VMEbus Extensions for Instrumentation

TCP/IP-IEEE 488.2 Instrument Interface Specification

VXI-11.3

Draft 0.3

July 17, 1995

NOTICE

The information contained in this document is subject to change without notice.

The VXIbus Consortium, Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The VXIbus Consortium, Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

VMEbus Extensions for Instrumentation TCP/IP-IEEE 488.2 Instrument Interface Specification VXI-11.3, Draft 0.3 is authored by the VXIbus Consortium, Inc. and its sponsor members:

GenRad, Inc.
Hewlett-Packard Co.
National Instruments, Corp.
Racal-Dana Instruments, Inc.
Tektronix, Inc.
Wavetek, Inc.

Table of Contents

List of Tables

List of Figures

VMEbus Extensions for Instrumentation:

TCP/IP-IEEE 488.2 Instrument Interface Specification

A.INTRODUCTION

The need to connect instruments to computer networks has developed in the test and measurement industry. The connections required may be to either local-area networks (LANs) or wide-area networks (WANs). This specification, which is part of the VXIbus set of specifications, describes how instrumentation can be directly connected to industry-standard networks. The communications and programming paradigms supported by this specification are similar in nature to the techniques supported by IEEE 488.2 devices. The methods described allow ASCII-based communications between a controller and a device over a TCP/IP network. The reader should be knowledgeable about networks, the Internet Protocol Suite, ONC RPC, and IEEE 488.2.

A.1. SCOPE

This specification is part of the VXIbus set of specifications and defines a TCP/IP-IEEE 488.2 Instrument Interface.

The only networks directly considered by this specification are those which support the Internet Protocol Suite. The techniques defined in this specification could be used over other networks, such as networks which support the OSI protocol standards, but this document does not address that mapping. This specification uses Open Network Computing (ONC) remote procedure calls on top of the Internet Protocol Suite.

Other network protocols may also be supported by a TCP/IP-IEEE 488.2 Instrument Interface.

A.2. DOCUMENT STRUCTURE

This document is divided into two sections. The first section, an introduction, is intended to familiarize readers with the intent and scope of the document.

The second section, TCP/IP-IEEE 488.2 Instrument Interface, defines the operation of a TCP/IP-IEEE 488.2 Instrument Interface , including the mapping between the network transactions defined by VXI-11, TCP/IP Instrument Protocol Specification, and the internal operation of an instrument.

A.3. SPECIFICATION OBJECTIVES

In addition to the objectives in VXI-11, this specification has the following objectives:

1. To define the operation of a TPC/IP-IEEE 488.2 Instrument Interface.
2. To define a mapping from network transactions to internal instrument operations.

A.4. DEFINITION OF TERMS

This document uses the same definitions as VXI-11.

The following terms are used to identify the contents of paragraphs, as in other VXIbus Specifications. These definitions are the same as those in IEEE 1155-1992.

RULE: Rules **SHALL** be followed to ensure compatibility for cards in the system. A rule is characterized by the use of the words **SHALL** and **SHALL NOT**. These words are not used for any other purpose other than stating rules.

RECOMMENDATION: Recommendations consist of advice to implementors which will affect the usability of the final device. Discussions of particular hardware to enhance throughput would fall under a recommendation. These should be followed to avoid problems and to obtain optimum performance.

PERMISSION: Permissions are included to clarify the areas of the specification that are not specifically prohibited. Permissions reassure the reader that a certain approach is acceptable, and will cause no problems. The word **MAY** is reserved for indicating permissions.

OBSERVATION: Observations spell out implications of rules and bring attention to things that might otherwise be overlooked. They also give the rationale behind certain rules, so that the reader understands why the rule must be followed.

Any text that appears without a heading should be considered as description of the standard.

A.5. REFERENCES

This specification references the following documents in addition to those referenced by the TCP/IP Instrument Protocol Specification, VXI-11:

[1] IEEE Std 488.2-1992, IEEE Standard Codes, Formats, Protocols, and Common Commands For Use With IEEE Std 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation.

A.6. RELATED DOCUMENTS

This specification is one document in a set of specifications which describe a method for ASCII-based communication over a network between controllers and devices. This specification describes the mapping from the protocol to internal instrument operations, along with the operation of a TCP/IP-IEEE 488.2 Instrument Interface. Other specifications in the group describe the protocol itself as well as the mapping from the protocol to other interface types. Those specifications listed below are currently part of this group:

[1] VMEbus Extensions for Instrumentation: TCP/IP Instrument Protocol Specification, VXI-11.
[2] VMEbus Extensions for Instrumentation: TCP/IP VXIbus Interface Specification, VXI-11.1.

[3] VMEbus Extensions for Instrumentation: TCP/IP IEEE 488.1 Specification, VXI-11.2.

B. TCP/IP-IEEE 488.2 INSTRUMENT INTERFACE

The TCP/IP-IEEE 488.2 Instrument Interface converts *network instrument* messages into internal instrument communication protocols. It allows controllers to control instruments connected directly to the network via the TCP/IP-IEEE 488.2 Instrument Interface.

OBSERVATION B.1:
A TCP/IP IEEE 488.2 Instrument Device executes RPCs in the order received. See VXI-11, B.2.1.

RECOMMENDATION B.1:
A TCP/IP-IEEE 488.2 Instrument Interface should support two or more *network instrument* servers simultaneously.

RULE B.1:
A TCP/IP-IEEE 488.2 Instrument Interface **SHALL** support at least two concurrent links per *network instrument* server.

RECOMMENDATION B.2:
The number of *network instrument* servers and links supported by a TCP/IP-IEEE 488.2 Instrument Interface should be based on available resources, not on arbitrary predetermined limits.

Figure B.1 shows a typical TCP/IP-IEEE 488.2 Instrument system. The box labeled Message Exchange Interface provides the same functionality as described in IEEE 488.2, 6. Likewise, the functionality provided by the box labeled Status Reporting is the same as defined in IEEE 488.2, 11.

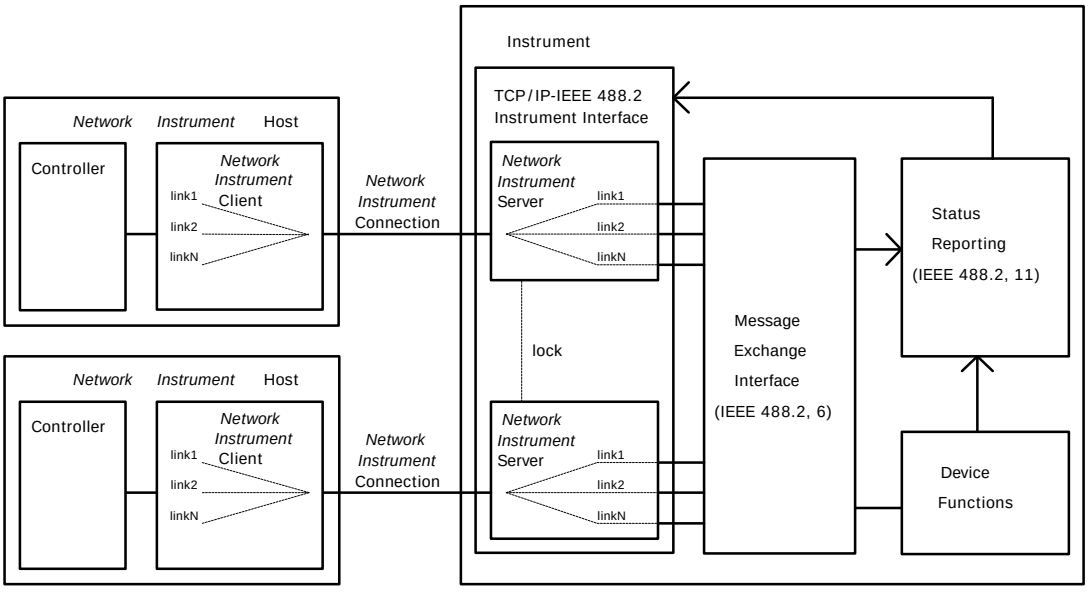

Figure B.1 Typical TCP/IP-Instrument System

B.1. DEVICE STRING FORMAT

The contents of the device string determines which Message Exchange Interface is associated with a link. This structure allows multiple instruments to use the same IP address.

RULE B.1.1:

A TCP/IP-IEEE 488.2 Instrument Interface **SHALL** support a *device* string of the following format:

```
<inst_name>
```

where:

<inst_name> A name corresponding to a single instrument.

RULE B.1.2:

The TCP/IP-IEEE 488.2 Insturment Inteface **SHALL** recognize an <inst_name> of "inst0" as the first or only instrument within the TCP/IP-IEEE 488.2 Instrument Interface. Additional instruments **SHALL** be identified by "instN" where N is a non-negative integer assigned sequentially beginning at one.

RULE B.1.3:

One and only one Message Exchange Interface and Status Reporting **SHALL** exist for each instN supported in a TCP/IP IEEE 488.2 Instrument Interface even when connections to multiple controllers are active.

OBSERVATION B.1.1:

A *network instrument* server could potentially support links using protocols described in related documents. See A.6. For example, it might be able to convert *network instrument* protocols to IEEE 488.1 operations as well as to internal instrument operations. In such a case, the allowed contents of the *device* string depend on the total capability of the *network instrument* server.

B.2. LAN FUNCTIONS

RULE B.2.1:

A TCP/IP-IEEE 488.2 Instrument Interface **SHALL** support the *network instrument* protocol, and **SHALL** accept and execute all of the RPCs defined by that protocol. Support of the *network instrument* protocol includes support for the entire protocol stack defined for use by *network instrument* devices.

PERMISSION B.2.1:

A TCP/IP-IEEE 488.2 Instrument Interface **MAY** support other LAN protocols.

RULE B.2.2:

A TCP/IP-IEEE 488.2 Instrument Interface **SHALL** support an Ethernet/802.3 Data Link Layer and an 802.3/10BASE-T Physical Layer with an RJ-45 connector for 10BASE-T.

OBSERVATION B.2.1:

The intent of this rule is that the default configuration of the TCP/IP-IEEE 488.2 Instrument Interface provides an RJ-45 connector for 10BASE-T. Other connectors may be supported, and may replace the RJ-45 connector.

PERMISSION B.2.2:

A TCP/IP-IEEE 488.2 Instrument Interface **MAY** support other protocols at the Data Link and Physical layers.

Application	*Network Instrument* Protocol
Presentation	XDR
Session	ONC/RPC
Transport	TCP
Network	IP
Data Link	Ethernet/802.3
Physical	802.3/10BASE-T

Figure B.2 TCP/IP-IEEE 488.2 Instrument Interface Protocol Stack

The resulting protocol stack is shown in Figure B.2.

B.3. LINK MANAGEMENT

This section covers the operations which manage links or operations on links. These operations are implemented primarily within the TCP/IP-IEEE 488.2 Instrument Interface itself.

B.3.1. Link Creation

The *create_link* and *destroy_link* RPCs do not directly map to any internal operations.

RULE B.3.1:

The *create_link* and *destroy_link* RPCs **SHALL** be implemented entirely within the TCP/IP-IEEE 488.2 Instrument Interface . The TCP/IP-IEEE 488.2 Instrument Interface **SHALL** maintain links as defined by the *network instrument* protocol.

B.3.2. Interrupt Channel Operation

The *create_intr_chan* and *destroy_intr_chan* RPCs does not directly map to any internal operations.

RULE B.3.2:

The *create_intr_chan* and *destroy_intr_chan* RPCs **SHALL** be implemented entirely within the TCP/IP-IEEE 488.2 Instrument Interface.

B.3.3. Device Locking

The *device_lock* and *device_unlock* RPCs do not directly map to any internal operations.

RULE B.3.3:
The *device_lock* and *device_unlock* RPCs **SHALL** be implemented entirely within the TCP/IP-IEEE 488.2 Instrument Interface. The TCP/IP-IEEE 488.2 Instrument Interface **SHALL** maintain locks as defined by the *network instrument* protocol.

B.3.4. Abort Operation

The *device_abort* RPC does not directly map to any internal operations.

RULE B.3.4:
The *device_abort* RPC **SHALL** be implemented entirely within the TCP/IP-IEEE 488.2 Instrument Interface to abort any active operations associated with the link.

B.4. INSTRUMENT COMMUNICATIONS

This section covers the operations which communicate directly with the internal operations of the instrument. In addition to the actions specified, the RPCs have interaction with other aspects specified by IEEE 488.2, see B.5.

B.4.1. Data Transfer from Interface to Instrument

RULE B.4.1:
When the TCP/IP-IEEE 488.2 Instrument Interface receives a *device_write* RPC it **SHALL** transfer the bytes in the *data* parameter into the Input Buffer of the Message Exchange Interface. If the end flag is set in the *flags* parameter, then the last byte **SHALL** have an END associated with it.

RULE B.4.2:
If the Input Buffer becomes full during the transfer operation, the TCP/IP-IEEE 488.2 Instrument Interface **SHALL** wait until room becomes available.

OBSERVATION B.4.1:
Waiting for room in the Input Buffer may cause the *device_write* RPC to timeout.

OBSERVATION B.4.2:
The associated action of *device_write* is complete when the contents of *data* have been transferred into the Input Buffer. The action does not depend on the contents being processed.

B.4.2. Data Transfer from Instrument to Interface

RULE B.4.3:
When the TCP/IP-IEEE 488.2 Instrument Interface receives a *device_read* RPC, it **SHALL** transfer bytes from the Output Buffer in the Message Exchange Interface into the *data* response parameter until a termination condition is encountered.

RULE B.4.4:
If the Output Queue becomes empty during the transfer operation, the TCP/IP-IEEE 488.2 Instrument Interface **SHALL** wait until more data becomes available.

OBSERVATION B.4.3:
Waiting for more data in the Output Queue may cause the *device_read* RPC to timeout.

B.4.3. Device Clear Operation

RULE B.4.5:
When the TCP/IP-IEEE 488.2 Instrument Interface receives a *device_clear* RPC, it **SHALL** send a **dcas** message to the Message Exchange Interface.

RULE B.4.6:
The *device_clear* RPC **SHALL NOT** return until all the operations associated with **dcas** are complete. See B.5.2.

B.4.4. Trigger Operation

PERMISSION B.4.1:
When the TCP/IP-IEEE 488.2 Instrument Interface receives a *device_trigger* RPC, it **MAY** perform no action and return with error set to 8, operation not supported, indicating it does not support *device_trigger*.

RULE B.4.7:
It the TCP/IP-IEEE 488.2 Instrument Interface supports the *device_trigger* RPC, when the TCP/IP-IEEE 488.2 Instrument Interface receives a *device_trigger* RPC, it **SHALL** put a GET token into the Input Buffer of the Message Exchange Interface.

RULE B.4.8:
If the Input Buffer is full so it cannot accept the GET token, the TCP/IP-IEEE 488.2 Instrument Interface **SHALL** wait until room becomes available.

OBSERVATION B.4.4:
Waiting for room in the Input Buffer may cause the *device_trigger* RPC to timeout.

OBSERVATION B.4.5:
The associated action of *device_trigger* is complete when the GET token has been transferred into the Input Buffer. It does not wait for any action associated with the GET to start or finish.

B.4.5. Remote/Local Operation

RULE B.4.9:
When the TCP/IP-IEEE 488.2 Instrument Interface receives a *device_remote* RPC, it **SHALL** disable local operation of all programmable local controls and the rtl local message which is equivalent to RWLS. See IEEE 488.2, 5.6.3 and 5.6.6.

RULE B.4.10:
When the TCP/IP-IEEE 488.2 Instrument Interface receives a *device_local* RPC, it **SHALL** enable local operation of all programmable local controls which is equivalent to LOCS. See IEEE 488.2, 5.6.4 and 5.6.5.

B.4.6. SRQ Operation

The TCP/IP-IEEE 488.2 Instrument Interface sends the *device_intr_srq* RPC based on the RQS message and whether service requests are enabled with *device_enable_srq*.

RULE B.4.11:
If the interrupt channel is already established and service requests are already enabled via *device_enable_srq* when the RQS message changes from FALSE to TRUE, the TCP/IP-IEEE 488.2 Instrument Interface **SHALL** send *device_intr_srq*.

RULE B.4.12:
If the interrupt channel is already established and the RQS message is already TRUE when service requests are enabled via *device_enable_srq*, the TCP/IP-IEEE 488.2 Instrument Interface **SHALL** send *device_intr_srq*.

RULE B.4.13:
The TCP/IP-IEEE 488.2 Instrument Interface **SHALL NOT** send *device_intr_srq* under any other circumstances.

B.4.7. Read Status Byte Operation

RULE B.4.14:
When the TCP/IP-IEEE 488.2 Instrument Interface receives a *device_readstb* RPC, it **SHALL** return the contents of the Status Byte Register (using RQS), see IEEE 488.2, 11.2, in the *stb* response parameter. It **SHALL** set the rsv message FALSE, see IEEE 488.2 11.3.3, so that the RQS message will be FALSE if a *read_stb* RPC is received again before a new reason for service.

B.4.8. Generic Operations

RULE B.4.15:
The TCP/IP-IEEE 488.2 Instrument Interface **SHALL NOT** perform any action when it receives a *device_docmd* RPC, but **SHALL** terminate with error set to 8, operation not supported,.

B.5. RELATIONSHIP TO IEEE 488.2

A TCP/IP-IEEE 488.2 Instrument Interface and its associated instrument follows all the requirements in IEEE 488.2 except where the difference between the TCP/IP-IEEE 488.2 Instrument Interface and IEEE 488.1 requires clarification.

B.5.1. Remote/Local

RULE B.5.1:
All requirements in IEEE 488.2, 5.6 Remote/Local Requirements **SHALL** be followed.

OBSERVATION B.5.1:
A TCP/IP-IEEE 488.2 Instrument Interface does does not support states equivalent to LWLS and REMS.

B.5.2. Device Clear

RULE B.5.2:
All requirements in IEEE 488.2, 5.8 Device Clear Requirements **SHALL** be followed.

B.5.3. Message Exchange Control Protocol

RULE B.5.3:
All requirements in IEEE 488.2, 6., "Message Exchange Control Protocol", **SHALL** be followed with a redefinition of the **bav**, **brq**, **get**, **dcas**, and **RMT-sent** signals shown in Table B.1.

bav	is set TRUE when *device_write* is received with *data.data_len* greater than zero. is set FALSE by *device_write* immeidately before it sends its reply.
brq	is set TRUE when *device_read* is received with *requestSize* greater than zero. is set FALSE by *device_read* immediately before it sends its reply.
get	is sent when *device_trigger* is received.
dcas	is set TRUE when *device_clear* is received. is set FALSE by *device_clear* immediately before it sends its reply.
RMT-sent	is set TRUE by *device_read* when it has transferred a <RESPONSE MESSAGE TERMINATOR> from the Output Queue. is set FALSE by **bav** or **brq**

Table B.1 Message Exchange Control Messages

OBSERVATION B.5.2:
These signals are not changed unless the link identifier is correct and either this link has the lock or no link has the lock.

OBSERVATION B.5.3:
If *device_write* is called with *data.data_len* set to zero, no byte transfer is attempted, so **bav** is not set TRUE. If *device_read* is called with *requestSize* set to zero, it does not try to remove a byte, so **brq** is not set TRUE.

OBSERVATION B.5.4:
If *device_write* is called with *data.data_len* greater than zero value, **bav** is set TRUE even if no bytes are transferred because the Input Buffer is full. If *device_read* is called with *requestSize* greater than zero, **brq** is set TRUE even if no bytes are transferred because the Output Queue is empty.

OBSERVATION B.5.5:
UNTERMINATED and INTERRUPTED errors may eventually result in *device_read* timing out. In any case, *device_read* returns a reply which has the effect of setting **brq** FALSE.

OBSERVATION B.5.6:
The **get** message is not set TRUE or FALSE in IEEE 488.2. It is a message sent by the I/O control.

B.5.4. Device Listening Formats and Talking Elements

The TCP/IP-IEEE 488.2 Instrument Interface has no EOI line, so a different method is used to transmit the END message.

RULE B.5.4:

All requirements in IEEE 488.2, section 7., "Device Listening Formats" and section 8., "Device Talking Elements", **SHALL** be followed except for the handling of the END message.

1. *device_write* **SHALL** associate an END message with the last byte in *data* when the end flag in *flags* is set.
2. *device_read* **SHALL** send a <RESPONSE MESSAGE TERMINATOR> by putting a newline as the last character in *data* and setting the end flag in *reason*.

B.5.5. Common Commands and Queries

RULE B.5.5:

All requirements in IEEE 488.2, section 10., "Common Commands and Queries" **SHALL** be followed.

RULE B.5.6:

A TCP/IP Instrument Interface which supports *device_trigger* **SHALL** implement *TRG.

PERMISSION B.5.1:

A TCP/IP Instrument Interface which supports *device_trigger* **MAY** implement *DDT and *DDT?.

RULE B.5.7:

The optional commands in the Auto Configure and Controller groups do not apply and **SHALL NOT** be implemented.

B.5.6. Device Status Reporting

IEEE 488.2, 11.3.3 describes how an instrument with an IEEE 488.1 interface reports a service request by asserting the SRQ line.

RULE B.5.8:

All requirements in IEEE 488.2, section 11., "Status Reporting" **SHALL** be followed except that. a TCP/IP IEEE 488.2 Instrument Interface **SHALL** use the *device_intr_srq* RPC to report a service request.

B.5.7. Device/Controller Synchronization Techniques

RULE B.5.9:

All requirements in IEEE 488.2, section 12., "Device/Controller Synchronization Techniques" **SHALL** be followed.