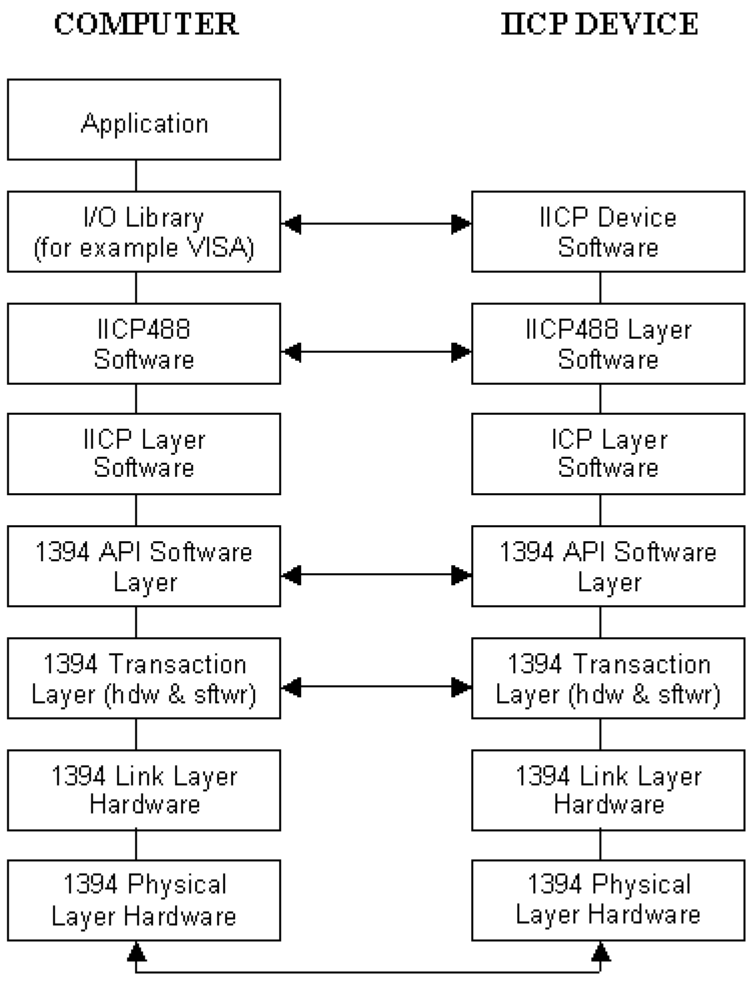

Figure 1 illustrates the IICP488 communication model with the IICP488 protocol layered above IICP. Developing measurement and control systems with IICP and Firewire should sound familiar to anyone who has done similar work with other data communication protocols. Implementation is uncomplicated.
COMPUTER IICP DEVICE
Figure 1. IICP488 Communication Model

An IICP connection consists of IICP plugs on the computer and on the hardware devices. A plug contains two ports: one port typically is used to send and receive data frames while the other port is used to send and receive control frames. (A frame is a set of logical, contiguous data.) An example of a data frame, when using IICP488, is a SCPI command or response. An example of a control frame in IICP488 is an SRQ packet. Each plug port has device memory mapped to 1394 space.

For those using GPIB devices in their systems, another advantage of Firewire and IICP488 is the ability to reuse major chunks of application code. As alluded to above, sending and receiving GPIB messages is done through a data port. Since data port messages are pure data, there is nothing that needs to be stripped off and no message parsing is required. An IICP488 data message is received and acted on just as if it were transmitted over a GPIB bus. Only the transaction and physical layers of the communication system have changed.

Many implementation details of IICP are left to the developer. However, implementing a 1394 API for a PC platform typically is a matter of obtaining one from a third party source, or writing your own as a layer above the Microsoft 1394 bus class driver. In either case, the API should provide the means to:

- Send, compare and swap the lock transactions and responses
- Write quadlet transactions
- Write block transactions
- Map buffers and plug registers to 1394 space
- Receive notification when these are accessed by the connected node

Node Management

Any node can act as an IICP connection manager to establish plug connections. This is done by first locking connection registers and then sending a sequence of connection request packets. Each packet contains information needed by the hardware device to determine the address of the plug created on the connected node. By using this method, an IICP connection client never has to process simultaneous connection requests from multiple threads or multiple controllers. The connection manager also is responsible for maintaining the connection in case of bus resets, new node addresses appearing or new IICP devices being added.

Firewire bus resets are completely handled by the IICP layer. Discovery of Firewire IICP devices is accomplished with software that goes through a process of determining the protocol a particular device understands. The IEEE 1394-1995 and IEEE 1212 specifications define the use of a configuration ROM for this purpose.

IICP recognizes there are many devices, such as sensors and low-cost D/A converters, that use memory mapped I/O. Therefore, a simple interrupt mechanism is defined for these devices, which in some instances may eliminate the need for polling by a controller. Also, Firewire defines a 48-bit address space, some of which a memory-mapped I/O device can use for device-dependent functions.

All these features are designed to make Firewire use as easy as possible. It is a communication bus worthy of consideration when you need higher data rates and a simple method of interconnecting measurement and control devices.

Gary Sakmar is Development Group Manager for Keithley Instruments, Inc. where he is responsible for new technology and software development. He also is co-chairman of the Industrial Instrument and Control Working Group (IIWG) of the 1394 Trade Association. He has over 15 years of experience in test, measurement and data acquisition systems.