VMEbus Extensions for Instrumentation

TCP/IP-IEEE 488.1 Interface Specification

VXI-11.2

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 488.1 Interface Specification VXI-11.2, 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 488.1 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 488.1 and IEEE-488.2 devices. The methods described allow ASCII-based communications to take place 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 488.1.

A.1. SCOPE

This specification is part of the VXIbus set of specifications and defines a TCP/IP-IEEE 488.1 interface device.

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 488.1 Interface Device.

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 488.1 Interface, defines the operation of a TCP/IP-IEEE 488.1 Interface, including the mapping between the network transactions defined by VXI-11, TCP/IP Instrument Protocol Specification, and IEEE 488.1 operations.

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 488.1 Interface.
2. To define a mapping from network transactions to IEEE 488.1 operations.

A.4. DEFINITION OF TERMS

This specification uses the following terms in addition to those defined in the TCP/IP Instrument Protocol Specification, VXI-11:

TCP/IP-IEEE 488.1 Interface Device: a *network instrument* host which converts *network instrument* messages into IEEE 488.1 operations and routes the request to the appropriate IEEE 488.1 device.

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 488.1-1992, IEEE Standard Digital Interface for Programmable Instrumentation.
[2] 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 IEEE 488.1 operations, along with the operation of a TCP/IP-IEEE 488.1 Interface

Device. 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, VXI-11.1.
[3] VMEbus Extensions for Instrumentation: TCP/IP-IEEE 488.2 Instrument Interface, VXI-11.3.

B. TCP/IP-IEEE 488.1 INTERFACE

The TCP/IP-IEEE 488.1 Interface Device converts *network instrument* messages into IEEE 488.1 operations and routes the request to the appropriate IEEE 488.1 device. It allows controllers to control IEEE 488.1 devices connected to the network via the TCP/IP-IEEE 488.1 Interface.

RULE B.1:

A TCP/IP-IEEE 488.1 Interface Device **SHALL** comply with all the requirements in:

1. IEEE 488.1 and
2. IEEE 488.2, section 15., "IEEE 488.2 Controller Requirements".

PERMISSIONS B.1:

A TCP/IP-IEEE 488.1 Interface Device may provide a mechanism to control the request system control (rsc) local message which can place it in the system control not active state (SNAS). See IEEE 488.1 section 2.12, "Controller (C) Interface Function."

RULE B.2:

A TCP/IP-IEEE 488.1 Interface Device **SHALL** contain one of the Talker subsets T5-T8, IEEE 488.1, 2.5, and one of the Listener subsets L3 or L4, IEEE 488.1, 2.6.

OBSERVATION B.1:

The effect of **RULE B.1** is that a TCP/IP-IEEE 488.1 Interface Device cannot have a secondary address. Other requirements on a TCP/IP-IEEE 488.1 Interface Device do, however, require it to send secondary addresses.

RULE B.3:

The RPC reply **SHALL NOT** be sent until the associated action is complete.

RECOMMENDATION B.1:

A TCP/IP-IEEE 488.1 Interface Device should support two or more *instrument LAN* servers simultaneously.

RULE B.4:

A TCP/IP-IEEE 488.1 Interface Device **SHALL** support at least 14 concurrent links per *network instrument* server.

RECOMMENDATION B.2:

The number of *network instrument* servers and links supported by a TCP/IP-IEEE 488.1 Interface Device should be based on available resources, not on arbitrary predetermined limits.

RULE B.5:

After pon, a TCP/IP-IEEE 488.1 Interface Device **SHALL** set REN true.

Figure B.1 shows a typical TCP/IP-IEEE 488.1 system.

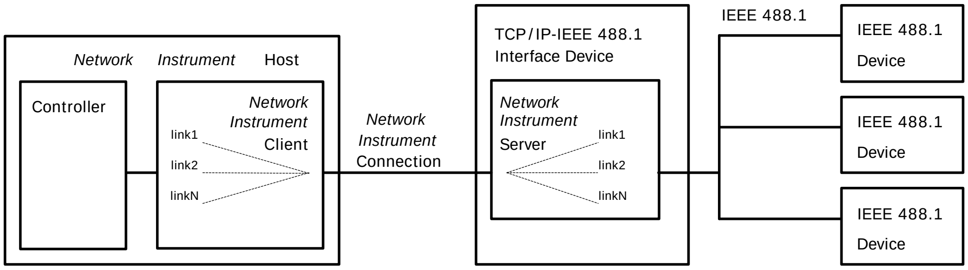

Figure B.1 Typical TCP/IP-IEEE 488.1 System

B.1. DEVICE STRING FORMAT

The routing of messages from the LAN to the appropriate IEEE 488.1 device takes place via the *create_link* RPC. This RPC is used to create a *network instrument* link. The created link is associated with a particular IEEE 488.1 device or interface via the *device* parameter. This parameter is a character string which is parsed by the TCP/IP-IEEE 488.1 Interface Device to determine the IEEE 488.1 device or interface associated with the link.

RULE B.1.1:

A TCP/IP-IEEE 488.1 Interface Device **SHALL** support a *device* string of the following format:

```
<intf_name>[,<primary_addr>[,<secondary_addr>]]
```

where:

<intf_name> A name corresponding to a single IEEE 488.1 interface. This name **SHALL** uniquely identify the interface on the TCP/IP-IEEE 488.1 Interface Device.
<primary_addr> The primary address of a IEEE 488.1 device on the IEEE 488.1 interface (optional).
<secondary_addr> The secondary address of a IEEE 488.1 device on the IEEE 488.1 interface (optional).

RULE B.1.2:

The TCP/IP-IEEE 488.1 Interface Device **SHALL** recognize an <intf_name> of "gpib0" as the first or only IEEE 488.1 interface within the TCP/IP-IEEE 488.1 Interface Device. Additional interfaces **SHALL** be identified by "gpibN" where N is a non-negative integer assigned sequentially starting at one.

RULE B.1.3:

A *device* string which contains just <intf_name> **SHALL** be interpreted by the TCP/IP-IEEE 488.1 Interface Device as a link to the interface.

RULE B.1.4:

A *device* string which contains <intf_name> and an address **SHALL** be interpreted by the TCP/IP-IEEE 488.1 Interface Device as a link to a device at that address on the interface.

Some examples of valid *device* strings are:

gpib0 link is associated with the interface
gpib0,12 link is associated with a device at primary address 12
gpib0,12,5 link is associated with a device at primary address 12 and secondary address 5

OBSERVATION B.1.1:

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

RECOMMENDATION B.1.1:
A controller should use links to the interface with great care. They must be used with device_docmd. They should only be used with device_write, device_read, and device_trigger, when a device needs a special addressing sequence. Some older devices connected to an IEEE 488.1 bus do not properly implement the state machines for the Talker and Listener functions and require a specific order of unlisten, my talk or listen address, and controller's listen or talk address. Where this required order is different from the one specified in this standard, a device_docmd must be used to establish the addressing followed by an RPC that does no addressing. In other situations, links to a device should be used.

B.2. LAN FUNCTIONS

RULE B.2.1:
A TCP/IP-IEEE 488.1 Interface Device **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 488.1 Interface Device **MAY** support other LAN protocols.

RULE B.2.2:
A TCP/IP-IEEE 488.1 Interface Device **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 488.1 Interface Device provides an RJ-45 connector for 10BASE-T. Other connectors may be supported, and may replace the RJ-45 connector.

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

PERMISSION B.2.2:
A TCP/IP-IEEE 488.1 Interface Device **MAY** support other protocols at the Data Link and Physical layers.

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 488.1 Interface Device itself.

B.3.1. Link Creation

The *create_link* and *destroy_link* RPCs do not directly map to any IEEE 488.1 operations.

RULE B.3.1:
The *create_link* and *destroy_link* RPCs **SHALL** be implemented entirely within the TCP/IP-IEEE 488.1 Interface Device. The TCP/IP-IEEE 488.1 Interface Device **SHALL** maintain links as defined by the *network instrument* protocol.

Application	*Instrument LAN*
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 488.1 Interface Device Protocol Stack

B.3.2. Interrupt Channel Creation

The *create_intr_chan* and *destroy_intr_chan* RPCs do not directly map to any IEEE 488.1 operations.

RULE B.3.2:
The *create_intr_chan* and *destroy_intr_chan* RPCs **SHALL** be implemented entirely within the TCP/IP-IEEE 488.1 Interface Device. The TCP/IP-IEEE 488.1 Interface Device **SHALL** maintain the interrupt channel as defined by the *network instrument* protocol.

B.3.3. Device Locking

The *device_lock* and *device_unlock* RPCs do not directly map to any IEEE 488.1 operations.

RULE B.3.3:
The *device_lock* and *device_unlock* RPCs **SHALL** be implemented entirely within the TCP/IP-IEEE 488.1 Interface Device.

B.3.4. Abort Operation

The *device_abort* RPC does not directly map to any IEEE 488.1 operations.

RULE B.3.4:
The *device_abort* RPC **SHALL** be implemented entirely within the TCP/IP-IEEE 488.1 Interface Device to abort any active operations associated with the link.

B.4. INSTRUMENT COMMUNICATIONS

This section covers the operations which communicate directly with IEEE 488.1 devices. Each section typically contains two rules. The first rule specifies how to use a link associated with a device. The second rule covers a link to the interface.

OBSERVATION B.4.1:
In an application uses interface links for the operations described in rules B.4.2, B.4.4, B.4.6, or B.4.8, it will not be portable to interface types other than the IEEE 488.1 interface supported by TCP/IP-IEEE 488.1 Interface Devices

B.4.1. Data Transfer from Interface Device to IEEE 488.1

RULE B.4.1:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_write* RPC and the link identifier is associated with a device, it **SHALL** transfer the bytes in the *data* parameter using the SEND control sequence described in IEEE 488.2, 16.2.4, where the supplied listen address is the one associated with the link and the terminator is null if the end flag in *flags* is false or send the last DAB with END if the end flag in *flags* is true.

RULE B.4.2:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_write* RPC and the link identifier is associated with an interface, it **SHALL** transfer the bytes in the *data* parameter using the SEND DATA BYTES control sequence described in IEEE 488.2, 16.2.3, where the terminator is null if the end flag in *flags* is false or send the last DAB with END if the end flag in *flags* is true.

B.4.2. Data Transfer from IEEE 488.1 to Interface Device

RULE B.4.3:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_read* RPC and the link identifier is associated with a device, it **SHALL** transfer bytes into the *data* parameter using the RECEIVE control sequence described in IEEE 488.2, 16.2.7, where the supplied talk address is the one associated with the link and the stop-handshaking condition is the same as the termination condition.

RULE B.4.4:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_read* RPC and the link identifier is associated with an interface, it **SHALL** transfer bytes into the *data* parameter using the RECEIVE RESPONSE MESSAGE control sequence described in IEEE 488.2, 16.2.6, where the stop-handshaking condition is the same as the termination condition.

B.4.3. Device Clear Operation

RULE B.4.5:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_clear* RPC and the link identifier is associated with a device, it **SHALL** execute the DEVICE CLEAR control sequence for a selected device described in IEEE 488.2, 16.2.9.1, where the supplied address is the one associated with the link.

RULE B.4.6:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_clear* RPC and the link identifier is associated with an interface, it **SHALL** execute the DEVICE CLEAR control sequence for all devices described in IEEE 488.2, 16.2.9.2.

B.4.4. Trigger Operation

RULE B.4.7:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_trigger* RPC and the link identifier is associated with a device, it **SHALL** execute the TRIGGER control sequence for selected devices described in IEEE 488.2, 16.2.19.1, where the supplied address is the one associated with the link.

RULE B.4.8:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_trigger* RPC and the link identifier is associated with an interface, it **SHALL** execute the TRIGGER control sequence for all addressed devices described in IEEE 488.2, 16.2.19.2.

B.4.5. Remote/Local Operation

RULE B.4.9:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_remote* RPC and the link identifier is associated with a device, it **SHALL** execute the SET RWLS control sequence described in IEEE 488.2, 16.2.12, where the supplied listen address is the one associated with the link.

RULE B.4.10:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_remote* RPC and the link identifier is associated with an interface, it **SHALL** return with *error* set to 8, operation not supported.

RULE B.4.11:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_local* RPC and the link identifier is associated with a device, it **SHALL** execute the ENABLE LOCAL CONTROLS control sequence for selected devices described in IEEE 488.2, 16.2.10.1, where the supplied listen address is the one associated with the link.

RULE B.4.12:
When a TCP/IP-IEEE 488.1 Interface Device receives a *device_local* RPC and the link identifier is associated with an interface, it **SHALL** return with *error* set to 8, operation not supported.

B.4.6. SRQ Operation

The TCP/IP-IEEE 488.1 Interface Device sends the *device_intr_srq* RPC based on the state of the SRQ line and whether service requests are enabled. The SRQ line is the logical OR of all the RQS messages in an IEEE 488.1 system. Whenever any device on the interface requests service, a *device_intr_srq* will be sent for every link that has enabled service requests .

RULE B.4.13:
For each link, if the interrrupt channel is already established and service requests are already enabled by *device_enable_srq* when the SRQ line changes from FALSE to TRUE , the TCP/IP-IEEE 488.1 Interface Device **SHALL** send *device_intr_srq*.

RULE B.4.14:

For each link, if the interrrupt channel is already established and the SRQ line is already TRUE when service requests are enabled by *device_enable_srq* when they were previously disabled, the TCP/IP-IEEE 488.1 Interface Device **SHALL** send *device_intr_srq*.

RULE B.4.15:

The TCP/IP-IEEE 488.1 Interface Device **SHALL NOT** send *device_intr_srq* under any other circumstances.

RECOMMENDATION B.4.1:

A *network instrument* client should create the interrupt channel before enabling service requests and maintain the interrupt channel while service requests are enabled. Otherwise, service requests might be lost.

B.4.7. Read Status Byte Operation

RULE B.4.16:

When a TCP/IP-IEEE 488.1 Interface Device receives a *device_readstb* RPC and the link identifier is associated with a device, it **SHALL** execute the READ STATUS BYTE control sequence described in IEEE 488.2, 16.2.18, where the supplied talk address is the one associated with the link. The stored status byte and RQS message **SHALL** be returned in *stb*.

RULE B.4.17:

When a TCP/IP-IEEE 488.1 Interface Device receives a *device_readstb* RPC and the link identifier is associated with an interface, it **SHALL** return with *error* set to 8, operation not supported.

B.5. INTERFACE COMMUNICATION

The *device_docmd* RPC is a general purpose RPC which provides interface specific operations not covered by the other defined RPCs. Table B.1 describes the allowed commands.

RULE B.5.1:

A TCP/IP-IEEE 488.1 Interface Device **SHALL** support the *cmd* values listed in Table B.1 and no others. If a TCP/IP-IEEE 488.1 Interface Device receives a *device_docmd* RPC with *cmd* set to a value other than one listed in Table B.1, the *device_docmd* RPC **SHALL** return immediately with *error* set to 8, operation not supported.

All values of *cmd* in the range 020000_{16} through $02FFFF_{16}$ are reserved for use by TCP/IP-IEEE 488.1 Interface Devices.

Table B.1 Allowed Generic Commands

Name	*cmd*	*data_in.* *data_in_len*	*datasize*
Send Command	020000_{16}	0-128	1
Bus Status	020001_{16}	2	2
ATN Control	020002_{16}	2	2
REN Control	020003_{16}	2	2
Pass Control	020004_{16}	4	4
Bus Address	$02000A_{16}$	4	4
IFC Control	020010_{16}	0	X

RULE B.5.2:

When a TCP/IP-IEEE 488.1 Interface Device receives a *device_docmd* RPC and the link identifier is associated with a device, it **SHALL** perform no action and return with *error* set to 8, operation not supported.

RULE B.5.3:

If the values received in *data_in.data_in_len* or *datasize* conflict with those shown in Table B.1 for a *cmd*, the TCP/IP-IEEE 488.1 Interface Device **SHALL** perform no action and return with *error* set to 5, parameter error. For an entry of X in the *datasize* column, a TCP/IP-IEEE 488.1 Interface Device **SHALL** accept any value.

RULE B.5.4:

The values in *data_in* and *data_out* are unsigned integers of *datasize* bytes.

B.5.1 Send Command

RULE B.5.5:

In response to a *device_docmd* RPC whose *cmd* value is 020000_{16}, a TCP/IP-IEEE 488.1 Interface Device **SHALL** execute the SEND COMMAND control sequence described in IEEE 488.2, 16.2.1, where the commands sent are contained in *data_in*. The returned *data_out* **SHALL** be same as the received *data_in*.

B.5.2 Bus Status

RULE B.5.6:

In response to a *device_docmd* RPC whose *cmd* value is 020001_{16}, a TCP/IP-IEEE 488.1 Interface Device **SHALL** return a value in the response parameter *data_out* which depends on the received value in *data_in* as described in table B.2. The size of the returned *data_out, data_out.data_out_len,* **SHALL** be two.

Table B.2 Received and Returned Values for Bus Status

Name	***data_in***	**value returned in *data_out***
REMOTE	1	1 if the REN message is true, 0 otherwise
SRQ	2	1 if the SRQ message is true, 0 otherwise
NDAC	3	1 if the NDAC message is true, 0 otherwise
SYSTEM CONTROLLER	4	1 if the TCP/IP-IEEE 488.1 Interface Device is in the system control active state, 0 otherwise
CONTROLLER-IN-CHARGE	5	1 if the TCP/IP-IEEE 488.1 Interface Device is not in the controller idle state, 0 otherwise
TALKER	6	1 if the TCP/IP-IEEE 488.1 Interface Device is addressed to talk, 0 otherwise
LISTENER	7	1 if the TCP/IP-IEEE 488.1 Interface Device is addressed to listen, 0 otherwise
BUS ADDRESS	8	the TCP/IP-IEEE 488.1 Interface Device's address (0-30)

B.5.3 ATN Control

RULE B.5.7:

In response to a *device_docmd* RPC whose *cmd* value is 020002_{16}, a TCP/IP-IEEE 488.1 Interface Device **SHALL**:

1. If the *data_in* parameter is non-zero, then set the ATN line true.
2. If the *data_in* parameter is zero, then set the ATN line false.

The returned *data_out* **SHALL** be same as the received *data_in*.

B.5.4 REN Control

RULE B.5.8:

In response to a *device_docmd* RPC whose *cmd* value is 020003_{16}, a TCP/IP-IEEE 488.1 Interface Device **SHALL**:

1. If the *data_in* parameter is non-zero, then set the sre (send remote enable) message true.
2. If the *data_in* parameter is zero, then set the sre (send remote enable) message false.

The returned *data_out* **SHALL** be same as the received *data_in*.

B.5.5 Pass Control

RULE B.5.9:

In response to a *device_docmd* RPC whose *cmd* value is 020004_{16}, a TCP/IP-IEEE 488.1 Interface Device **SHALL** execute the PASS CONTROL control sequence described in IEEE 488.2, 16.2.14 where the talk address is constructed from the value in *data_in* bitwise ORed with 80_{16}. The returned *data_out* **SHALL** be same as the received *data_in*.

B.5.6 Bus Address

RULE B.5.10:

In response to a *device_docmd* RPC whose *cmd* value is $02000A_{16}$ and *data_in* contains a value between 0 and 30 inclusive, a TCP/IP-IEEE 488.1 Interface Device **SHALL** set its address to the contents of *data_in*. If *data_in* does not contain a legal value, *device_docmd* **SHALL** return immediately with *error* set to 5, parrameter error. The returned *data_out* **SHALL** be same as the received *data_in*.

B.5.7 IFC Control

RULE B.5.11:

In response to a *device_docmd* RPC whose *cmd* value is 020010_{16}, a TCP/IP-IEEE 488.1 Interface Device **SHALL** execute the SEND IFC control sequence described in IEEE 488.2, 16.2.8. The returned *data_out* **SHALL** have *data_out.data_out_len* set to zero.