IEEE Standard Serial Interface for Programmable Instrumentation

Sponsored by the

Automated Instrumentation Committee
of the
IEEE Instrumentation and Measurement Society

Approved 30 January 2000

IEEE-SA Standards Board

Abstract: A serial, full-duplex, asynchronous, 9-pin data terminal equipment (DTE) communications port for programmable instrumentation that follows ANSI/TIA/EIA-574-1990 and related standards is defined in this standard.
Keywords: controller, data communication equipment (DCE), data terminal equipment (DTE), device, IEEE 488.1, IEEE 488.2, interchange circuit, null modem cable, observation, permission, recommendation, rule

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2001 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 14 February 2001. Printed in the United States of America.

Print: ISBN 0-7381-2706-X SH94897
PDF: ISBN 0-7381-2707-8 SS94897

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

IEEE is the sole entity that may authorize the use of certification marks, trademarks, or other designations to indicate compliance with the materials set forth herein.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

Introduction

(This introduction is not part of IEEE Std 1174-2000, IEEE Standard Serial Interface for Programmable Instrumentation.)

The development of this standard began with an ambitious plan to layer IEEE Std 488.1-1987 and IEEE Std 488.2-1992 so that several media could take advantage of the installed base of IEEE 488 protocols. This initiative was taken in cooperation with the IEC/SC65C Working Group 3 (WG3). It appeared that this was not possible without a considerable change of the format of the IEEE 488 standards. This was considered as undesirable. Therefore, the scope was much reduced to just the support of what is popularly known as the "RS-232 serial interface" which is ubiquitous on personal computers. This standard draws from existing standards as much as possible but seeks to provide a standardization for an instrument design that chooses the slower speed serial communication because of the simplicity of the interface, the availability of most any personal computer as a controller, or other factors.

At the time of the publication of this standard, the members of the Working Group were:

Robert S. Cram, *Chair*

Stephen Greer	Joe Mueller	Carsten Schilling
Steve Lomas	John Pieper	Jochen Wolle

In the development of this standard, the Working Group was supported by many other individuals from many different organizations who contributed time, effort, and technical expertise. In particular, the Working Group wishes to acknowledge the following individuals for their contributions:

Chris Bagge	Gerrit de Vries	Knud Johansen
Dieter Burkhart	Carl Hagerling	Allen Tracht

The following members of the balloting committee voted on this standard:

David Anderson	Michael G. Harrison	Eric Sacher
Elaine Cooney	Ben Lee	Larry C. Sollman
Roy Courtney	Steve Lomas	Joseph J. Stanco
Robert S. Cram	Doug Lyons	Robin Steele
Stephen Greer	Marlyn Miner	Andrew Thomson
Carl Hagerling	Robert C. Rassa	Allen Tracht

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

When the IEEE-SA Standards Board approved this standard on 30 January 2000, it had the following membership:

Richard J. Holleman, *Chair*
Donald N. Heirman, *Vice Chair*
Judith Gorman, *Secretary*

Satish K. Aggarwal
Dennis Bodson
Mark D. Bowman
James T. Carlo
Gary R. Engmann
Harold E. Epstein
Jay Forster*
Ruben D. Garzon
James H. Gurney
Lowell G. Johnson
Robert J. Kennelly
E. G. "Al" Kiener
Joseph L. Koepfinger*
L. Bruce McClung
Daleep C. Mohla
Robert F. Munzner
Louis-François Pau
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Akio Tojo
Hans E. Weinrich
Donald W. Zipse

*Member Emeritus

Also included is the following nonvoting IEEE-SA Standards Board liaison:

Robert E. Hebner

Andrew D. Ickowicz
IEEE Standards Project Editor

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

Contents

1. Overview 1

1.1 Scope 1

2. References 2

3. Definitions, acronyms and abbreviations, and terminology 3

3.1 Definitions 3
3.2 Acronyms and abbreviations 4
3.3 Terminology 4

4. Mechanical, electrical, and circuit requirements 5

4.1 Introduction 5
4.2 IEEE 1174 Performance Class 1 compliance 5
4.3 Mechanical 5
4.4 Electrical 9
4.5 Interchange circuits 10
4.6 Signaling 12

5. IEEE Std 488.1-1987 emulation 14

5.1 Introduction 14
5.2 IEEE 1174 Performance Class 2 compliance 14
5.3 IEEE Std 488.1-1987 function emulation 14

6. IEEE Std 488.2-1992 emulation 19

6.1 Introduction 19
6.2 IEEE 1174 Performance Class 3 compliance 19
6.3 IEEE 488.2 function emulation 20

Annex A (informative) Message exchange control operation 26

Annex B (informative) IEEE Performance classification 30

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

IEEE Standard Serial Interface for Programmable Instrumentation

1. Overview

Clause 4 defines electrical, mechanical, and interchange circuit requirements suitable for controlling programmable instruments over a serial interface. Clause 5 defines techniques to emulate IEEE Std 488.1-1987[1] which are suitable for controlling programmable instruments over a serial interface as defined in Clause 4. Clause 6 defines techniques to emulate IEEE Std 488.2-1992 which are suitable for controlling programmable instruments over a serial interface as defined in Clause 4 and Clause 5. In conjunction, Clause 4 through Clause 6 form a set describing a method for ASCII-based communication over a serial link between a controller and a device using concepts found in IEEE Std 488.2-1992. Figure 1 shows the relationship among the standards. Annex A describes a state machine implementation which follows the rules of Clause 6, and Annex B summarizes the allowed Performance Classes to an IEEE 1174 interface.

1.1 Scope

This standard defines a serial, full-duplex, asynchronous, 9-pin data terminal equipment (DTE) communications port that follows ANSI/TIA/EIA-574-1990 and related standards. Portions of several standards will be referenced along with specific extensions applying to programmable instruments.

This standard applies point-to-point instrument systems composed of a measurement, stimulus, or interconnect device and an instrumentation controller. It may also apply to certain devices outside the scope of the instrument system environment.

This standard covers mechanical, electrical, and functional requirements of the interface. Intended readers include electrical, mechanical, and software engineers.

Topics covered in IEEE Std 488.2-1992 include the following:

a) Extraction of items in IEEE Std 488.2-1992, which are independent of IEEE Std 488.1-1987.
b) Alternatives for IEEE Std 488.2-1992 functionality that require IEEE Std 488.1-1987.

[1]Information on references can be found in Clause 2.

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

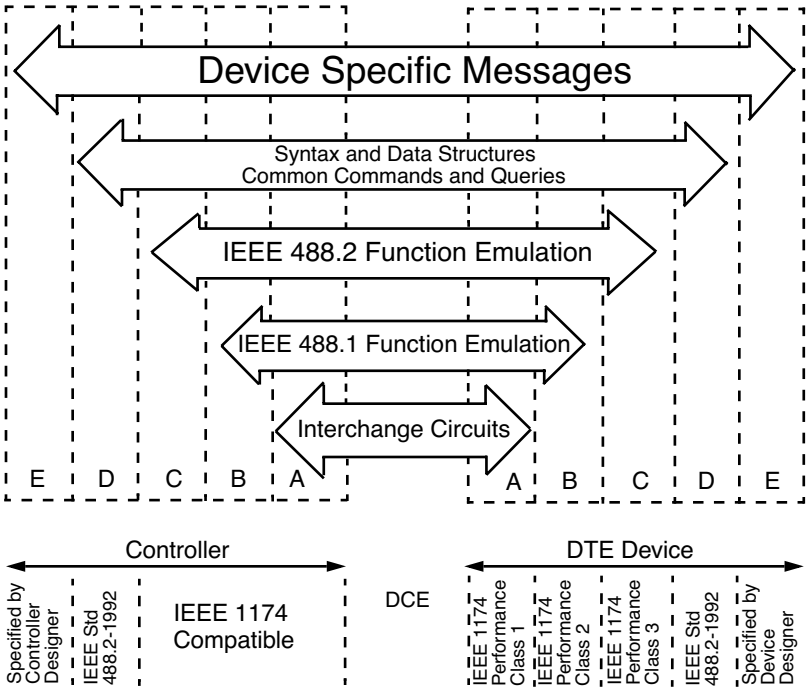

Figure 1—Relationship between IEEE Std 1174-2000 and IEEE 488 standards

2. References

This standard shall be used in conjunction with the following publications. When the following standards are superseded by an approved revision, the revision shall apply.

ANSI X3.4-1986 (R1997), American National Standard Code for Information Interchange (7-bit ASCII)—Information Systems—Coded Character Sets—7-bit.[2]

ANSI/TIA/EIA-232-F-1997, Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange.[3]

ANSI/TIA/EIA-404-B-1995, Standard for Start-Stop Signal Quality for Non-Synchronous Data Terminal Equipment.[4]

ANSI/TIA/EIA-562-1990 (R1998), Electrical Characteristics for an Unbalanced Digital Interface.

ANSI/TIA/EIA-574-1990 (R1998), 9-Position Non-Synchronous Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange.

IEEE Std 488.1-1987 (Reaff. 1994), IEEE Standard Digital Interface for Programmable Instrumentation[5]

[2]ANSI publications are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[3]EIA publications are available from Global Engineering Documents, 15 Inverness Way East, Englewood, CO 80112, USA (http://global.ihs.com/).

[4]TIA publications are available from Global Engineering Documents, 15 Inverness Way East, Englewood, CO 80112, USA (http://global.ihs.com/).

[5]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://standards.ieee.org/).

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

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

ISO 1177:1985, Information Processing—Character Structure for Start/Stop and Synchronous Character Oriented Transmission[6]

ITU Recommendation V.14 (03/93)—Transmission of Start-Stop Characters Over Synchronous Bearer Channels[7]

ITU Recommendation V.24 (02/00)—List of Definitions for Interchange Circuits Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE)

ITU Recommendation V.28 (03/93)—Electrical Characteristics for Unbalanced Double-Current Interchange Circuits

ITU Recommendation V.32 (03/93)—A Family of 2-Wire, Duplex Modems Operating at Data Signaling Rates of up to 9600 bit/s for Use on the General Switched Telephone Network and on Leased Telephone-type Circuits

ITU Recommendation V.42 (10/96)—Error-Correcting Procedures for DCE's Using Asynchronous-To-Synchronous Conversion

3. Definitions, acronyms, abbreviations, and terminology

3.1 Definitions

3.1.1 controller: Data terminal equipment (DTE) that sends program message to and receives response messages from the device.

3.1.2 data communication equipment (DCE): In a data station, the equipment that provides the signal conversion and coding between the DTE and the line.

3.1.3 data terminal equipment (DTE): The part of a data station that serves as a data source, a data sink, or both[7]. In general, DTE may include terminals, computers, instruments, and other devices involved with the end-point generation and reception of interchange circuits. Specifically, in this document, DTE refers to instruments.

3.1.4 device: A DTE that receives a program message from and sends response messages to the controller.

3.1.5 interchange circuit: An interconnecting circuit at the interface between DTE and DCE for the transfer of binary data, control, and timing signals.[8]

3.1.6 null modem cable: A cable that allows two DTE devices to be directly connected by simulating, through cross wiring in the cable, an intermediate DCE-line-DCE circuit.

---

[6]ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[7]ITU publications are available from the International Telecommunications Union, Sales Section, Place des Nations, CH-1211, Genève 20, Switzerland/Suisse (http://www.itu.int/). They are also available in the United States from the U.S. Department of Commerce, Technology Administration, National Technical Information Service (NTIS), Springfield, VA 22161, USA.

[8]Refer to ITU Recommendation V.24 (02/00) and ITU Recommendation V.28 (03/93).

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

3.2 Acronyms and abbreviations

ANSI	American National Standards Institute
CCITT	International Telegraph and Telephone Consultative Committee
CTS	Clear To Send
DCD	Data Carrier Detected
DCE	Data Communication Equipment
DSR	Data Set Ready
DTE	Data Terminal Equipment
DTR	Data Terminal Ready
EIA	Electronic Industries Association
IEEE	Institute of Electrical and Electronics Engineers
ITU	International Telecommunications Union (formerly CCITT)
RFR	Ready For Receiving data
RI	Ring Indicator
RLSD	Receive Line Signal Detected
RXD	Received Data
TIA	Telecommunications Industries Association
TXD	Transmitted Data
XOFF	Transmit Off
XON	Transmit On

3.3 Terminology

Throughout this standard, the following headings identify the effects of the paragraphs on an implementation.

3.3.1 Rule

Rules shall be followed to ensure compatibility. A rule is characterized by the use of the words *shall* and *shall not*. These words are not used for any other purpose than stating rules.

3.3.2 Recommendation

Recommendations consist of advice to implementers which will affect the usability of the final device. These words are not used for any other purpose than stating rules.

3.3.3 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.

3.3.4 Observation

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

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

4. Mechanical, electrical, and circuit requirements

4.1 Introduction

This clause defines electrical, mechanical, and interchange circuit requirements suitable for controlling programmable instruments over a serial interface. It describes a serial, full-duplex, asynchronous, 9-pin DTE communication port that follows ANSI/TIA/EIA-574-1990 and related standards. Portions of several other standards are also referenced along with specific extensions as they apply to instrument control.

This clause is one part in a set that together describe a method for ASCII-based communication over a serial link between a controller and an instrument. The other clauses in the set, Clause 5 and Clause 6, deal with emulating selected functionality of IEEE Std 488.1-1987 and IEEE Std 488.2-1992. Figure 1 shows the relationship among the standards.

This clause applies to point-to-point instrument systems composed of a measurement, stimulus, or interconnect device and an instrumentation controller. The connection may be hard-wired, by a modem interconnection, electrical to fiber optic transceivers, or other applicable means. The performance defined by this clause may also apply to certain devices outside the scope of the instrument system environment. The clause covers mechanical, electrical, and functional requirements of the interface. Intended readers include electrical, mechanical, and software engineers. This standard does not cover the interaction of multiple interfaces in a given device.

The objective of this clause is to provide a definition for the electrical, mechanical, and signalling requirements of a serial control interface suitable for use with programmable instruments. The interface defined by this standard is a serial, full-duplex, asynchronous, 9-pin DTE communication port that follows ANSI/TIA/EIA-574-1990 and related standards. Portions of several standards are referenced here along with specific extensions applying to programmable instruments.

4.2 IEEE 1174 Performance Class 1 compliance

RULE: A device claiming compliance to IEEE 1174 Performance Class 1 (Base Functionality) shall follow all rules as mentioned in Clause 4.

RULE: An IEEE 1174-compliant device shall follow all requirements of IEEE 1174 Performance Class 1.

4.3 Mechanical

Mechanical requirements follow ANSI/TIA/EIA-574-1990. This clause details the references for each mechanical requirement area and adds further rules and recommendations as needed.

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS Not for Resale

4.3.1 Connector

4.3.1.1 Dimensions

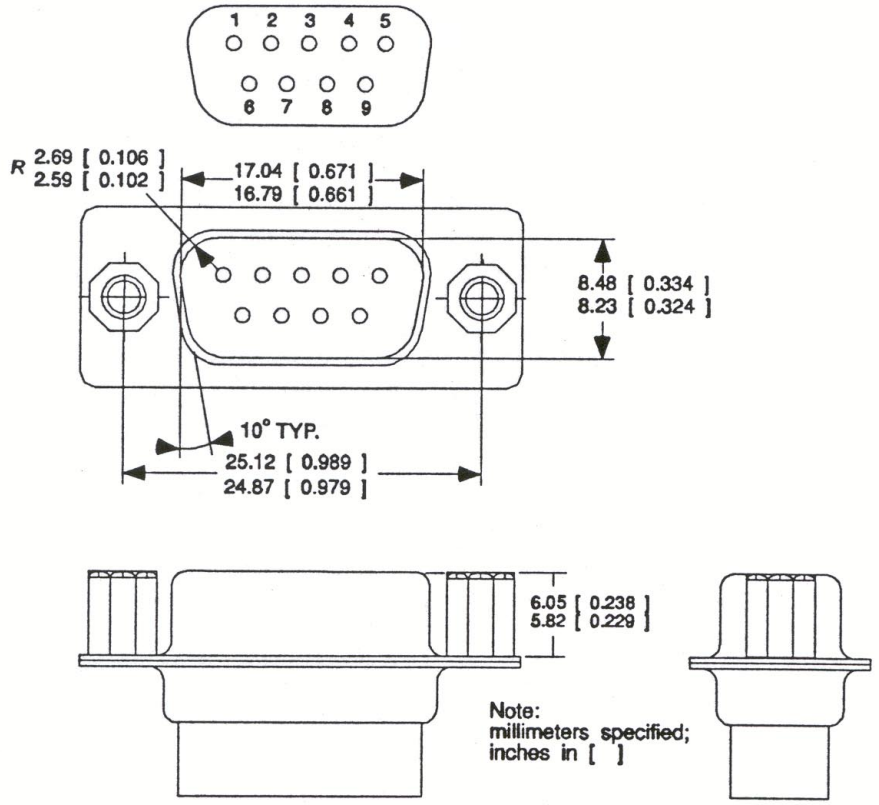

Figure 2—DTE interface connector

RULE: The interface shall follow the recommendations and requirements of clause 2.2 of ANSI/TIA/EIA-574-1990 as it applies to a 9 position (9-pin) DTE connector.

OBSERVATION: This connector uses male (pin) connectors and a female shell (plug connector).

OBSERVATION: The 9-pin connector was chosen for its size as panel space on instruments is often limited.

4.3.1.2 Labeling

RULE: The interface connector shall be labeled IEEE 1174.

PERMISSION: Additional labeling covering related or supporting standards may be added.

4.3.1.3 Contact number assignments

RULE: The contact number assignments for the 9-pin male connector shall follow Figure 2.7 of ANSI/TIA/EIA-574-1990.

RULE: Contact number 7 shall implement the functionality of circuit 133.

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

PERMISSION: Connection to contact numbers 6 (circuit 107) and 9 (circuit 125) may be omitted.

RULE: If a device uses contact number 6 (circuit 107) or 9 (circuit 125), the usage shall be documented.

OBSERVATION: In most devices, contact numbers 6 and 9 are not implemented as they do not have a useful function.

The pin assignments from ANSI/TIA/EIA-574-1990 with the above rules and permissions applied is summarized in Table 1. Common names and abbreviations for the various circuits are also given.

4.3.2 Cable

4.3.2.1 Mechanical

RULE: The cable interface connector shall follow the requirements of clause 2.2 of ANSI/TIA/EIA-574-1990.

4.3.2.2 Shielding

RECOMMENDATION: The cable terminator should be enclosed in an electrically conductive housing that is specifically designed to minimize electromagnetic radiation from the cable assembly.

RECOMMENDATION: Cables should be shielded with the shield being electrically connected to the connector housing of both terminators.

OBSERVATION: The mutual capacitance of shielded cables may be greater than that of unshielded cables; therefore shielded cables may have to be shorter for reliable transmission.

4.3.2.3 Null modem cable

A null modem cable is often used to directly connect DTE. A cable wired as illustrated in Figure 3 should allow communication from a controller to a device.

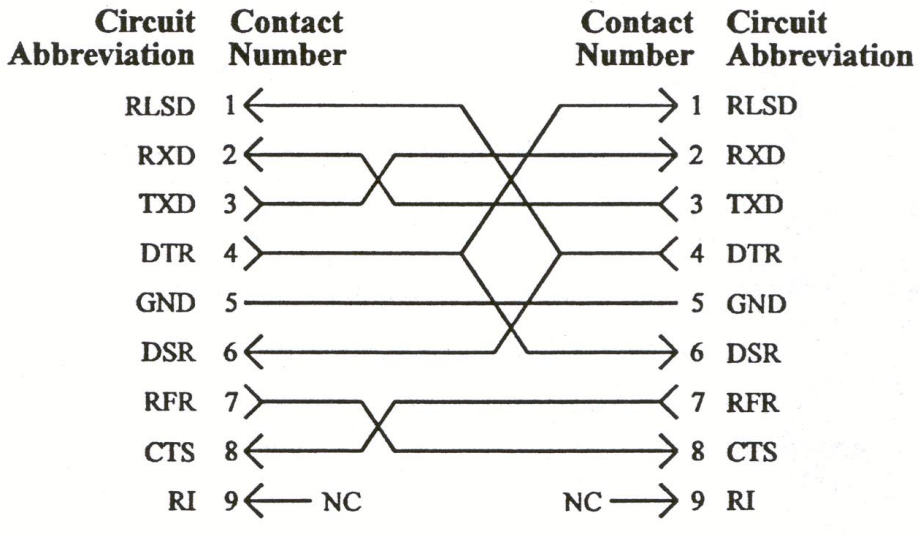

Figure 3—Null modem cable wiring

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS

Table 1—Device 9-pin receptacle contact number assignments

Direction	Circuit ITU (EIA)	Circuit abbreviation	Circuit name and description[a]	HW Handshake Level	SW Handshake Level	Modem	Contact number
To DTE	109 (CF)	RLSD[b]	*Received line signal detector:* This circuit indicates the presence or lack of a carrier.	NONE	NONE	V.32 V.42	1
To DTE	104 (BB)	RXD	*Received data:* The data signals generated by the DCE are transferred on this circuit to the DTE.	FULL	FULL	V.32 V.42	2
From DTE	103 (BA)	TXD	*Transmitted data:* The data signals originated by the DTE are transferred on this circuit to the DTE.	FULL	FULL	V.32 V.42	3
From DTE	108/2 (CD)	DTR	*Data terminal ready:* This circuit indicates the readiness of the DTE to accept or produce signals on all other circuits.	HIGH	NONE	V.32	4
—	102 (AB)	GND	*Signal ground or common return:* This conductor establishes the signal common return and the DC reference potential for all interchange circuits.	FULL	FULL	V.32 V.42	5
To DTE	107 (CC)	DSR	*Data set ready:* This circuit indicates the readiness of the DCE to accept and process signals on all other circuits.	NONE	NONE	V.32	6 (See note)
From DTE	133 (CJ)	RFR[c]	*Ready for receiving:* This circuit indicates whether the DTE is ready to accept data on circuit 104 (RXD).	FULL	NONE	V.32	7
To DTE	106 (CB)	CTS	*Ready for sending:* This circuit indicates to the DTE whether the DCE is prepared to accept data signals on circuit 103 (TXD).	FULL	NONE	V.32 V.42	8
To DTE	125 (CE)	RI	*Ring indicator:* This circuit indicates the presence of a ring signal on the General Switched Telephone Network attached to the DCE.	NONE	NONE	—	9 (See note)

NOTE—indicates that the connection may be omitted. See 4.3.1.3.

[a]A more complete description of each interchange circuit can be found in ITU Recommendation V.24 (02/00).
[b]Historically called DCD.
[c]Historically, the RTS circuit was assigned to this contact.

Not for Resale
Copyright © 2001 IEEE. All rights reserved.

4.4 Electrical

4.4.1 Interface

RULE: The interface shall follow the ANSI/TIA/EIA-562-1990 electrical requirement specifications.

OBSERVATION: Following ANSI/TIA/EIA-562-1990 assures compatibility with other ANSI/TIA/EIA-562-1990 interfaces as well as ITU Recommendation V.28 (03/93) and ANSI/TIA/EIA-232-F-1997 electrical interfaces.

OBSERVATION: ANSI/TIA/EIA-562-1990 allows baud rates to 64,000 baud.

4.4.2 Cable

RECOMMENDATION: The interface connecting cable should follow the guidelines given in Annex A of ANSI/TIA/EIA-562-1990.

OBSERVATION: The maximum acceptable length of the interface cable is not defined but is determined by the electrical specifications given in ANSI/TIA/EIA-562-1990. For typical maximum cable lengths at certain bit rates and cable characteristics, see Figure 4.

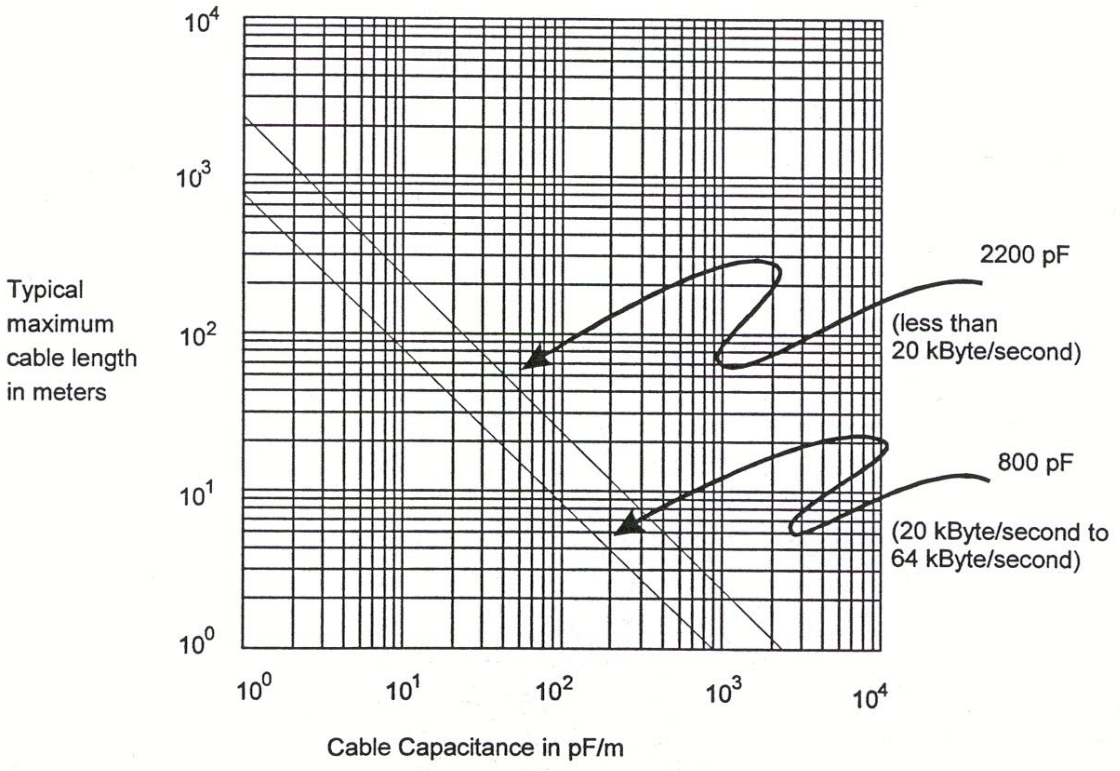

Figure 4—Typical maximum cable lengths

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

4.5 Interchange circuits

4.5.1 Circuit definitions

A device uses a set of interconnecting circuits, called interchange circuits, to communicate with a DCE or another DTE through a null modem cable. The level of support of interchange circuits depends upon which flow control method(s) is implemented. See 4.5.3 for more information.

RULE: Devices shall function as DTE.

RULE: Devices that support hardware flow control (RFR/CTS) shall support the interchange circuits, shown in Table 1, whose entries in the column titled "HW Handshake Level" is FULL. The definition of these interchange circuits shall be as defined by Clause 3 of ITU Recommendation V.24 (02/00).

RULE: Devices that support software flow control (XON/XOFF) shall support the interchange circuits shown in Table 1 whose entries in the column titled "SW Handshake Level" is FULL. The definition of these interchange circuits shall be as defined by Clause 3 of ITU Recommendation V.24 (02/00).

RULE: For both flow control options, the interchange circuits marked as NONE in the appropriate "Level" column of Table 1 shall either be unconnected or fully implemented as per ITU Recommendation V.24 (02/00).

RULE: For both flow control options, the interchange circuits marked as HIGH in the appropriate "Level" column of Table 1 shall either be driven to a mark (1) or fully implemented as per Clause 3 of ITU Recommendation V.24 (02/00).

OBSERVATION: The column in Table 1 titled "Modem" shows other ITU modem standards which expect the existence of the associated interchange circuit as described in ITU Recommendation V.24 (02/00).

4.5.2 Circuit operation

RULE: The interchange circuit operation shall follow Clause 4 of ITU Recommendation V.24 (02/00), along with the additional requirements in 4.5.3.1 of this standard, for circuits 103 (TXD), 104 (RXD), 106 (CTS), and 133 (RFR).

RULE: The interface on a device (DTE) shall be designed such that circuit 109 (RLSD) maintains an OFF condition when nothing is attached to the interface connector. The open circuit voltage on the interface connector shall not exceed 2 V. These are requirements of ITU Recommendation V.28 (03/93) and ANSI/TIA/EIA-232-F-1997.

RULE: The interface on a device (DTE) shall be designed such that circuit 107 (DSR), if implemented, maintains an OFF condition when nothing is attached to the interface connector. The open circuit voltage on the interface connector shall not exceed 2 V. These are requirements of ITU Recommendation V.28 (03/93) and ANSI/TIA/EIA-232-F-1997.

RULE: The interface on a device (DTE) shall be designed such that circuit 125 (RI), if implemented, maintains an OFF condition when nothing is attached to the interface connector. The open circuit voltage on the interface connector shall not exceed 2 V. These are requirements of ITU Recommendation V.28 (03/93) and ANSI/TIA/EIA-232-F-1997.

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

4.5.3 Flow control

Flow control is used to control the transfer of data on circuits 103 (TXD) and 104 (RXD) and is recommended in order to prevent data overrun errors. When the DCE indicates to the DTE that it is unable to accept data, The DTE ceases transmission. When the DCE indicates it is able to accept data, the DTE may resume transmission. The DTE provides similar indications to the DCE to control transfer in the opposite direction.

RECOMMENDATION: A device should implement either soft (XON/XOFF), or hard (RFR/CTS), or both flow control methods.

RULE: Device documentation shall state the flow control method(s) implemented.

OBSERVATION: Zero or more methods of flow control can be active at any time.

PERMISSION: Selection of hard, soft, or neither flow control methods may be achieved locally or remotely.

OBSERVATION: Commercially available integrated circuits used for generating the serial data stream may have an transmit buffer of up to three characters that cannot be removed. The protocol for flow control must take this into account and ensure that the temporary interruption of the data transfer can be performed without generating error conditions in the data stream. This transmit buffer results in a need for a buffer in the receiving end after the generation of a ‘DTE not-ready’ or ‘DCE not-ready’ condition.

RECOMMENDATION: Device documentation should include information about the presence and size of transmit and receive buffers.

4.5.3.1 RFR/CTS flow control

RULE: RFR/CTS flow control shall use two interchange circuits for the ‘DTE not-ready’ and ‘DCE not-ready’ conditions. Circuit 106 (CTS) shall be used to indicate ‘DCE not-ready’. Circuit 133 (RFR) shall be used to indicate ‘DTE not-ready’.

RULE: A device (DTE) that interrupts a transmission of Received Data on circuit 104 (RXD) by turning circuit 133 (RFR) OFF shall accept the character that was being received on circuit 104 (RXD) when the ON-to-OFF transition occurred. After accepting this character, the DTE shall accept at least six more characters without generating an overflow condition.

RULE: A DTE that has its transmission interrupted in the middle of transmitting characters on circuit 103 (TXD) by an ON-to-OFF transition on circuit 106 (CTS) shall continue the transmission of the character until the STOP unit element is sent. After the STOP unit element is sent, the DTE shall transmit no more than four additional characters. Circuit 103 (TXD) shall then be held in the mark (1) condition, as long as circuit 106 (CTS) remains in the OFF condition.

RECOMMENDATION: After sensing an ON-to-OFF transition on circuit 106 (CTS) a DTE should send as few additional characters as possible to avoid an overflow in the DCE.

OBSERVATION: An attached DCE that interrupts a transmission by turning circuit 106 (CTS) OFF is expected to accept the character that was in the process of being transmitted on circuit 103 (TXD) when the ON-to-OFF transition occurred. After accepting this character, the DCE should accept at least six more characters without generating an overflow condition.

RULE: During idle periods [intervals when circuit 106 (CTS) is ON and no data are available for transmission), the DTE shall maintain a mark (1) condition on circuit 103 (TXD) (i.e., no transmission shall occur).

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS Not for Resale

RULE: In devices that implement and support hard wired RFR/CTS flow control, the operation of circuits 106 (CTS) and 133 (RFR) shall be according to clause 7.3.1 of ITU Recommendation V.42 (10/96).

RULE: The interface on a device (DTE) shall be designed such that circuit 106 (CTS) maintains an OFF condition when nothing is attached to the interface connector. The open circuit voltage on the interface connector shall not exceed 2 V. These are requirements of ITU Recommendation V.28 (03/93) and ANSI/TIA/EIA-232-F-1997.

4.5.3.2 XON/XOFF flow control

RULE: XON/XOFF flow control shall use two characters from the ANSI X3.4-1986 ASCII 7-bit code to set and clear the 'DTE not-ready' and 'DCE not-ready' conditions. DC3 (XOFF) shall be used to indicate 'DCE not-ready' and 'DTE not-ready'. DC1 (XON) shall be used to clear the 'DCE not-ready' and 'DTE not-ready' conditions.

RULE: A device (DTE) that interrupts a transmission by sending XOFF shall accept the character that was being received on circuit 104 (RXD). After accepting this character, the DTE shall accept at least 60 more characters without generating an overflow condition.

RULE: A DTE that has its transmission interrupted in the middle of transmitting characters on circuit 103 (TXD) by receipt of an XOFF character shall continue the transmission of the character until the STOP unit element is sent. After the STOP unit element is sent, the DTE shall transmit no more than 30 additional characters. Circuit 103 (TXD) shall then be held in the mark (1) condition until an XON character is received.

RECOMMENDATION: After receiving an XOFF character, a DTE should send as few additional characters as possible to avoid an overflow in the DCE.

RULE: While transmission is suspended due to receipt of an XOFF character, subsequent XOFF characters shall be ignored until an XON character is received.

RULE: While transmission is enabled due to receipt of an XON character, subsequent XON characters shall be ignored until an XOFF character is received.

OBSERVATION: The use of XON/XOFF flow control precludes the reliable exchange of binary data. While XON/XOFF flow control is active, this standard does not provide any method of sending XON or XOFF characters without having them affect flow control. Applications which require the exchange of arbitrary binary data should use RFR/CTS flow control.

4.6 Signaling

4.6.1 Data bytes

RULE: Data is transferred on circuits 103 (TXD) and 104 (RXD).

RULE: Data shall consist of characters sent using a start/stop data transmission system. Each character shall contain exactly 8 bits preceded by a start bit, space (0), and followed by a stop bit, mark (1), to create a frame.

RULE: Bit sequencing shall follow Clause 3 of ISO 1177:1985.

RULE: A device shall accept received characters with a stop element of either one- or two-unit intervals duration (one or two stop bits). Acceptance of characters with a stop element of two unit intervals duration (two stop bits) does not require action on the part of the receiver (i.e., no framing or overflow protocol will result).

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

OBSERVATION: Parity bits are not used.

PERMISSION: Error detection and/or correction may be handled by other layers or entities. For example, error correcting modems or electrical-to-fiber-optic transceivers may be used in electrically noisy environments to ensure communication integrity.

RULE: Signal quality shall be as specified in ANSI/TIA/EIA-404-B-1995. The transmitting circuit in the DTE on circuit 103 (TXD) shall fulfill Performance Category II for a Transmitting Data Terminal and the receiving circuit in the DTE on circuit 104 (RXD) shall fulfill Performance Category C for a Receiving Data Terminal as specified in ANSI/TIA/EIA-404-B-1995.[9]

4.6.2 Break signal

RULE: Devices shall be able to receive and recognize a break signal as defined in Clause 7.3 of ITU Recommendation V.14 (03/93).

OBSERVATION: ITU Recommendation V.14 (03/93) defines the break duration to be two character sequences plus three stop bits or greater. A break's duration is about 25 ms at 1200 baud.

RULE: A device shall not generate a break signal.

OBSERVATION: There is no protocol specified in this standard defining how a controller sends a break signal. There is also no protocol which defines how a device interprets a break.

4.6.3 Bit rate

RULE: Each bit of a frame shall be signified by a mark (1) or space (0) on the circuit for a fixed period of time, referred to as the clocking rate. The circuit is held in a mark (1) between frames.

PERMISSION: Starting spaces need not fall on an integer multiple of the base clocking rate.

RULE: Bit rates of 1200, 2400, 4800, and 9600 bits/second shall be supported.

RECOMMENDATION: Additional bit rates of 19 200 and 38 400 are recommended and any additional bit rates should be documented.

PERMISSION: Other bit rates are allowed.

RULE: A device shall have a local means that is not a hard local control (see 5.6.1.5 of IEEE Std 488.2-1992) to select any of the bit rates implemented by the device.

RULE: The device shall provide the means to remotely program the device to all bit rates supported by the device.

OBSERVATION: The above rule allows a controller to change the bit rate of the media over which communication is occurring.

RULE: The transmit and receive bit rates shall be the same.

RULE: The bit rate shall be retained during power-off.

OBSERVATION: The above rule requires non-volatile memory in the device.

[9]It is anticipated that the above mentioned categories of ANSI/TIA/EIA-404-B-1995 will be accepted as the ITU worst-case figures.

--`,,`-`-`,,`,,`,`,,`---
Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

5. IEEE Std 488.1-1987 emulation

5.1 Introduction

This clause defines techniques to emulate IEEE Std 488.1-1987 suitable for controlling programmable instruments over a serial interface as defined in Clause 4. That clause describes a serial, full-duplex, asynchronous 9-pin DTE communication port that follows ANSI/TIA/EIA-574-1990 and related standards.

This clause, in conjunction with Clause 4, form a set describing a method for ASCII-based communication over a serial link between a controller and an instrument. Figure 1 shows the relationship among the standards.

This clause applies to point-to-point instrument systems composed of a measurement, stimulus, or interconnect device and an instrumentation controller. It is intended to apply to interfaces that comply to the IEEE 1174 Performance Class 1.

The objective of this clause is to provide a definition for emulating IEEE Std 488.1-1987 functionality, identified as Performance Class 2, that resides on the IEEE 1174 Performance Class 1 interface.

5.2 IEEE 1174 Performance Class 2 compliance

RULE: A device claiming compliance to the IEEE 1174 Performance Class 2 (IEEE Std 488.1-1987 Emulation) shall conform to the IEEE 1174 Performance Class 1 and shall follow all rules as mentioned in Clause 5 of this standard.

OBSERVATION: Compliance to the IEEE 1174 Capability Class 2 is optional, but IEEE 1174 devices must follow all requirements of IEEE 1174 Capability Class 1.

RULE: The device shall recognize and record the following protocol errors:

a) *Framing Error:* Received data did not detect the stop bit (for example, due to baud rate mismatch).
b) *Input buffer overrun:* A software or hardware input buffer overflowed due to improper or nonexistent flow control.

RULE: Characters received with a framing error shall be discarded.

OBSERVATION: Further handling of protocol errors is outside the scope of this standard.

RULE: The device shall provide a method for reporting these errors to the controller.

5.3 IEEE Std 488.1-1987 function emulation

This clause provides for the emulation of certain messages and protocols found in IEEE Std 488.1-1987. As a serial interface has no “out-of-band” signal path, these codes are emulated using the normal receive and transmit data bytes but employing an encoding that allows the signals to be trapped from the normal data path. This is illustrated in Figure 5.

--`,,`-`-`,,`,,`,`,,`---
Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

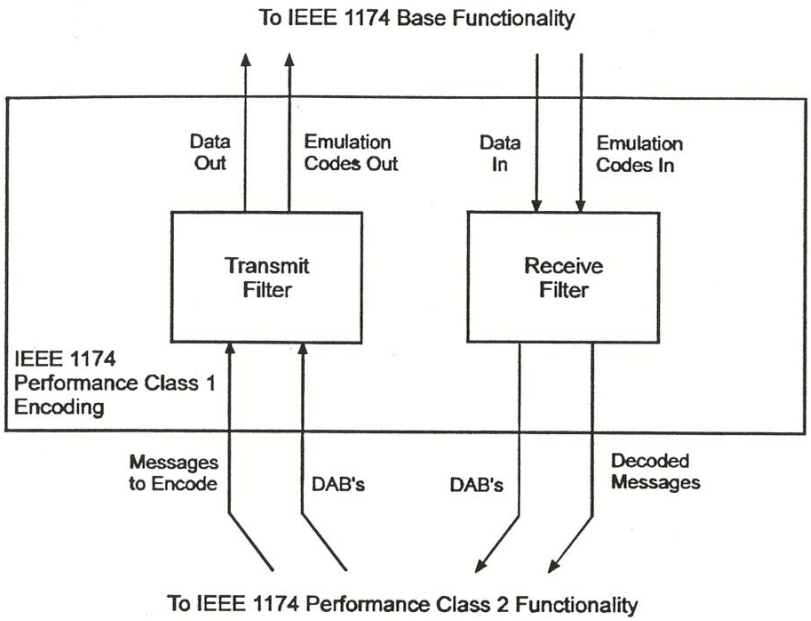

Figure 5—IEEE 1174 encoding

RULE: A device shall either implement all the IEEE Std 488.1-1987 emulation codes, in the cases shown, as described in Table 2, or none of the emulation codes. The choice shall be stated in user documentation.

Table 2—IEEE Std 488.1-1987 function emulation

Code	Originator	Function
&SRQ CRLF[a]	device	requests service (asynchronous)
&ddd CRLF[a]	device	sends STB and RQS as three decimal digits (response to &POL)
&DCL CRLF[a]	device	acknowledges device clear completion (emulated by a break)
&GET	controller	emulates device trigger
&GTL	controller	changes device R/L state to local
&LLO	controller	changes device R/L state to lockout
&POL	controller	performs serial poll (device responds with &ddd CRLF)
&SFC	controller	enables software (XON/XOFF) flow control
&HFC	controller	enables hard wired (RFR/CTS) flow control
&DFC	controller	disables flow control

[a]CRLF indicates that a carriage return, followed by a line feed, is appended to the emulation code.

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

OBSERVATION: Many system applications can operate effectively without ever sending or interpreting emulation codes found in IEEE Std 488.1-1987. Emulation codes are useful to minimize changes to existing application software designed to operate with protocols of IEEE Std 488.1-1987.

OBSERVATION: This specification only supports selected functions found in IEEE Std 488.1-1987.

RULE: The controller generated emulation code messages shall be trapped by a receive filter in the device interface and shall be interpreted case insensitive as out-of-band messages with behavior as described in this clause. The device shall document the operating mode of the receive filter.

The receive filter shall operate in one of the following two modes:

a) *IEEE 488.1 Mode:* Emulation codes shall be recognized and trapped anywhere in the input stream. The character sequence && shall be interpreted as a single & on both input and output.
b) *IEEE 488.2 Mode:* Emulation codes shall be recognized and trapped anywhere in the input stream outside of an IEEE 488.2 <ARBITRARY BLOCK PROGRAM DATA> or <STRING PROGRAM DATA> syntactic element. They shall be considered normal data inside these elements.

5.3.1 Device Clear function

The IEEE 488.1 Device Clear is emulated by the break signal. A device receiving the break behaves as if it were handshaking a DCL message.

RULE: A device shall emulate the IEEE 488.1 Device Clear (DC) function by entering the equivalent of the IEEE 488.1 Device Clear Active State (DCAS) on receipt of a break signal. In this state, the device shall send an internal message to the device function(s) causing it (them) to be cleared.

RULE: To indicate completion of the clearing process and the readiness to accept data, the device shall always return a response composed of the four characters, &DCL, followed by a CRLF terminator, followed by a delay of five character times.

OBSERVATION: This is the acknowledgment protocol for the IEEE 488.1 DC function and corresponds to an IEEE 488.1 DC function returning from DCAS to Device Clear Idle State (DCIS). As there is no 'DCL holdoff' possible, the application needs to wait for the response to insure subsequent bytes sent are not inadvertently cleared.

RECOMMENDATION: The device documentation should state the maximum time required for performing a device clear action.

OBSERVATION: ITU Recommendation V.42 (10/96) compliant modems, using the destructive/expedited break operating mode, may approximate the Device Clear action of a direct connection obtained with a null modem cable.

5.3.2 Device Trigger function

The IEEE 488.1 Device Trigger function is emulated in an IEEE 1174 system by the receipt of the Group Execute Trigger (GET) emulation code. A device receiving this code behaves as if it were handshaking an IEEE 488.1 GET message.

PERMISSION: A device may support a GET.

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

RULE: If the device supports a GET, it shall emulate the IEEE 488.1 Device Trigger function by entering the equivalent of the IEEE 488.1 Device Trigger Active State (DTAS) on receipt of the &GET emulation code. In this state, the device shall respond by sending an internal message causing the device to start its basic operation. After sending the internal message, the device automatically returns to the equivalent of the IEEE 488.1 Device Trigger Idle State (DTIS).

RULE: A device which does not support a group execute trigger shall recognize an &GET emulation code, but perform no action.

OBSERVATION: There is no acknowledgment of the received trigger signal. The assumption is that the trigger will be executed in the order it is received such as is specified in IEEE Std 488.2-1992.

5.3.3 Remote/Local function

An IEEE 1174 device of Performance Class 2 uses the two remote/local emulation code messages to emulate the IEEE 488.1 Remote Local (RL) Interface Function.

RULE: A device shall emulate the IEEE 488.1 Remote/Local function by making the transition to the appropriate equivalent IEEE 488.1 Remote/Local state on receipt of the &GTL and &LLO emulation codes, the return to local (rtl) and power on (pon) local messages, and the receipt of a data byte and RLSD[10] line false as shown in Figure 6. The enabling and disabling action of local controls shall be the same as defined for these four states in subclause 2.8.3 of IEEE Std 488.1-1987.

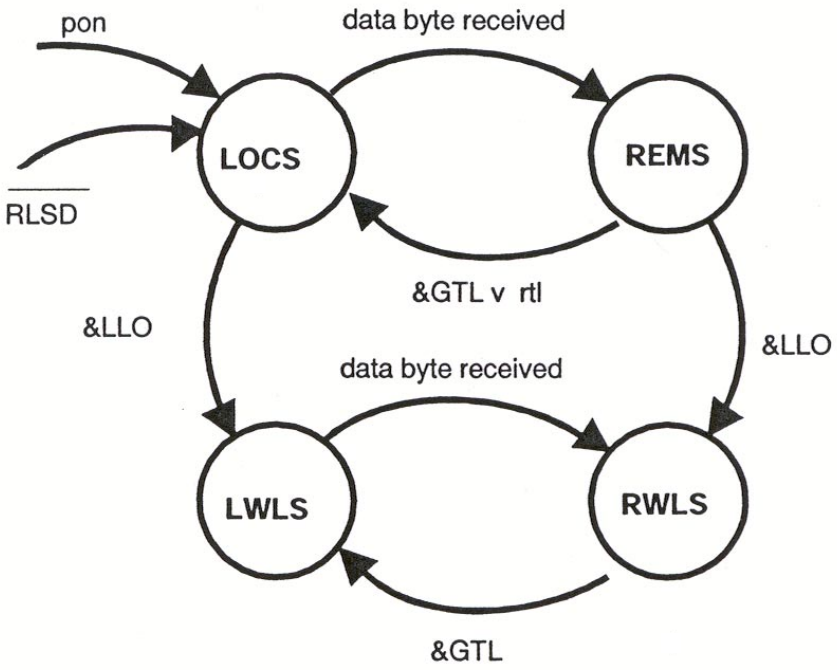

Figure 6—Remote/Local state emulation diagram

[10]Received Line Signal Detector; see Table 1.

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

5.3.4 Service Request function

An IEEE 1174 device of Performance Class 2 emulates the requesting of service by sending the service request emulation code as a separate response message.

RULE: The equivalent of the IEEE 488.1 Service Request state diagram, found in subclause 2.7.2 of IEEE Std 488.1-1987, shall be followed. The device shall send an equivalent of the IEEE 488.1 local request service (rsv) message to request service. Upon entering Service Request State (SQRS), the device shall emulate the transmission of the IEEE 488.1 SRQ message by sending the &SRQ CRLF emulation code once.

RULE: The sending of the &SRQ CRLF emulation code shall wait for the completion of any in-progress response message.

A controller emulates a serial poll by sending the &POL emulation code and receiving the poll response as &ddd.

RULE: The device shall respond to an &POL emulation code by returning the STB and RQS messages in an emulation code message of the form &ddd CRLF, where ddd represents the three ASCII-encoded decimal digits which form the decimal representation of the eight-bit quantity normally returned by an IEEE 488.1 serial poll. See subclause 6.5.2 of IEEE Std 488.1 -1987.

RULE: The equivalent of the IEEE 488.1 Service Request state diagram transitions shall be followed where the receipt of an &POL emulation code shall create the equivalent of the Serial Poll Active State (SPAS) true condition, and the sending of the &ddd CRLF response shall create a SPAS false condition. If the Service Request state is Affirmative Poll Response State (APRS), the RQS message shall be sent true.

OBSERVATION: SPAS is herein defined only for its interaction with the SR diagram for the purpose of performing a serial poll as an atomic operation and sending the proper RQS message. There is no emulation of SPE, SPD, etc.

5.3.5 Flow control

RULE: A device which implements XON/XOFF flow control shall respond to an &SFC emulation code by turning on its XON/XOFF flow control method.

RULE: A device which does not implement XON/XOFF flow control shall recognize an &SFC emulation code, but perform no action.

RULE: A device which implements RFR/CTS flow control shall respond to an &HFC emulation code by turning on its RFR/CTS flow control method.

RULE: A device which does not implement RFR/CTS flow control shall recognize an &HFC emulation code, but perform no action.

RULE: A device which implements either flow control method shall respond to an &DFC emulation code by turning off all active flow control methods. Before disabling flow control, the device shall wait until it is ready to receive data.

OBSERVATION: Both flow controls may be active simultaneously.

RULE: A device which does not implement any of the flow control methods shall recognize an &DFC emulation code, but perform no action.

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

6. IEEE Std 488.2-1992 emulation

6.1 Introduction

This clause defines techniques to emulate IEEE Std 488.2-1992 suitable for controlling programmable instruments over a serial interface as defined in Clause 4 and Clause 5. These two clauses describe a serial, full-duplex, asynchronous, 9-pin DTE communication port that follows ANSI/TIA/EIA-574-1990 and related standards and emulates selected IEEE Std 488.1-1987 functionality.

This clause, in conjunction with Clause 4 and Clause 5, forms a set describing a method for ASCIl-based communication over a serial link between a controller and a device using IEEE 488.2 concepts. Figure 1 shows the relationship among the standards.

This clause applies to point-to-point instrument systems composed of a measurement, stimulus, or interconnect device and an instrumentation controller. It is intended to apply to interfaces that comply to the IEEE 1174 Performance Classes 1 and 2.

The objective of this clause is to provide a definition for emulating IEEE Std 488.2-1992 functionality, identified as Performance Class 3, that resides on the IEEE 1174 Performance Class 1 and IEEE 1174 Performance Class 2 interfaces.

6.2 IEEE 1174 Performance Class 3 compliance

RULE: A device claiming compliance to IEEE 1174 Performance Class 3 (Full Capability) shall conform to the IEEE 1174 Capability Classes 1 and 2 and shall follow all rules as mentioned in Clause 6 of this standard.

OBSERVATION: Compliance to IEEE 1174 Performance Class 3 is optional, but IEEE 1174 devices must follow all requirements of IEEE 1174 Capability Class 1.

PERMISSION: This standard does not prohibit full-duplex operation. That is, devices may be able to simultaneously transmit and receive eight data-bit characters over the interface. Full-duplex operation may provide enhanced performance.

RULE: The device report shall framing errors and input buffer overrun by setting the DDE bit in the Standard Event Status Register.

RULE: When one of these errors occurs, the parser shall discard subsequent characters until an LF is encountered, and then accept new characters as the beginning of a new message. The Input Buffer and Output Queue shall not be flushed.

RULE: When a <COMMAND MESSAGE UNIT> is received which changes the bit rate, the change shall take place when the <PROGRAM MESSAGE TERMINATOR> is parsed.

RULE: All IEEE 488.1 emulation codes defined in IEEE 1174 Performance Class 2 shall be implemented. The emulation code receive filter shall be implemented in the IEEE 488.2 mode (see 5.3).

RULE: The &GET emulation codes defined in IEEE 1174 Performance Class 2 shall be implemented. The emulation code receive filter shall be implemented in the IEEE 488.2 mode (see 5.3).

OBSERVATION: The emulation generates selected interface messages and state transitions normally provided by the IEEE 488.1 functions. The intent is that the relevant portions of IEEE 488.2 that interface directly with IEEE 488.1 will now interact in an analogous way with the emulated codes and protocols of IEEE 1174 Performance Class 2.

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

6.3 IEEE 488.2 function emulation

In general, IEEE 1174 Performance Class 2 devices use the same rules, recommendations, and permissions as IEEE Std 488.2-1992. The differences between an IEEE 488.1 and a serial interface, however, require some clarification of the intent of IEEE Std 488.2-1992 in certain areas.

RULE: Conformance to IEEE 1174 Performance Class 3 requires all rules, recommendations, and permissions of IEEE Std 488.2-1992 to be followed except when explicitly specified in this clause.

6.3.1 Listening and talking formats

RULE: Devices shall use the syntax described in Clause 7 and Clause 8 of IEEE Std 488.2-1992, with the modifications listed in this clause, to receive <PROGRAM MESSAGE> elements and send <RESPONSE MESSAGE> elements, respectively.

OBSERVATION: This specification does not support the END message.

6.3.1.1 Listening

RULE: The <PROGRAM MESSAGE TERMINATOR> shall be defined as optional white space followed by the line feed character, LF.

OBSERVATION: White space follows the IEEE Std 488.2-1992 definition and includes all control characters, including CR, except LF. Therefore, CRLF is an acceptable terminator.

OBSERVATION: Received IEEE 488.1 emulation codes are recognized anywhere outside of <ARBITRARY BLOCK PROGRAM DATA> and <STRING PROGRAM DATA> elements. See 5.3 for a description of emulation codes and the IEEE 488.2 receive filter mode.

OBSERVATION: The #0 indefinite length option for <ARBITRARY BLOCK PROGRAM DATA> is not expected to be received. Devices will accept data bytes until an LF character is received or until a **dcas** message is received.

6.3.1.2 Talking

RULE: The <RESPONSE MESSAGE TERMINATOR> shall be the carriage return character, followed by the line feed character, CRLF.

OBSERVATION: Some popular application programs need to see input from a serial port terminated with CRLF.

RULE: <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> shall not be sent.

RULE: <ARBITRARY ASCII RESPONSE DATA> shall be sent with CRLF as the terminator.

6.3.2 Common commands and queries

RULE: A device shall implement all the common commands and queries required in Clause 10 of IEEE Std 488.2-1992.

PERMISSION: A device may implement any optional common command and queries mentioned in Clause 10 of IEEE Std 488.2-1992, except those in the Auto Configure and Controller groups.

--`,,`-`-`,,`,,`,`,,`---
Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

6.3.3 Status reporting

RULE: A device shall follow all the requirements in Clause 11 of IEEE Std 488.2-1992, where references to IEEE Std 488.1-1987 are to be replaced by the IEEE 1174 Performance Class 2 Service Request function emulation.

OBSERVATION: For example, IEEE 1174 Performance Class 2 requires a device to send &SRQ CRLF rather than assert the SRQ line when the requirements of subclause 11.3 of IEEE Std 488.2-1992 would have the device change RQS from FALSE to TRUE.

OBSERVATION: The effect of the &POL emulation code on Status Reporting is the same as if the device had been serial polled; that is, the Set rsv State Diagram in Figure 11-5 of IEEE Std 488.2-1992 makes the transition from SRWS to SRAS to SRIS.

6.3.4 Device/Controller synchronization

RULE: A device shall follow all the requirements in Clause 12 of IEEE Std 488.2-1992.

6.3.5 Message exchange protocol

The Message Exchange Control (MEC) is the state machine portion of the device's MEC Interface (see Figure 7). The MEC observes the exchange of <PROGRAM MESSAGE> and <RESPONSE MESSAGE> elements between the controller and the device. When the MEC recognizes that the controller has violated the expected protocol, the MEC performs actions which resynchronize the protocol. The MEC also detects if the MEC Interface is deadlocked.

The controller and device normally exchange complete <PROGRAM MESSAGE> and <RESPONSE MESSAGE> elements. Message exchanges are initiated by the controller. The device does not send anything to the controller until the Response Formatter places bytes into the Output Queue as a result of a valid query. The device may send emulation codes from the transmit filter.

One or more <QUERY MESSAGE UNIT> elements can be placed in a single <PROGRAM MESSAGE>. Also, the device is allowed to generate one or more <RESPONSE MESSAGE UNIT> elements for each <QUERY MESSAGE UNIT> received. The <RESPONSE MESSAGE UNIT SEPARATOR> (semicolon, ";") is used to separate <RESPONSE MESSAGE UNIT> elements within the complete compound <RESPONSE MESSAGE>. The <RESPONSE MESSAGE TERMINATOR> terminates the last <RESPONSE MESSAGE UNIT>. Since the required terminator for a <RESPONSE MESSAGE UNIT> is dependent upon whether a subsequent <RESPONSE MESSAGE UNIT> is in the same <RESPONSE MESSAGE>, the device is not able to terminate each <RESPONSE MESSAGE UNIT> until the Parser receives either another <QUERY MESSAGE UNIT> or <PROGRAM MESSAGE TERMINATOR>. If any of the <QUERY MESSAGE UNIT> elements of a compound query return more than one <RESPONSE MESSAGE UNIT>, a controller application program cannot use the semicolon delimiter (;) to distinguish the end of the response to each <QUERY MESSAGE UNIT>.

Compound queries do not span more than one complete <PROGRAM MESSAGE>. If a compound query exceeds a device's ability to hold the associated <RESPONSE MESSAGE>, the device's ability to format, execute, and parse may become blocked while waiting until room is available.

RULE: The MEC Interface shall generate exactly one <RESPONSE MESSAGE> for each <PROGRAM MESSAGE> it receives that generates a response. Until the MEC Interface receives a <PROGRAM MESSAGE> that generates a <RESPONSE MESSAGE>, the MEC Interface shall not generate any <RESPONSE MESSAGE> elements.

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS Not for Resale

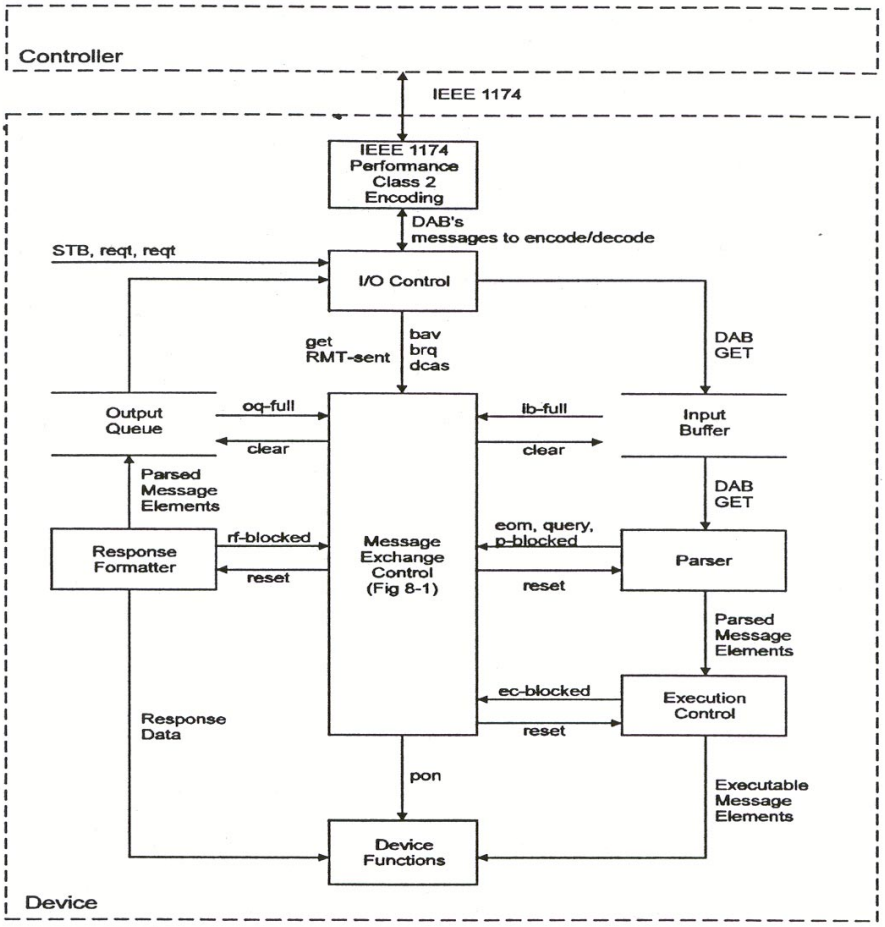

Figure 7—Message exchange control interface functional blocks

For example, a device is asked to take five separate readings and return them in a single <RESPONSE MESSAGE>:

```
Command:    MEAS_FIVE? <PMT>
Response:   6;2;9;2;4 <RMT>
```

As another example, a device is asked to return all its setting information about its OHMS function:

```
Command:    OHMS-SET? <PMT>
Response:   RANGE 100; AVGS 10; ... ; AUTOZERO 1 <RMT>
```

RULE: Information about any <QUERY MESSAGE UNIT> which returns more than one <RESPONSE MESSAGE UNIT> shall be supplied to the user.

The controller normally does not attempt to read any part of a <RESPONSE MESSAGE> from the device until it has sent the entire <PROGRAM MESSAGE> to the device. The controller normally does not send any part of another <PROGRAM MESSAGE> to the device until it has read the entire <RESPONSE MESSAGE> from the device.

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

At any given time, the device is doing one of the following things:

a) The device is idle, waiting for a message from the controller.
b) The device is receiving a message from the controller. The controller will normally not attempt to read data bytes from the device until it has sent a <PROGRAM MESSAGE TERMINATOR> to the device.
c) The device is sending a <RESPONSE MESSAGE> to the controller. The controller will normally not attempt to send data bytes to the device until it has received a <RESPONSE MESSAGE TERMINATOR> from the device.

RULE: The I/O Control shall generate the signals:

bav	when a data byte is received.
brq	when a data byte is sent.
get	when an emulated GET is received (optional). See 5.3.2.
dcas	when an emulated DCL is received. See 5.3.1.
RMT-sent	when a <RESPONSE MESSAGE TERMINATOR> is sent.

RULE: An MEC shall be non-intrusive when the application program follows the message exchange protocol. It shall also detect when the application program deviates from the message exchange protocol and then perform certain actions designed to re-establish message exchange synchronization as soon as possible after the device detects a violation.

6.3.5.1 Normal behavior

During normal MEC operation, the controller sends complete <PROGRAM MESSAGE> elements. Whenever the controller sends a <PROGRAM MESSAGE> or **get** which generates a response, the controller reads the entire <RESPONSE MESSAGE> element. The controller does not ask for a response which overflows the response message buffering in the MEC Interface. The controller never asks for a response which contains an indefInite response element followed by more response data.

RULE: The MEC shall not perform any actions while:

a) The controller only sends <PROGRAM MESSAGE> elements which do not generate any response.
b) The controller sends a <PROGRAM MESSAGE> element which generates a <RESPONSE MESSAGE> element, and the controller reads the entire <RESPONSE MESSAGE> element.

6.3.5.2 INITIALIZE action

Whenever the Message Exchange Interface is first powered on, or the controller sends a break, the Message Exchange Protocol shall INITIALIZE.

RULE: When the MEC receives a **pon** signal from the Device Functions or a **dcas** signal from the I/O control, it shall

a) Clear the Input Buffer
b) Clear the Output Queue
c) Reset the Parser
d) Reset the Execution Control
e) Reset the Response Formatter.

--`,,`-`-`,,`,,`,`,,`---
Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS Not for Resale

6.3.5.3 INTERRUPTED action

Whenever the controller sends a new message before reading the entire response, the Message Exchange Protocol has been INTERRUPTED.

RULE: The MEC shall detect an INTERRUPTED violation of the message exchange protocol when the device receives a <PROGRAM MESSAGE> which generates a <RESPONSE MESSAGE> element and then receives more data bytes from the controller before the entire response has been sent. After detecting an INTERRUPTED violation, any remaining portion of the <RESPONSE MESSAGE> element not yet sent shall be discarded. A Query Error shall be reported to Status Reporting. The incoming program message which interrupted the response shall be processed normally, including the generation of any appropriate response.

OBSERVATION: If the controller application program sends two <PROGRAM MESSAGE> elements where both generate a response without reading the response from the first, and then attempts to read the response to the second, it may receive some data bytes from the first response followed by the complete second response. The result may or may not be a syntactically correct <RESPONSE MESSAGE>. System timing may allow the controller to send the second <PROGRAM MESSAGE> and begin its read operation before the Parser detects the end of the first <PROGRAM MESSAGE>. In this case, the MEC Interface starts to send the first response to the controller. When the terminator of the first <PROGRAM MESSAGE> is parsed, the INTERRUPTED violation is finally detected, the remainder of the first response is discarded and the Parser processes the second query.

6.3.5.4 UNTERMINATED action

Whenever the controller asks for a response without the controller sending a complete message which generates a response, the Message Exchange Protocol is UNTERMINATED.

OBSERVATION: With a full-duplex serial interface allowed, however, the device cannot determine when the controller requests the response, so it cannot detect an UNTERMINATED violation.

6.3.5.5 DEADLOCK action

By sending several <QUERY MESSAGE UNIT> elements in a single <PROGRAM MESSAGE> element, a controller may ask the device to buffer more data than is possible. In this case, the Message Exchange Protocol is DEADLOCKed.

RULE: The MEC shall detect a DEADLOCK violation when

a) The controller sends more data than the Message Exchange Interface can buffer, so that more data cannot be received, and
b) The response generated by the portion of the <PROGRAM MESSAGE> element processed so far is so large that it cannot be buffered.

After detecting a DEADLOCK violation, all of the <RESPONSE MESSAGE> element shall be discarded. A Query Error shall be reported to Status Reporting. The Parser shall continue to parse and execute the <PROGRAM MESSAGE> until the end of the <PROGRAM MESSAGE> (e.g., a <PROGRAM MESSAGE TERMINATOR>) is identified. The next <PROGRAM MESSAGE> shall be processed normally, including generating any appropriate response.

OBSERVATION: When the controller attempts to read the <RESPONSE MESSAGE> after a DEADLOCK, the device does not send any data bytes. Not sending any data bytes may cause the controller to time out. When this happens, an *ESR? (Standard Event Status Register Query) can be used to determine that no data was sent because a Query Error had occurred.

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

OBSERVATION: A full-duplex serial interface allows the controller to read the <RESPONSE MESSAGE> from the device while still sending the <PROGRAM MESSAGE>. Doing so should avoid DEADLOCK.

OBSERVATION: Flow control may be used to avoid DEADLOCK.

6.3.5.6 Query after indefinite response action

Certain compound queries may attempt to generate an illegal <RESPONSE MESSAGE>.

RULE: If a <QUERY MESSAGE UNIT> which generates an <ARBITRARY ASCII RESPONSE DATA> element is not the last <QUERY MESSAGE UNIT> in a <PROGRAM MESSAGE>, the device shall report a Query Error and not send the response from the following <QUERY MESSAGE UNIT> elements. Once the device recognizes the <PROGRAM MESSAGE TERMINATOR> and sends the <RESPONSE MESSAGE TERMINATOR>, the message exchange protocol is re-established.

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS Not for Resale

Annex A

(informative)

Message exchange control operation

This annex describes a state machine implementation which follows the rules in 6.3.5. The state machine diagram for this implementation is shown in Figure A.1. This figure is similar to Figure 6-2 of IEEE Std 488.2-1992.

At any time, the device is in one of the MEC states as shown in Figure A.1. Depending on the state, various functional elements shown in Figure 7 are active or inactive. Transitions are caused by the logical combination of messages from the functional elements.

A.1 Message exchange events

An event is something which happens once to trigger a transition. Events are shown in bold. The nine events are:

bav	a byte arrived at the I/O Control.
get	a group execute trigger, &GET, arrived at the I/O Control. Note that group execute trigger also generates an end of message. See subclause 6.1.6.2.2 of IEEE Std 488.2-1992 for further details.
brq	a byte was sent by the I/O Control.
dcas	a device clear (break) arrived on the I/O Control.
pon	the device was powered on.
eom	the Parser identified an LF as a <PROGRAM MESSAGE TERMINATOR>.
query	the Parser identified something which generates a response. See IEEE Std 488.2-1992.
RMT-sent	a <RESPONSE MESSAGE TERMINATOR> was sent.
ib-blocked	the Parser tried to get a byte from the Input Buffer, but it was empty.

A.2 Message exchange conditions

A condition has a logical value, TRUE or FALSE. Conditions are shown in italics. They are used with an event to determine which transition is taken. The six conditions are:

ib-empty	TRUE when the Input Buffer has nothing in it.
ib-full	TRUE when the Input Buffer is full so more data bytes generate input overflow and cannot generate another **bav**.
oq-full	TRUE when the Output Queue is full blocking the Response Formatter.
p-blocked	TRUE when the Parser is waiting for the Execution Control.
ec-blocked	TRUE when the Execution Control is waiting for the Device Functions.
rf-blocked	TRUE when the Response Formatter is waiting for room in the Output Queue.

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

A.3 Message exchange state machine

Transitions caused by the expected message exchange are shown with solid lines. Transitions which result from a violation of the message exchange protocol are shown with dashed lines. Certain transitions require the MEC to perform an action. The need to perform an action may result from a violation of the protocol. Actions associated with a transition are shown in Figure 7 below a line by the reason for the transition.

A.3.1 READ state

In the READ state the I/O Control is reading data bytes and GET messages from the interface and placing them in the Input Buffer. The Parser and Execution Control are active and the Output Queue is empty. A **brq** cannot occur while in the READ state.

The MEC remains in the READ state when either **dcas** or **pon** occurs. The device performs the INITIALIZE action described in 6.3.5.2.

The MEC remains in the READ state when **bav**, **get**, **eom**, or **ib-blocked** occurs.

The MEC enters the QUERY state when **query** occurs.

A.3.2 QUERY state

In the QUERY state the Parser has recognized a valid query within a <PROGRAM MESSAGE>. The I/O Control Emulator continues reading data bytes and GET messages from the interface and placing them in the Input Buffer.

The MEC remains in the QUERY state when **bav**, **get**, **query**, or **ib-blocked** occurs.

The MEC enters the SEND state when **brq** occurs.

The MEC enters the RESPONSE state when **eom** occurs and *ib-empty* is TRUE.

The MEC enters the READ state when an **eom** occurs and *ib-empty* is FALSE. This condition indicates that the controller started sending a new message before reading all of the response. The device performs the INTERRUPTED action described in 6.3.5.3.

The MEC enters the DEADLOCK state when *ib-full*, *oq-full*, *p-blocked*, *ec-blocked*, and *rf-blocked* are all TRUE. This condition indicates that the controller is sending more data to the device and expecting more data from the device than the device can buffer. The device performs the DEADLOCK action described in 6.3.5.5.

The MEC enters the READ state when either **dcas** or **pon** occurs. The device performs the INITIALIZE action described in 6.3.5.2.

A.3.3 SEND state

In the SEND state the device starts to send the <RESPONSE MESSAGE> to the controller. The device continues to receive data bytes from the interface, and to parse and execute the Query Message stored in the Input Buffer. The Response Formatter prepares the <RESPONSE MESSAGE>, and places it in the Output Queue.

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS Not for Resale

A deadlock situation is possible in the SEND state. The controller or communication channel may buffer part of the response even through the application has not requested any data. This buffering is actually an extension of the Output Queue, but the device has no control over it. If this extra buffer should fill, so the device is prevented from sending more data while the controller is still sending the <PROGRAM MESSAGE>, the Input Buffer may also fill, thus deadlocking the message exchange.

The device could possibly detect this situation, but it cannot clear the extra buffers, so it merely stays in the SEND state until the controller clears the deadlock.

The MEC remains in the SEND state when **bav**, **get**, **brq**, **query** or **ib-blocked** occurs.

The MEC remains in the SEND state when *ib-empty* becomes TRUE. This transition might be considered UNTERMINATED since it appears that the controller failed to send an **eom** before it started to read the response. The nature of a full-duplex serial interface, however, allows bytes to be transferred and buffered in both directions. The **eom** may still be on its way from the controller so the device waits for it in SEND.

The MEC enters the RESPONSE state when **eom** occurs and *ib-empty* is TRUE.

The MEC enters the READ state when **eom** occurs and *ib-empty* is FALSE. This condition indicates that the controller started sending a new message before reading all of the response. The device performs the INTERRUPTED action described in 6.3.5.3.

The MEC enters the READ state when either **dcas** or **pon** occurs. The device performs the INITIALIZE action described in 6.3.5.2 of this standard.

A.3.4 RESPONSE state

In the RESPONSE state the Parser has received a <PROGRAM MESSAGE TERMINATOR> and the device is sending the complete <RESPONSE MESSAGE> to the controller. The Response Formatter formats the <RESPONSE MESSAGE> and the I/O Control sends data bytes from the Output Queue to the controller. An **eom**, **query**, or **ib-blocked** cannot occur because the Input Buffer is empty.

The MEC remains in the RESPONSE state when a **brq** occurs.

The MEC enters the READ state when **RMT-sent** occurs.

The MEC enters the READ state when **bav** or **get** occurs. This condition indicates that the controller started sending a new message before reading all of the response. The device performs the INTERRUPTED action described in 6.3.5.3 of this standard.

The MEC enters the READ state when either **dcas** or **pon** occurs. The device performs the INITIALIZE action described in 6.3.5.2 of this standard.

A device may have nothing to say while the MEC is in the RESPONSE state if all of the queries in the Query Message failed to generate any <RESPONSE DATA> because of syntax (Command), semantic (Execution), or Device-specific errors. The device does not send a <RESPONSE MESSAGE TERMINATOR> to the controller. The MEC remains in the RESPONSE state until either **bav**, **get**, **pon**, or **dcas** occurs.

A.3.5 DEADLOCK state

In the DEADLOCK state the device has been asked to buffer more data than it has room to store. The Output Queue was full, blocking the Response Formatter, Execution Control, and Parser. The Input Buffer was full, and the controller was waiting to send more data bytes to the device.

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

The MEC breaks the deadlock by clearing the Output Queue and resetting the Response Formatter. The MEC also reports a Query Error to the Status Reporting. While in the DEADLOCK state, the MEC continues to parse and execute <PROGRAM MESSAGE UNIT> elements as usual, except that it discards query responses rather than placing them into the Output Queue.

Because the Output queue is kept empty in DEADLOCK, **brq** can never occur.

The MEC remains in the DEADLOCK state when a **bav**, **get**, **query** or, **ib-blocked** occurs.

The MEC enters the READ state when **eom** occurs.

The MEC enters the READ state when either **dcas** or **pon** occurs. The device performs the INITIALIZE action, described in 6.3.5.2 of this standard.

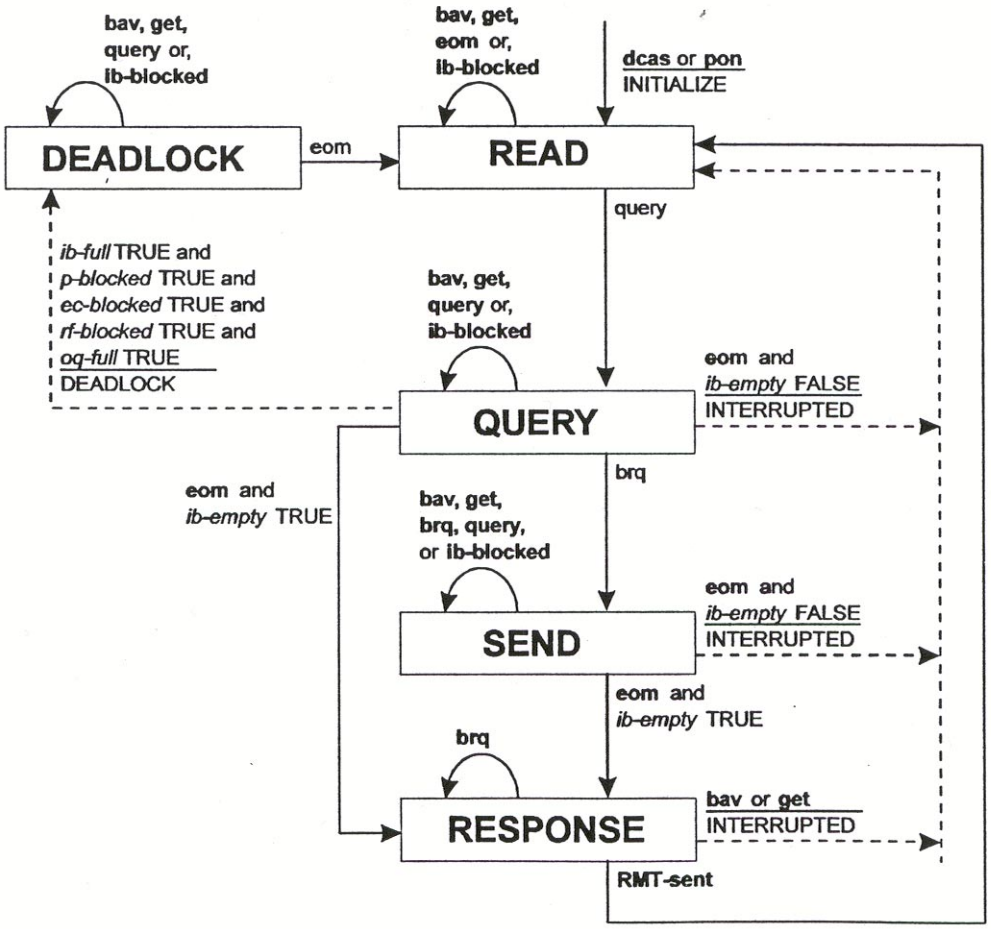

Figure A.1—Message exchange control state diagram

Copyright © 2001 IEEE. All rights reserved.

Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale

Annex B

(informative)

IEEE 1174 Performance classification

IEEE 1174 interfaces can be equipped with several functional capabilities. For that purpose, the standard groups the interface functionality in several performance classes. The device designer can place any device capability information at the appropriate places on the physical equipment and in the relevant device documentation. Table B.1 summarizes the allowable Performance Classes to an IEEE 1174 interface.

Table B.1—IEEE 1174 Performance Classes

Identification	Description	Capability	Other capabilities required
CAP1	Performance Class 1	Base functionality	none
CAP2	Performance Class 2	IEEE 488.1 emulation	CAP1
CAP3	Performance Class 3	Full capability	CAP1 and CAP2

Copyright © 2001 IEEE. All rights reserved.
Copyright The Institute of Electrical and Electronics Engineers, Inc.
Provided by IHS under license with IEEE
No reproduction or networking permitted without license from IHS
Not for Resale