

1451.3™

IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems

IEEE Instrumentation and Measurement Society

Sponsored by the
Technical Committee 9 on Sensor Technology (TC-9)



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

31 March 2004

Print: SH95174
PDF: SS95174

IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems

Sponsor

**TC-9 on Sensor Technology of the
IEEE Instrumentation and Measurement Society**
in cooperation with the
**National Institute of Standards and Technology (NIST)
United States Department of Commerce**

Approved 11 September 2003

IEEE-SA Standards Board

Approved 29 December 2003

American National Standards Institute

Abstract: A digital interface for connecting multiple physically separated transducers to a single processor over a single pair of wires. The interface can support both asynchronous and isochronous data transfers. Several Transducer Electronic Data Sheets (TEDS) and their data formats are described. An electrical interface, channel identification protocols, time synchronization protocols, read and write logic functions to access the TEDS, and transducers with a wide variety of attributes are defined. This standard does not specify signal conditioning, signal conversion, or how an application uses the TEDS data.

Keywords: 1451, communication protocol, digital interface, event sensor, microprocessor, NCAP, plug and play, sensor interface, smart actuator, smart sensor, smart sensor interface, smart transducer interface, TBIM, TEDS

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

Copyright © 2004 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 31 March 2004. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the institute of Electrical and Electronics Engineers, Inc.

HomePNA is a trademark of Home Phoneline Networking Alliance.

Print: ISBN 0-7381-3822-3 SH95174
PDF: ISBN 0-7381-3823-1 SS95174

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.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied “**AS IS.**”

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.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

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 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. Current interpretations can be assessed at the following URL: <http://standards.ieee.org/reading/ieee/interp/index.html>.

Errata, if any, for this and all other standards can be assessed at the following URL: <http://standards.ieee.org/reading/ieee/updates/errata/index.html>. Users are encouraged to check this URL for errata periodically.

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. 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.

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 the Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923, USA; + 1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

[This introduction is not part of IEEE Std 1451.3-2003, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems.]

The main objectives of this standard are to:

- Enable plug and play at the transducer level by providing a common communication interface for transducers which are physically separated
- Enable and simplify the creation of groups of networked smart transducers
- Facilitate the support of multiple networks

The existing fragmented sensor market is seeking ways to build low-cost, networked smart sensors. Many sensor network or fieldbus implementations are currently available, each with its own strengths and weaknesses for a specific application class. Interfacing transducers to all these control networks and supporting the wide variety of protocols represents a significant and costly effort to transducer manufacturers. A universally-accepted transducer interface standard would not only allow for the development of smart sensors and actuators, it could also lead to lower development costs. Therefore, the goal of this standard is not to propose another control network, but to define a smart transducer interface that will isolate the choice of transducers from the choice of networks. This would relieve the burden from the manufacturer of supporting a cross product of sensors versus networks, and would help to preserve the user's investment if it becomes necessary to migrate to a different network standard.

There is currently only the IEEE Std 1451.2TM-1997^a common digital communication interface standard between transducers and Network Capable Application Processors (NCAPs). IEEE Std 1451.2-1997 supports transducers that can be physically included within one housing, but it does not support transducers that are physically separated but need to make their network connection through a single NCAP. IEEE Std 1451.2-1997 does, however, provide a comprehensive set of TEDSs that are the basis for the TEDS described in this document. Care has been taken throughout the development of this standard to modify the IEEE Std 1451.2-1997 TEDS only as required to meet different requirements of the different environment and to provide for a translation path between the IEEE Std 1451.3-2003 TEDS and the IEEE Std 1451.2-1997 TEDS.

IEEE Std 1451.3-2003 utilizes the techniques designed to implement networking in the home by interconnection devices on the telephone lines. For IEEE Std 1451.3-2003, a single pair of conductors will be used to provide the following functions:

- synchronized data acquisition for an array of transducers
- communicating with an array of Transducer Bus Interface Modules (TBIM)
- providing power for operation of transducers on the bus and their associated electronics

Transducers built per this standard can be plugged into an IEEE Std 1451.3-2003 compatible system and be used without having to add special drivers, profiles or make any other changes to the system. The IEEE Std 1451.3-2003 TEDS provides for self-identifying transducers. The TEDS contains fields that fully describe the type, operation, and attributes of one or more transducers (sensors or actuators). By having the TEDS associated with the transducer and module containing the transducer, the resulting hardware partition encapsulates the measurement aspects to a module. The application related aspects of the measurement are on the bus controller or NCAP and do not become a concern for the transducer manufacturer.

^aInformation on references can be found in Clause 2.

Data output over the bus may be in integer; single precision real or double precision real formats. The data is passed to the bus controller and from the bus controller to the rest of the system. Further processing of this data may take place both in the bus controller and in other processors in the larger system. Throughout this document it is assumed, but not required, that all processing will be performed on data in a single or double precision real format. All fields in the TEDS are specified based on the assumption that, unless specifically stated to the contrary, all data will be converted to single or double precision real before any processing is performed.

This standard provides areas that are “open to manufacturers.” It should be noted that any use of these areas compromises the “plug and play” potential of bus controllers and TBIMs.

The IEEE Std 1451.3-2003 transducer interface is adaptable to the IEEE Std 1451.2-1997 interface and compatible with the IEEE Std 1451.1TM-1999 information model standard.

Participants

At the time this standard was completed, the Distributed Multidrop Systems Working Group had the following membership:

Larry A. Malchodi, *Chair*

Stephen Smith, *Vice Chair*

Daniel Maxwell, *Secretary*

Lee Eccles, *Editor*

Jørgen Bække
L. Wayne Catlin
Fernando GenKuong
Robert Johnson
Charles Jones
James Lathrop

Alice Law
Kang Lee
Roberto Lenarduzzi
Torben Licht
Allan Owen
David Perrussel

Jim Read
Andrew Segal
Robert Sinclair
Mark Slack
Charles Summey
Rich Valde

Other individuals who have contributed to this standard are as follows:

Dennis Albertsen
Gregory Balchin
Arati Baruah
Thurston Brooks
Mark Buckner
Russell Dominique
Craig Evensen
Brett Gidge

David Goetzinger
Richard Hambly
Rob Hammett
James Hooper
Paul Hufnagel
Richard Jones
Rodney Juelfs
Eric Lange
Carlos Lopez-Reyna

Paul Moffitt
Mike Moore
Chao Pang
Ray Ritmiller
Mark Schiefer
T. J. Shahabi
Bryan Tracy
Jay Zemel

The following members of the balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Martin Baur
Jacob Ben Ary
L. Wayne Catlin
Keith Chow
Guru Dutt Dhingra
Fernando GenKuong

Richard Hambly
Charles Jones
James Kemerling
Kang Lee
Gregory Luri
Tremont Miao

Gary Michel
Mark I. Schiefer
Robert Sinclair
Stephen Smith
Steven Tilden
Stephen C. Webb

When the IEEE-SA Standards Board approved this recommended practice on 11 September 2003, it had the following membership:

Don Wright, *Chair*

Howard M. Frazier, *Vice Chair*

Judith Gorman, *Secretary*

H. Stephen Berger

Joseph A. Bruder

Bob Davis

Richard DeBlasio

Julian Forster*

Toshio Fukuda

Arnold M. Greenspan

Raymond Hapeman

Donald N. Heirman

Laura Hitchcock

Richard H. Hulett

Anant Kumar Jain

Lowell G. Johnson

Joseph L. Koepfinger*

Thomas J. McGreen

Steve M. Mills

Daleep C. Mohla

William J. Moylan

Paul Nikolich

Gray S. Robinson

Malcolm V. Thaden

Geoffrey O. Thompson

Doug Topping

Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Alan Cookson, *NIST Representative*

Satish K. Aggarwal, *NRC Representative*

Savoula Amanatidis

IEEE Standards Managing Editor

Contents

1. Overview.....	1
1.1 Scope.....	2
1.2 Purpose	2
1.3 Conformance.....	2
2. References	3
3. Definitions, acronyms, and abbreviations.....	5
3.1 Acronyms and abbreviations	5
3.2 Definitions.....	6
4. Data types	9
4.1 Unsigned octet integer for counting	9
4.2 Unsigned octet integer for enumeration	9
4.3 Unsigned 16-bit integer for counting.....	10
4.4 Unsigned 16-bit integer for field length.....	10
4.5 Signed 32-bit integer	10
4.6 Unsigned 32-bit integer for counting.....	10
4.7 Unsigned 32-bit integer for field length.....	10
4.8 Single precision real	11
4.9 Double precision real.....	11
4.10 String.....	11
4.11 Physical units	11
4.12 Universal unique identification	12
4.13 Arbitrary octet array.....	13
4.14 Time-of-day.....	14
5. Smart transducer functional specification.....	14
5.1 Plug and play capability	15
5.2 Addresses	15
5.3 Common characteristics.....	16
5.4 TransducerChannel type descriptions	17
5.5 Embedded TransducerChannels.....	20
5.6 TransducerChannel groups	21

5.7	TransducerChannel proxy.....	21
5.8	Attributes and operating modes	22
5.9	Triggering.....	26
5.10	Synchronization	29
5.11	CommunicationsChannels.....	30
5.12	Status	31
5.13	Service request logic.....	37
5.14	Hot-swap capability	38
6.	Commands	38
6.1	Standard commands	39
6.2	Manufacturer-defined commands	53
7.	Transducer Electronic Data Sheet (TEDS) specification.....	53
7.1	General format for TEDS.....	53
7.2	Transmission of the TEDS	54
7.3	Meta-TEDS.....	54
7.4	TransducerChannel TEDS.....	60
7.5	Calibration TEDS.....	73
7.6	Frequency Response TEDS.....	82
7.7	Transfer Function TEDS.....	84
7.8	Text-based TEDS.....	90
7.9	End user application specific TEDS	94
7.10	Commissioning TEDS	94
7.11	Manufacturer-defined TEDS	95
7.12	PHY TEDS	95
8.	Upper layers of the ISO model (layer 3 through layer 7)	97
8.1	Data transmission order and bit significance	97
8.2	Protocol identifiers.....	99
8.3	Datagram protocol	99
8.4	Streaming data protocol	102
8.5	Trigger protocol.....	102
8.6	Command services protocol	104
8.7	Reply protocol.....	104
9.	Standard services offered by the Data Link Layer	106
9.1	Syntax and semantics.....	106

9.2	Streaming mode management services on the TBC	107
9.3	Streaming mode protocol services	110
9.4	Discovery services in the TBC.....	112
9.5	Discovery services in the TBIM	113
9.6	Transmitter control services	114
9.7	Datagram protocol services	115
9.8	Trigger protocol services.....	117
9.9	Delay measurement services	119
10.	Lower layers of the ISO model (layer 1 and layer 2)	120
10.1	Data Link Layer protocol specifications	121
10.2	Data link—Logical Link Control (LLC) sublayer	124
10.3	Data link—MAC sublayer.....	125
10.4	Physical layer	137
10.5	Physical specifications	141
	Annex A (informative) Bibliography	147
	Annex B (normative) XML schema for text-based TEDS	148
	Annex C (informative) Example Meta-Identification TEDS	153
	Annex D (informative) Example TransducerChannel Identification TEDS.....	154
	Annex E (informative) Example Calibration Identification TEDS	156
	Annex F (informative) Example Commands TEDS.....	157
	Annex G (informative) Example Location and Title TEDS	160
	Annex H (informative) Example physical units.....	162
	Annex I (informative) TEDS Read and Write protocols	169
	Annex J (informative) Trigger logic configurations.....	171

IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems

1. Overview

This standard introduces the concept of a Transducer Bus Interface Module (TBIM) and a Transducer Bus Controller (TBC) connected by a transducer bus. A TBIM is a module that contains the bus interface, signal conditioning, Analog-to-Digital and/or Digital-to-Analog conversion and in many cases the transducer. A TBIM can range in complexity from a single sensor or actuator to units containing many transducers (sensors and actuators). A TBC is the hardware and software in the Network Capable Application Processor (NCAP) or host processor that provides the interface to the transducer bus. The transducer bus provides the communications path between an NCAP or host processor and one or more TBIMs. Three types of transducers are recognized by this standard. They are sensors, event sensors, and actuators.

A TransducerChannel is denoted *smart* in this context because of three features:

- It is described by a machine-readable, Transducer Electronic Data Sheet (TEDS).
- The control and data associated with the TransducerChannel are digital.
- Triggering, status, and control are provided to support the proper functioning of the TransducerChannel.

A TBC, that may be part of an NCAP or host processor, controls a TBIM by means of a dedicated digital bus. The bus controller or NCAP mediates between the TBIM and a higher-level digital network, and may provide local intelligence.

This standard provides for TBIMs that can be plugged into a system and be used without having to add special drivers, profiles, or make any other changes to the system. This is referred to as *plug and play* operation. The primary features that enable plug and play operation are the TEDS and the basic command set. A TBIM may be added to or removed from an active transducer bus with no more

than a momentary impact on the data being transferred over the bus. “Hot swap” is the term used to refer to this feature.

This standard is organized as follows: Clause 1: “Overview” provides the scope of this standard. Clause 2: “References” lists references to other standards and documents that are useful in applying this standard. Clause 3: “Definitions, acronyms, and abbreviations” provides definitions that are either not found in other standards, or have been modified for use with this standard. Clause 4: “Data types” defines the data types used in the standard. Clause 5: “Smart transducer functional specification” specifies the functions required of a TBIM and of each TransducerChannel it comprises. Clause 6: “Commands” provides the command syntax and the expected replies. Clause 7: “Transducer Electronic Data Sheet (TEDS) specification” specifies the transducer electronic data sheet structure. Clause 8: “Upper layers of the ISO model (layer 3 through layer 7)” specifies the protocols used for the interface between the TBIM and the bus controller. Clause 9 “Standard services offered by the Data Link Layer” specifies the services provided by the Data Link Layer to the upper layers of the protocol stack. Clause 10: “Lower layers of the ISO model (layer 1 and layer 2)” specifies the physical transducer interface between the TBIM and the bus controller and the lower layers of the protocol stack.

1.1 Scope

This standard defines a digital interface for connecting multiple physically separated transducers. It leverages off the IEEE Std 1451.1TM-1999 and IEEE Std 1451.2TM-1997 standards. This standard defines the TEDS format, the electrical interface, channel identification protocols, hot swap protocols, time synchronization protocols, and the read and write logic functions used to access the TEDS and transducer data. This standard does not specify signal conditioning, signal conversion, or how an application uses the TEDS data.

1.2 Purpose

There is currently no defined independent standard for interfacing multiple physically separated transducers that allows time synchronization of data. Without such a standard, custom transducer interface solutions are required which are time-consuming and costly. This standard provides a minimum implementation that allows multidrop, hot swapping, self-identification and configuration of transducers that may not be located in the same enclosure, but are confined to a relatively small space.

1.3 Conformance

The philosophy underlying the conformance requirements of this subclause is to provide the structure necessary to raise the level of interoperability of transducers and systems built to this standard, while leaving open opportunity for continued technical improvement and differentiation.

A TBIM implementation shall be deemed in conformance with IEEE Std 1451.3-2003 provided the following requirements are met:

- The TBIM supports the required functional performance specified in Clause 5.
- The TBIM supports the required commands specified in Clause 6.
- The TBIM supports TEDS which have the format specified in Clause 7.
- The TBIM’s physical interface implements the interface and protocols defined in Clause 10.

As a minimum, a compliant TBIM shall implement the following protocols specified in Clause 8:

- Datagram protocol
- Trigger protocol

- Command services protocol
- Reply protocol

NOTE—There are several features that are highly desirable, and are supported by this standard, but are not practical to make into hard requirements. It is desirable that the sense element for a sensor be an integral part of the TBIM, but for sensing elements like structural strain gauges and thermocouples this is not practical so it has not been made into a hard requirement. In addition, it is very desirable that the TEDS be located within the TBIM, but there are systems where the environment and/or the physical size make this impractical so the standard allows the TEDS to be located remote from the TBIM.

As a minimum, a compliant TBIM shall implement the functionality required by the following services specified in Clause 9:

- Discovery services in the TBIM
- Transmitter control services
- Datagram protocol services
- Trigger protocol services

A TBC implementation shall be deemed in conformance with this standard provided it supports the hardware and/or software required by the services required in Clause 9, the physical interface defined in Clause 10, and the following protocols specified in Clause 8:

- Datagram protocol
- Trigger protocol
- Command services protocol

1.3.1 Conformance keywords

Several keywords are used to differentiate among various levels of requirements and optionality, as follows:

1.3.1.1 Shall

The key word “shall” indicates a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to IEEE Std 1451.3-2003.

1.3.1.2 Recommended

“Recommended” is a key word indicating flexibility of choice with a strong preference alternative. The word “should” has the same meaning.

1.3.1.3 Should

“Should” is a key word indicating flexibility of choice with a strong preference alternative. The phrase “it is recommended” has the same meaning.

1.3.1.4 May

“May” is a key word that indicates flexibility of choice with no implied preference.

2. References

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

ANSI INCITS 4-1986 (Reaff 2002), Information Systems—Coded Character Sets—7-Bit American National Standard Code for Information Interchange (7-Bit ASCII).¹

EN 60947-5-2-1999, Low-Voltage Switchgear and Controlgear—Part 5: Control Circuit Devices and Switching Elements—Section 2 Proximity Switches (E).²

Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, 6 October 2000.³

IEC 60603-7-1996-11, Connectors for Frequencies Below 3 MHz for Use with Printed Boards—Part 7: Detail Specification for Connectors, 8-Way, Including Fixed and Free Connectors with Common Mating Features, with Assessed Quality.⁴

IEEE Std 754TM-1985 (Reaff 1990), IEEE Standard for Binary Floating-Point Arithmetic.^{5,6}

IEEE Std 802.3TM-2002, IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

IEEE Std 1451.1-1999, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Network Capable Application Processor (NCAP) Information Model.

IEEE Std 1451.2-1997, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats.

IEEE Std 1588TM-2002, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.

ISO 639-1:2002, Codes for the Representation of Names of Languages—Part 1: Alpha-2 Code.⁷

ISO 639-2:1998, Codes for the Representation of Names of Languages—Part 2: Alpha-3 Code.

ISO/IEC 14750:1999, Information Technology—Open Distributed Processing—Interface Definition Language.

ITU-T Recommendation G.989.1-2001, Phoneline Networking Transceivers—Foundation.⁸

¹ANSI publications are available from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (<http://www.ansi.org/>).

²EN publications are available from the European Committee for standardization (CEN), 36, rue de Stassart, B-1050 Brussels, Belgium (<http://www.cenorm.be>).

³XML 1.0 Second Edition can be obtained from the World Wide Web Consortium (<http://w3c.org>).

⁴IEC publications are available from the Sales Department of the International Electrochemical Commission, Case Postale 131, 3, rue de Varembe, CH-1211, Geneva 20, Switzerland/Suisse (<http://www.iec.ch/>). IEC publications are also available in the United States from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (<http://www.ansi.org/>).

⁵The IEEE standards or products referred to in Clause 2 are trademarks owned by the Institute of Electrical and Electronics Engineers, Inc.

⁶IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, P.O. Box 1331 Piscataway, NJ 08855 1331, USA (<http://standards.ieee.org/>).

⁷ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembe, CH-1211, Geneva 20, Switzerland/Suisse (<http://www.iso.ch/>). ISO/IEC publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (<http://global.ihs.com/>). Electronic copies are available in the United States from the American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (<http://www.ansi.org/>).

⁸ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (<http://www.itu.int/>).

ITU-T Recommendation G.989.2-2001, Phoneline Networking Transceivers—Payload Format and Link Layer Requirements.

MIL-STD-1560A, Insert Arrangements for MIL-C-38999, MIL-C-29600 Series A, and MIL-C-27599 Electrical, Circular Connectors.⁹

MIL-DTL-38999, General Specification for Connectors, Electrical, Circular, Miniature, High Density, Quick Disconnect (Bayonet, Threaded, and Breech Coupling), Environment Resistant, Removable Crimp and Hermetic Solder Contacts.

NIST Technical Note 1297-1994, Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results.¹⁰

TIA/EIA 568-B.2-2001, Commercial Building Telecommunications Cabling Standard—Part 2: Balanced Twisted-Pair Cabling Components.¹¹

3. Definitions, acronyms, and abbreviations

This clause provides definitions that are either not found in other standards, or have been modified for use with this standard. Also included in this clause are definitions of data types, string language codes, compact physical units, and a universal unique identification system.

3.1 Acronyms and abbreviations

This subclause contains abbreviations of key terms as they are used in this standard. *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition [B2]¹² should be referenced for acronyms and abbreviations not defined in this clause.

ACK	acknowledgment
AD	anno domini
ADSL	asymmetrical digital subscriber line
AKE	acknowledgment expected
BPSK	binary phase-shift keying
CDMA	code-division multiple access
CSMA/CD	carrier sense multiple access/collision detection
DSSS	direct sequence spread spectrum
DTD	document type declaration, ref. Extensible Markup Language (XML)
EIA	Electronic Industries Associates
HomePNA™	Home Phoneline Networking Alliance ¹³
IFG	interframe gap
MAC	media access control
NaN	not a number
NCAP	Network Capable Application Processor
PHY	physical layer
PN	pseudo noise
PSN	predicted sequence number

⁹MIL publications are available from Customer Service, Defense Printing Service, 700 Robins Avenue, Bldg. 4D, Philadelphia, PA 19111-5094, USA.

¹⁰NIST publications are available from Global Engineering Documents, 15 Inverness Way East, Englewood, CO 80112, USA (<http://global.ihs.com/>).

¹¹TIA/EIA publications are available from Global Engineering Documents, 15 Inverness Way East, Englewood, CO 80112, USA (<http://global.ihs.com/>).

¹²The numbers in brackets correspond to those of the bibliography in Annex A.

¹³HomePNA is a trademark of Home Phone Networking Alliance.

SI	international system of units, reference <i>The International System of Units (SI)</i> [B4]
TBC	Transducer Bus Controller
TBIM	Transducer Bus Interface Module
TEDS	Transducer Electronic Data Sheet
TIA	Telecommunications Industries Association
ToD	time-of-day
UTC	universal coordinated time
UUID	universal unique identifier
Xdc	transducer
XML	extensible markup language

3.2 Definitions

For the purpose of this standard, the following terms and definitions apply. *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition [B2], should be referenced for terms not defined in this clause.

3.2.1 acknowledge: A reply message used to indicate that a command or trigger message was received.

3.2.2 actuator: A transducer that accepts a data sample or samples and converts them into physical action.

3.2.3 address: A character or group of characters that identifies a register, a particular part of storage, or some other data source or destination.

3.2.4 AddressGroup: A collection TransducerChannels that respond to a single address.

3.2.5 alias: An alternate designation for an object.

NOTE—Throughout this standard, it is a number used in place of the 80-bit UUID when addressing a TBIM.

3.2.6 buffer: An intermediate data storage location used to compensate for the difference in rate of flow of data or time of occurrence of events when transmitting information from one device to another.

3.2.7 bus controller: A device, possibly an NCAP, that acts as the bus master to control all of the devices on a transducer bus.

3.2.8 calibration: The process used to determine the information that resides in the Calibration TEDS to support correction.

3.2.9 code-division multiple access: A distinct spread spectrum encoding [in this case direct-sequence spread spectrum (DSSS)], to distinguish between different transmission channels that occur at the same time on the same carrier frequency. Other forms of multicarrier or multiple-frequency modulation techniques commonly categorized as spread spectrum, such as frequency hopping or orthogonal frequency-division multiplexing, are specifically not excluded but may also be implemented as adjuncts to the basic data transmission formats defined in the physical layer.

3.2.10 CommunicationsChannel: A communications path between the TBC and the TBIM. This functionality is used for data communications and other functions as well as controlling the operation of a TBIM.

3.2.11 ControlGroup: Manufacturer specifications that define the inherent relationships between the TransducerChannels of a multichannel TBIM. This grouping information is not normally used by the TBIM itself. This information is used to identify TransducerChannels that are used to control some characteristic of another TransducerChannel.

3.2.12 correction: The evaluation of a multinomial function using information from the Calibration TEDS together with data from one or more transducers.

3.2.13 Data CommunicationsChannel: The term used to denote a CommunicationsChannel when the logical properties are what are under consideration rather than the physical properties.

3.2.14 data conversion: The translation of data from one numeric form into another (e.g., converting a scaled integer to a floating point number during correction).

3.2.15 data model: The numeric format in which the TBIM shall output or accept data.

3.2.16 data set: The collection of samples acquired by a sensor (or applied by an actuator) in response to a trigger message.

3.2.17 data sheet: A set of information on a device that defines the parameters of operation and conditions of usage (usually produced by the device's manufacturer).

3.2.18 data structure: A group of digital data fields organized in some logical order for some specific purpose. A two-dimensional paper version is an empty fill-in-the-blanks form or an empty tabular chart with organized column and row headings. The template used when data is stored in computer memory.

3.2.19 digital interface: A set of wires and a protocol for transferring information by binary means only.

3.2.20 electronic data sheet: A data sheet stored in some form of electrically readable memory (as opposed to a piece of paper).

3.2.21 embedded transducer: A device that behaves as a transducer from the point of view of the bus controller even though nothing outside of the TBIM is sensed or changed.

3.2.22 enumeration: The process of declaring the encoding of human interpretable information in a manner convenient for digital electronic machine storage and interchange. This involves the assignment of a numeric value to a specific meaning within the context of a specific data field. Binary numbers are usually expressed in decimal terms for human convenience. Not all possible numeric values need have a specific meaning. Values without meaning are declared to be unused or reserved for future use. The meanings encoded in each data field shall be specific and unique to that data field and only that data field. The value becomes meaningless if not associated with the data field and its defining table.

3.2.23 epoch: The time interval measured from some received initial signal or message to a terminating signal or message. The terminating signal or message may be an independent signal or message or may be a second instantiation of the initial signal or message. The definition of the defining signal or message is a function of the physical layer.

3.2.24 event sensor: A sensor that detects a change of state in the physical world. The fact that a change of state has occurred and/or instant in time of the change of state, not the state value, is the *measurement*.

3.2.25 hot swap: The act of connecting or disconnecting a TBIM from a transducer bus without first turning off the power that is supplied to the TBIM over the bus.

3.2.26 isochronous transmission: A communications stream transport that is uniform in time. The delivery of the physical stream of information is recurring at regular intervals.

3.2.27 least significant bit (lsb): The bit in the binary notation of a number that is the coefficient of the lowest exponent possible.

3.2.28 message: Information that is to be passed between devices as a single logical entity. A message may occupy one or more packets.

3.2.29 meta-: A Greek prefix meaning: that which pertains to the whole or overall entity or that which is in common or shared with all member entities comprising the whole.

3.2.30 Meta-TEDS: The collection of those TEDS data fields that pertain to the whole or overall entity or those which are in common or shared with all member entities (TransducerChannels) comprising the whole product.

3.2.31 multinomial: A linear sum of terms involving powers of more than one variable.

$$\sum_{i_1=0}^{N_1} \sum_{i_2=0}^{N_2} \cdots \sum_{i_m=0}^{N_m} A(i_1, i_2, \dots, i_m) x_1^{i_1} x_2^{i_2} \cdots x_m^{i_m}$$

3.2.32 Network Capable Application Processor (NCAP): A device between the transducer bus and the network that performs network communications, TBIM communications, and data conversion or other processing functions.

3.2.33 not-a-number (NaN): As defined in IEEE Std 754-1985, a bit pattern of a single or double precision real number data type that is a result of an invalid floating point operation.

3.2.34 octet: A group of eight bits.

3.2.35 packet: Information that is to be passed by the physical layer between devices in a single transmission.

3.2.36 Physical CommunicationsChannel: The term used to denote a CommunicationsChannel when the physical properties are what are under consideration.

3.2.37 read frame: The process of transferring of one block of data from a TBIM to the bus controller.

3.2.38 sample latched: The term used in a sensor to identify that the sample has been acquired. This may be when a sample and hold circuit switches to the hold mode or other similar operation. For an actuator, it signals that the sample has been moved to the output logic.

3.2.39 sensor: A transducer that converts a physical, biological, or chemical parameter into an electrical signal.

3.2.40 setup time: The time between the initial request for a function to be performed and when the task is actually initiated.

3.2.41 signal conditioning: The transducer signal processing that involves operations such as amplification, compensation, filtering, and normalization.

3.2.42 smart actuator: An actuator version of a smart transducer.

3.2.43 smart sensor: A sensor version of a smart transducer.

3.2.44 smart transducer: A transducer that provides functions beyond those necessary for generating a correct representation of a sensed or controlled quantity. This functionality typically simplifies the integration of the transducer into applications in a networked environment.

3.2.45 synchronization signal: A signal transmitted by the TBC to provide clock or time synchronization signals to a TBIM or group of TBIMs.

3.2.46 transceiver: A device that is capable of both transmitting and receiving information.

3.2.47 transceiver chip set: An integrated circuit or set of integrated circuits that is capable of being used as a transceiver.

3.2.48 transducer: A device that converts energy from one domain into another. The device may either be a sensor or an actuator.

3.2.49 transducer bus: A local bus used to interconnect TBIM with a TBC.

3.2.50 Transducer Bus Controller (TBC): The part of the NCAP or other master device that forms the interface to the transducer bus. It may consist of both hardware and software.

3.2.51 Transducer Bus Interface Module (TBIM): A module that contains the TEDS, logic to implement the transducer bus interface, the transducer(s) or connection to the transducer(s) and any signal conversion or signal conditioning.

3.2.52 TransducerChannel: A transducer and all of the signal conditioning and conversion components associated with that transducer.

3.2.53 TransducerChannel address: The concatenation of the TBIM Alias in the most significant octet position and the TransducerChannel number that form the logical address of a specific TransducerChannel.

3.2.54 TransducerChannel number: An 8-bit number assigned to an individual TransducerChannel within a TBIM by the manufacturer.

3.2.55 TransducerChannel proxy: A device that is created to allow a collection of TransducerChannels to be treated as a single entity. It is similar to a normal TransducerChannel except that it does not require a TransducerChannel TEDS, it cannot have a Calibration TEDS, Transfer Function TEDS, or Frequency Response TEDS. It may support other TEDS and may respond to commands.

3.2.56 Transducer Electronic Data Sheet (TEDS): An electronic data sheet describing a TransducerChannel. The structures of multiple TEDS are described in this standard.

3.2.57 transfer: The act or process of moving information from one digital device to another.

3.2.58 trigger: A signal or message that is used to start an action.

3.2.59 trigger cycle: The complete cycle comprising the assertion of the trigger by the bus controller followed by the acknowledgment by the TBIM.

3.2.60 VectorGroup: Manufacturer specifications that define the inherent relationships between the TransducerChannels of a multichannel TBIM.

3.2.61 write frame: The process that implements the transfer of data from a bus controller to a TBIM.

4. Data types

All data types used throughout the remainder of this standard are defined in subordinate subclauses.

4.1 Unsigned octet integer for counting

Symbol: U8C

Size: 1 octet

This data type represents positive counting integers from zero to 255.

4.2 Unsigned octet integer for enumeration

Symbol: U8E

Size: 1 octet

All 1-octet enumerations are unsigned integers, with a value between zero and 255. Each enumerated value denotes a context-sensitive meaning.

4.3 Unsigned 16-bit integer for counting

Symbol: U16C

Size: 2 octets

This data type is used to represent positive counting integers from zero to 65 535.

4.4 Unsigned 16-bit integer for field length

Symbol: U16L

Size: 2 octets

This data type is used to represent unsigned integers from zero to 65 535. The value shall be interpreted as the length in octets of an associated field or data block.

When used to specify the length of a data block (more than one field), the length field shall not include the length of the length field itself.

4.5 Signed 32-bit integer

Symbol: S32

Size: 4 octets

This data type is used to represent a signed integer from $-2\,147\,483\,648$ to $2\,147\,483\,647$.

4.6 Unsigned 32-bit integer for counting

Symbol: U32C

Size: 4 octets

This data type is used to represent a positive counting integer from zero to 4 294 967 295.

4.7 Unsigned 32-bit integer for field length

Symbol: U32L

Size: 4 octets

This data type is used to represent unsigned integers from zero to 4 294 967 295. The value shall be interpreted as the length in octets of an associated field or data block.

When used to specify the length of a data block (more than one field), the length field shall not include the length of the length field itself.

4.8 Single precision real

Symbol: F32

Size: 4 octets

A single precision real number is a 32-bit binary sequence that encodes real numbers as specified in IEEE Std 754-1985.

4.9 Double precision real

Symbol: F64

Size: 8 octets

A double precision real number is a 64-bit binary sequence that encodes real numbers as specified in IEEE Std 754-1985.

4.10 String

The use of text strings as defined in IEEE Std 1451.2-1997 has been replaced with the use of the Extensible Markup Language (XML) in this standard. The controlling document for XML is the W3C Recommendation Extensible Markup Language (XML) 1.0 (Second Edition). All text strings in the TEDS shall conform to this recommendation or a future update to it.

4.11 Physical units

Symbol: UNITS

Size: 10 octets

Physical units are a binary sequence of 10 octets that encode physical units as described in Table 1 and Table 2. Each of the fields shall be interpreted as an unsigned integer. A unit shall be represented as a product of the SI base units, plus radians and steradians, each raised to a rational power. This structure encodes only the exponents; the product is implicit. For further information, see HPL-96-61-1996 [B1].

Table 1—Physical units data type structure

Field	Description	Data type	Number of octets
1	Physical units interpretation—see Table 2	U8E	1
2	$(2 * \langle \text{exponent of radians} \rangle) + 128$	U8C	1
3	$(2 * \langle \text{exponent of steradians} \rangle) + 128$	U8C	1
4	$(2 * \langle \text{exponent of meters} \rangle) + 128$	U8C	1
5	$(2 * \langle \text{exponent of kilograms} \rangle) + 128$	U8C	1
6	$(2 * \langle \text{exponent of seconds} \rangle) + 128$	U8C	1
7	$(2 * \langle \text{exponent of amperes} \rangle) + 128$	U8C	1
8	$(2 * \langle \text{exponent of kelvins} \rangle) + 128$	U8C	1
9	$(2 * \langle \text{exponent of moles} \rangle) + 128$	U8C	1
10	$(2 * \langle \text{exponent of candelas} \rangle) + 128$	U8C	1

Table 2—Physical units interpretation

Value	Manifest constant	Definition
0	PUI_SI_UNITS	Unit is described by the product of SI base units, plus radians and steradians, raised to the powers recorded in fields 2 through 10. Units for some quantities, such as the number of people through a turnstile, cannot be represented using these units. Enumeration zero, with fields 2–10 set to 128, is the appropriate enumeration for these cases when a quantity is being defined.
1	PUI_RATIO_SI_UNITS	Unit is U/U, where U is described by the product of SI base units, plus radians and steradians, raised to the powers recorded in fields 2 through 10.
2	PUI_LOG10_SI_UNITS	Unit is $\log_{10}(U)$, where U is described by the product of SI base units, plus radians and steradians, raised to the powers recorded in fields 2 through 10.
3	PUI_LOG10_RATIO_SI_UNITS	Unit is $\log_{10}(U/U)$, where U is described by the product of SI base units, plus radians and steradians, raised to the powers recorded in fields 2 through 10.
4	PUI_DIGITAL_DATA	The associated quantity is digital data (e.g., a bit vector) and has no unit. Fields 2–10 shall be set to 128. The <i>digital data</i> type applies to data that does not represent a quantity, such as the current positions of a bank of switches.
5	PUI_ARBITRARY	The associated physical quantity is represented by values on an arbitrary scale (e.g., hardness). Fields 2–10 are reserved, and shall be set to 128.
6–255	—	Reserved

The U/U forms (enumerations one and three) are for expressing *dimensionless* units such as strain (meters per meter) and concentration (moles per mole). The numerator and denominator units are identical, each being specified by subfields two through ten.

Boolean data (values in {0, 1} or {False, True}) shall be represented as digital data (enumeration 4).

Annex H contains examples of the physical units.

4.12 Universal unique identification

Symbol: UUID

Size: 10 octets

The UUID is an identification field associated with the TBIM whose value is unique on the planet. The UUID field shall be 10 octets long and consists of four sub-fields (from most to least significant order: location, manufacturer's, year, and time) defined in Table 3.

There shall be no requirement that the interpretation of the UUID reflect the actual place or time of manufacture of the TBIM. The use of time and location in the algorithm shall be used only to ensure uniqueness. Thus a manufacturer shall be allowed to use the fields defined by the algorithm as desired provided that when interpreted according to Table 3, there shall be no other possible manufacturer which could claim use of the same UUID.

Table 3—Universal unique identification data type structure

Field	Description	Number of bits
1	Location field: The value of this field shall be chosen by the TBIM manufacturer to identify a particular location on the earth, the <i>location</i>, over which the manufacturer has physical control. This value may represent the actual location of TBIM manufacture. A manufacturer may use different values of this field in his operation as long as all values meet the requirements of this subclause. The representation of the location field shall be 42 bits. The msb shall indicate North (asserted) or South (not asserted) latitude. The next 20 most significant bits of this field represent the magnitude of the <i>location</i> latitude as an integer number of arc seconds. The next most significant bit shall indicate East (asserted) or West (not asserted) longitude. The remaining 20 bits shall represent the magnitude of the <i>location</i> longitude as an integer number of arc seconds. Latitude magnitude values greater than 90 degrees are reserved. Longitude magnitude values of greater than 180 degrees are reserved. NOTE—One arc second at the equator is about 30 meters. Thus, the range represented by each 20-bit field is 0 to 1 048 575 arc seconds or 0 to 291 degrees, which is sufficient to represent latitude and longitude on the earth's surface.	42
2	Manufacturer's field: The value of this field may be chosen by the TBIM manufacturer for any purpose provided there is no interference in the use of the location field. Location field interference occurs when there is more than one manufacturer that could claim physical control over the location defined by the location field. If such interference exists, all manufacturers affected shall negotiate the use of the manufacturer's field values to resolve any interference. That is, the combination of the location field and the manufacturer's fields shall unambiguously define a specific TBIM manufacturer. This negotiation shall be reopened every time there is a change in the interference.	4
3	Year field: The value of this field shall be the value of the current year. The representation of the year field shall be a 12-bit integer value. The range of this field shall be the years 0 to 4095 AD. The beginning of the year shall be deemed to start on January 1, 00:00:00, UTC.	12
4	Time field: This value shall be chosen by the TBIM manufacturer such that in combination with the location, manufacturer's and year fields the resulting UUID is unique for all TBIMs made under the manufacturer's control. The choice of values for the time field shall be further restricted such that the values when interpreted as the time since the beginning of the year do not represent times prior to the manufacturer obtaining physical control over the <i>location</i> nor values in the future in which the manufacturer cannot guarantee physical control over the <i>location</i>. The representation of the time field shall be a 22-bit integer. The range shall be 0 to 4 194 303. When it is necessary to interpret this field as time since the beginning of the year, the representation shall be an integer number of ten-second intervals. In this case, time values greater than one year^a are reserved. The beginning of the year shall be deemed to start on January 1, 00:00:00, UTC.	22

^aThere are approximately 31 536 000 seconds per year.

4.13 Arbitrary octet array

Symbol: OctetArray

Size: varies

This data type comprises an arbitrary number of octets, treated as an aggregate entity that may or may not be interpreted as a number. An OctetArray can be a structure composed of one or more primitive data types, arrays of primitive data types, or smaller OctetArrays.

4.14 Time-of-day

Symbol: ToD

Size: 8 octets

The format of the ToD structure is the same as defined in IEEE Std 1588-2002. Conversion of this format into other possible formats is described in Annex B of IEEE Std 1588-2002. This structure consists of two numeric fields described in Table 4.

NOTE—This format is corrected for the flaw in negative time representation discovered in IEEE Std 1588-2002.

Table 4—Time-of-day data type structure

Field	Description	Data type	Number of octets
1	Seconds —This is an unsigned 32-bit number that represents the number of seconds from 0 hours on 1 January 1970.	U32C	4
2	Sign, Nanoseconds —This is an unsigned 32-bit integer that is composed of two smaller fields. The most significant bit will be interpreted as the sign bit of the entire ToD value. The least significant 31 bits represents the number of nanoseconds to be added to the value identified by the Seconds field before the sign is applied. The value specified in the Nanoseconds field is constrained to the domain 0 to 999 999 999, inclusive.	U32C	4

5. Smart transducer functional specification

As shown in Figure 1, a system shall consist of one TBC, one or more TBIMs, one or more Data Communications Channels and an optional synchronization signal.

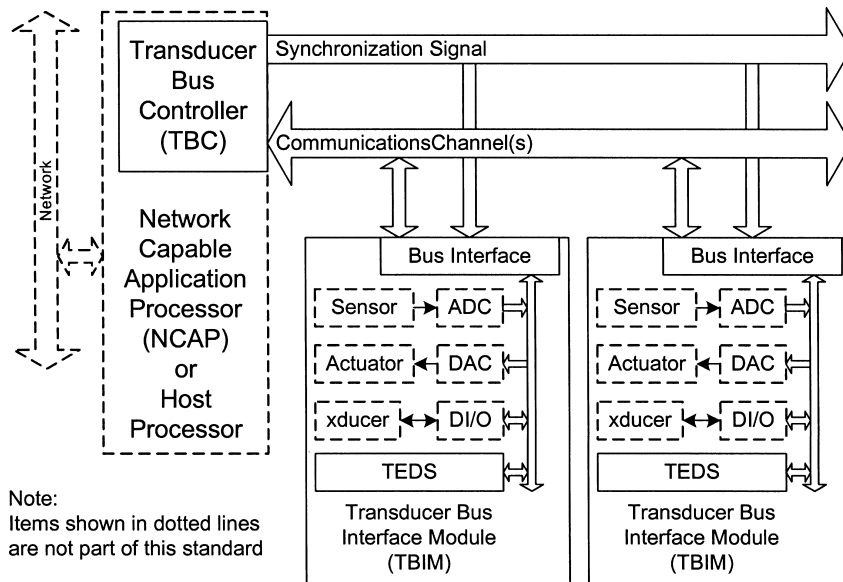


Figure 1—Functional context for the smart transducer interface

5.1 Plug and play capability

A TBIM(s) and TBC that are built to this standard must be able to be connected using a transducer bus and be able to operate without any changes to the system software. There shall be no need for different drivers, profiles, or other system software changes to provide basic operation of the transducer. Applications that use, produce, or display the data associated with a transducer are not included in this statement. Features that go beyond the requirements of this standard are allowed but will impact the plug and play capability of the TBIM or TBC.

The fact that connectors and bus power supply voltages are recommended by this standard and not required makes it necessary for the user to determine if a particular TBIM or TBC can be safely connected to a particular implementation of the transducer bus.

5.2 Addresses

Sixteen-bit addresses are used on the transducer bus. Concatenating the TBIM Alias and the TransducerChannel number forms the 16-bit address. The TBIM Alias is assigned by the system as part of the discovery process and forms the eight most significant bits of the address. The manufacturer of the TBIM permanently assigns each TransducerChannel number. The TransducerChannel number is the eight least significant bits of the address.

Table 5—Classes of addresses

Address class	TBIM Alias	TransducerChannel number
Global	Zero	Zero
AddressGroup	Zero	Non-zero
TBIM	Non-zero	Zero
TransducerChannel	Non-zero	Non-zero
TBC address	255	Zero

There are four classes of addresses and one special address as described in Table 5. Each row in the table defines a name for a class of addresses and that name is used throughout the standard.

Not all commands may be honored by all address classes. The address class that honors a particular command is defined in 6.1 along with the definition of the commands.

Commands addressed to the Global address shall be received and honored by all TBIMs on the bus.

Commands issued to an AddressGroup shall be honored by all TransducerChannels that have been initialized as members of that AddressGroup, irrespective of the TBIM on which they reside. There are eight AddressGroups available for assignment. Each AddressGroup is represented by one bit in the TransducerChannel number field of the address class. Using bitwise OR logic, one command may be issued to multiple AddressGroups simultaneously.

Commands issued to a TBIM apply to the TBIM as a unit. If a command is addressed to a TBIM and if it is legal to address that command to a TransducerChannel, that command shall apply to all TransducerChannels that reside on that TBIM.

Commands issued to a specific TransducerChannel apply to that TransducerChannel only.

There are no commands defined that shall be honored when addressed to the TBC address. Replies to commands and command acknowledgments are sent to this address.

5.3 Common characteristics

The following subclauses define the operating characteristics that are common to all TransducerChannel types.

5.3.1 Operating states

A high-level state transition diagram is provided as Figure 2. There are three basic operating states for a TransducerChannel after it comes out of initialization. The TransducerChannel shall enter the *halted* state when it comes out of initialization. A TransducerChannel accepts most commands when it is in the halted state. A TransducerChannel enters the operating state upon receipt of an enable TransducerChannel command and remains in this state until the **Halt** command is received. The TransducerChannel does not accept certain commands when in this state. A TransducerChannel is placed in a *Sleep* state by the **Sleep** command when it is in the halted state. The only command that a TransducerChannel shall accept when in the sleep state is the **Wake-up** command that returns it to the halted state. For more details about what commands may be accepted by a TransducerChannel in each state, see the detailed description of each command in Clause 6.

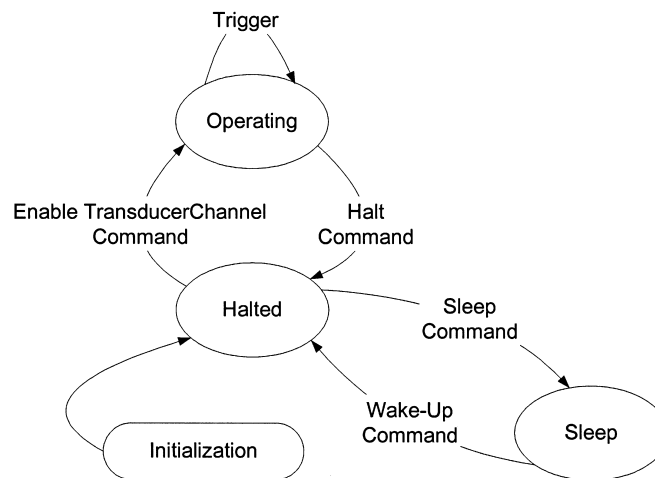


Figure 2—TransducerChannel operating states

5.3.2 Enabling and disabling triggers

A TransducerChannel may have its ability to be triggered enabled or disabled by means of commands. Unless otherwise noted, all operations described in this document are for TransducerChannels with triggers enabled.

5.3.3 Diagnostics

One prime reason for using Smart Transducers is to be able to have devices that can perform self-check or diagnostics. The standard provides a mechanism to engage diagnostics, but there are no requirements imposed on the scope of the diagnostic logic.

5.3.4 Structures used to store and transmit data

There are three structures that are used to store and transmit data in this standard. They are the data set, the message, and the packet. The following subclauses describe each of these structures. The Applications above the protocol stack deal with data sets. The upper layers of the protocol stack deal with messages. If a data set contains more octets than can be sent with a single message, it shall be broken up by the application into multiple messages.

5.3.4.1 Data sets

All TransducerChannels operate with *data sets*. Three fields within the TransducerChannel TEDS define a data set. The *Maximum data repetitions* field (see 7.4.2.12) defines the maximum number of individual data samples in a data set. The actual number of samples may be made lower than the number in the *Maximum data repetitions* field by the optional **Set TransducerChannel Data Repetition Count** command. When reset or powered up, the unit uses the default value. If the data repetition count is programmable, the default value shall be one. If the data repetition count is not programmable, the default value shall be the same as the number in the *Maximum data repetitions* field of the TEDS. The second field is the *Series increment* field (see 7.4.2.14). This field is used to determine the interval between samples and it may be overridden by a manufacturer-defined command or an embedded actuator. The third field is the *Series units* field (see 7.4.2.15). This field defines the units for the *Series increment* field. The implication of the *Series units* field is that the units of the *Series increment* field does not need to be time and that if this is the case the time interval between samples may not be uniform.

EXAMPLE—The *Series units* are degrees Kelvin and the *Series increment* is 0.5. This combination would cause a sensor to acquire data every 0.5 degrees. The samples would be at uniform temperature intervals instead of uniform time intervals.

5.3.4.2 Messages and packets

Messages may contain up to 65 520 octets plus 14 octets in headers. However, the data link and physical layers of the protocol stack transmit packets. The maximum packet length is defined in Clause 10. If a message is too long to fit within a single packet, it is the responsibility of the Data Link Layer in the protocol stack to break messages down into multiple packets for transmission.

5.4 TransducerChannel type descriptions

This subclause specifies the general behavior of three TransducerChannel types. Attributes in the TransducerChannel TEDS describe the capabilities of the various TransducerChannels to the system. The detailed timing and control of these TransducerChannel types are specified in subclauses 5.7 through 5.14 and Clauses 6 and 7 of this standard. The TransducerChannel types are:

- Sensor
- Event sensor
- Actuator

The term TransducerChannel as used in this subclause refers to the physical transducer and all electronics in the path that connects the transducer to the communications functions.

5.4.1 Sensor

A sensor shall measure some physical parameter on demand and return digital data representing that parameter. On the receipt of a trigger the sensor shall start the collection and storing of a data set

within the TBIM. The timing of the individual samples in the data set shall be controlled by the TBIM and is a function of the operating mode of the sensor. A sensor, in the operating state, shall respond to a read command by returning the appropriate data set. If a new data set is not available, the TransducerChannel shall respond with the same data that was returned on the previous read command with a flag that indicates that the data was previously read.

5.4.2 Event sensor

An event sensor differs from a sensor in that it does not determine the level of some physical phenomena but determines when a change of state has occurred. This change of state may be an analog signal crossing some threshold or set of discrete bits that match or fail to match a given bit pattern. Thus the output of an event sensor indicates the state of its input. The two allowable states are one and zero. There are two pieces of information that may be determined from the output of an event sensor. One is the current state of the input and the other is the time that a change in state occurred.

The TEDS definition for an event sensor is the same as for any other transducer.

5.4.2.1 Event sensor output

The data model of an event sensor output is defined in the TransducerChannel TEDS in the same way that the output of any other sensor is defined. However, the magnitude of the output shall be either zero or one.

5.4.2.2 Event time

The event sensor itself only reports the fact that an event has occurred. Other elements in the system must determine when the event occurred. There are multiple ways that this may be accomplished and just which method is chosen depends upon the requirements of the application. If it is acceptable to know the time within a few milliseconds, the event sensor may issue a trigger acknowledgment message and the TBC can determine when the event occurred by the time of arrival of the message. The physical layer is a major factor in determining how much lag there is between the event and the TBC recognition of the event. If a physical layer is used with multiple CommunicationsChannels operating in parallel, the time can be very short. However, if a packet based physical layer is used then there is more delay between the occurrence of the event and the TBC becoming aware of it. On the other end of the spectrum the time of occurrence of the event may be determined to within a microsecond or less by the use of an embedded time interval sensor or a time-of-day sensor in the TBIM with the event sensor. See Annex J.2 and Annex J.3 for a discussion of time interval and time-of-day sensors.

5.4.2.3 Analog event sensors

Analog event sensors have an input signal that is in an analog form. An event is determined when the input crosses a threshold. For an analog event sensor, upper and lower thresholds are defined. The lower threshold differs from the upper by a quantity called *hysteresis*. The hysteresis value shall be zero or greater. A rising transition (i.e. the condition of the input that causes the output state to transition from zero to one) shall be when the analog input value passes first through the lower and then through the upper threshold. A falling transition (i.e., the condition of the input that causes the output state to transition from one to zero) shall be when the analog input value passes first through the upper and then through the lower threshold. Figure 3 shows this behavior graphically. A change of the output state shall occur only after the input has passed through both thresholds. Changes of the output state shall only be reported after the device has been triggered. The upper threshold and the hysteresis may be fixed at the time of manufacturing the event sensor or they may be made programmable. The recommended method of making them programmable is to use embedded actuators to set the upper threshold and hysteresis.

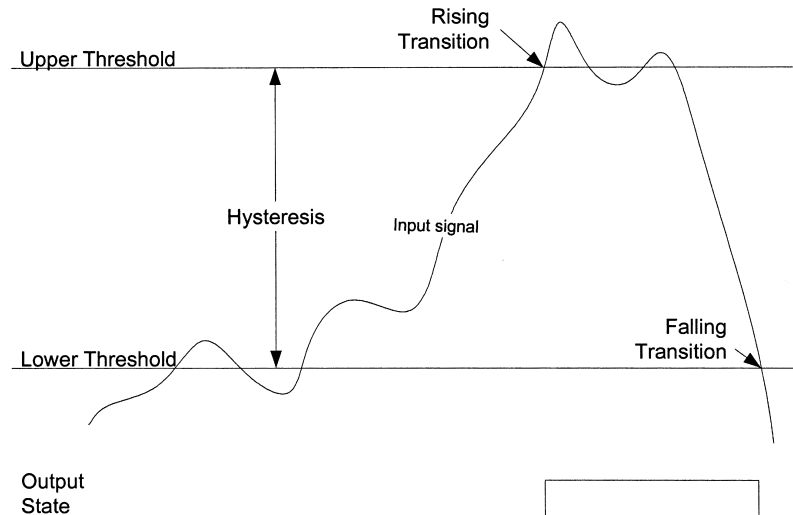


Figure 3—Analog event sensor behavior

5.4.2.4 Digital event sensors

Digital event sensors have one or more discrete input signals. An event is determined when the input matches a defined pattern for a manufacturer-defined fixed period of time. A rising transition shall be when the inputs match a specific digital pattern without change for a fixed period of time. The period of time should be determined by the manufacturer to match the characteristics of the event but may be controlled by manufacturer unique commands or an embedded actuator. A falling transition shall be when the inputs cease to match the pattern for some fixed time.

A digital event sensor may have several associated embedded *TransducerChannels*. This may include a sensor(s) to return the digital input on command and actuators to set up a complex event sensor. Embedded actuators may be used to define patterns, masks, time delays and combinations of several discrete inputs that are combined into one event. The *TransducerChannel TEDS* for these embedded *TransducerChannels* shall be appropriate to their function. The use and classification of embedded *TransducerChannels* is defined in 7.3.2.8.

5.4.2.5 Transitions reported

An event sensor may be designed to signal an event on falling transitions or rising transitions, or both. The **Set operating mode** command (see 6.1.7) allows the system to select which transitions to report as events. The edge-to-report capabilities of the event sensor along with the default condition are defined in the *TransducerChannel TEDS* (see 7.4.2.32). If the event sensor is in the operating state when an event occurs, an event sensor shall set the data/event bit in the status register and issue a trigger acknowledgment message if the trigger command requested an acknowledgment. If no acknowledgment was requested, the *TransducerChannel data/event* bit in the status register shall be set but no message shall be transmitted.

Whether or not an event has occurred since the last status read may be determined by examining the *TransducerChannel data/event* status bit as described in 5.12.6.

5.4.2.6 Event sensor status

An event sensor shall set the *TransducerChannel data/event* status bit whenever an event is detected in either the operating or halted states. If the event sensor is in the operating state but has not been triggered, it shall set the *TransducerChannel missed data or event* status bit coincident with any event

that occurs. The *TransducerChannel missed data or event* status bit shall not be set if the event sensor is in the halted state.

Consistency checks may be made by the TBIM if the bit pattern is changeable for a digital event sensor, or either the upper threshold or hysteresis is changeable for an analog event sensor. Inconsistent values are signaled by setting the *hardware error* status bit (see 5.12.7). If inconsistent values are checked, the check shall be made immediately following a change in any of these parameters. The check shall consist of verifying the following relationships:

For analog values:

Maximum sensed input > upper threshold $\geq$ (upper threshold – hysteresis) > minimum sensed input value

For digital patterns, the pattern is a legal input.

NOTE—Consistency checking by the TBIM shall only use data in the data model specified by the TransducerChannel TEDS of the embedded TransducerChannels.

5.4.2.7 Streaming event sensors

Like sensors and actuators, event sensors may be operated in the streaming mode. In the streaming mode (see 5.8.4), an event sensor in the operating state shall transmit a message each time an event occurs without subsequent triggers. In the non-streaming mode, an event sensor only reports the first event after the trigger.

5.4.2.8 Time between events

The minimum time between events that an event sensor can detect is limited by the design of the event sensor. The minimum time between events that can be detected by the event sensor shall be specified in the TransducerChannel TEDS field, *TransducerChannel sampling period*, 7.4.2.20.

5.4.3 Actuator

An actuator shall cause a physical or embedded output action to occur. The actuator output state changes to match the appropriate data set when a triggering event occurs. If there is more than one data point in a data set, the interval between issuing the individual data points shall be under the control of the TBIM.

An actuator may be built which does not require a data set be written to it before it performs an action. This type of device always uses a default data set or no data at all and performs a predefined action upon the receipt of a trigger.

5.5 Embedded TransducerChannels

Embedded TransducerChannels are TransducerChannels whose functionality is completely contained within a TBIM. An embedded TransducerChannel does not sense or control any function outside of the TBIM. For example, embedded actuators may be used to set the threshold and hysteresis of an event sensor. An embedded digital event sensor may be set up to detect and report any change in the status within a TBIM. Embedded TransducerChannels appear to the system as normal TransducerChannels. They respond to commands, have TEDS and are counted when determining the number of TransducerChannels in a TBIM.

NOTE—The advantage of using “embedded” TransducerChannels over control commands is that “embedded” TransducerChannels are supported by TEDS, which gives the overall system a standard way to obtain the details of operation of the TBIM. They should be used when it is difficult to specify the resolution of a control command in a standard way.

5.6 TransducerChannel groups

There are two types of TransducerChannel groups defined in this standard. They are ControlGroups and VectorGroups. They are similar in implementation but are used for two different functions.

5.6.1 ControlGroups

ControlGroups are used to define collections of TransducerChannels when one TransducerChannel is the primary channel and the remaining TransducerChannels in the group provide either additional information about the primary TransducerChannel or are used to control some aspect of the primary TransducerChannel. For example, a ControlGroup can be used to define up to three additional TransducerChannels associated with an analog event sensor. One is a sensor that is used to measure the analog input to the event sensor. The second is an actuator that is used to set the threshold for the event sensor. The third is an actuator that can be used to set the hysteresis for the event sensor.

5.6.2 VectorGroups

VectorGroups are used to define relationships between TransducerChannels within a single multi-channel TBIM that imply a display or mathematical relationship between the TransducerChannels. For example, they can be used to identify the relationships between the components of a three-axis accelerometer. VectorGroups are used by the software in the TBC/NCAP or host processor to group the outputs of the individual TransducerChannels into vectors for display or computational purposes.

NOTE—VectorGroups are in some ways similar to TransducerChannel proxies. TransducerChannel proxies are used to identify TransducerChannels that are grouped together for reasons such as efficient transmission of the data and/or simultaneous triggering. Proxies may or may not represent a vector for display or computational purposes. However, it is recommended that all vectors be implemented as proxies, especially in spatial vector applications (like velocity or acceleration), where the measurements of the three components at different points in time could lead to misinterpretation.

5.7 TransducerChannel proxy

A TransducerChannel proxy is an artificial construct used to combine the outputs of multiple sensors or the input to multiple actuators into a single structure. A TransducerChannel proxy has a TransducerChannel number and may be read or written but it does not have the other characteristics of a TransducerChannel. This means that a proxy does not have a TransducerChannel TEDS, a Calibration TEDS, a Transfer Function TEDS or a Frequency Response TEDS. The number of TransducerChannel proxies that exist in a TBIM are defined in the Meta-TEDS (see 7.3.2.12).

Like any other transducer, a TransducerChannel proxy is permanently assigned a TransducerChannel number by the manufacturer. Thus, a proxy may be addressed in a manner consistent with all other TransducerChannels implemented on the TBIM.

A proxy shall not represent a collection of incompatible transducer types. In other words, a proxy represents a group of sensors or a group of actuators but shall not represent a group containing both transducer types.

Reading or writing the data from the individual members of a proxy may be allowed or disallowed at the manufacturers discretion. If the manufacturer chooses to disallow reading or writing the individual members of a proxy, then the receipt of one of those commands shall set the *TBIM command rejected* bit in the status word.

There are two methods that may be used to combine the data sets for the proxy. They are the *block* and *interleave* methods. These two methods are shown graphically in Figure 4. The block method

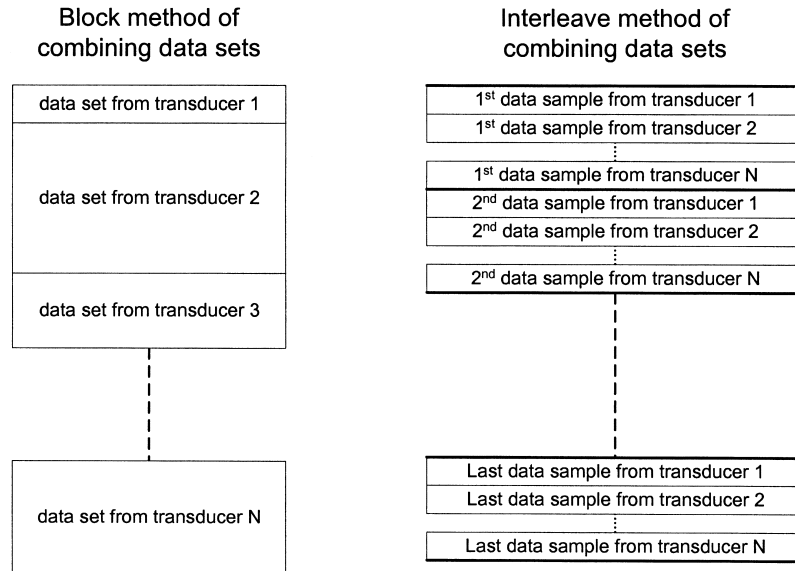


Figure 4—Methods of combining data sets

allows the data sets to be of different lengths. With the interleave method all data sets must have the same size.

5.8 Attributes and operating modes

The following subclauses define the operating modes that are described by the attributes found in the TransducerChannel TEDS. Each attribute indicates whether or not a TransducerChannel supports the associated operating mode. Some of these modes of operation are mutually exclusive and others may be enabled at the same time.

5.8.1 Sampling modes

A sensor or actuator may be operated in one of two sampling modes. The sampling modes have different relationships between the trigger and the sampling of the data by a sensor or the application of a sample to the output of an actuator. The mode or modes that a TransducerChannel is capable of operating in is defined by the attributes given in the TransducerChannel TEDS. For the possible values of these attributes, see 7.4.2.28.

The trigger-initiated and free-running sampling modes are mutually exclusive. The TransducerChannel must be operated in one of these two modes.

5.8.1.1 Trigger-initiated

In the trigger-initiated sampling mode, the TransducerChannel shall begin acquiring or outputting a data set immediately upon receipt of a trigger. Sample processing continues until all samples in the data set are processed at a rate determined by the TBIM and then stops.

5.8.1.2 Free-running without pre-trigger

In the free-running sampling mode, a sensor is measuring some physical parameter autonomously and continuously when in the operating state. The data being acquired and converted is discarded until a

trigger is received. Once a trigger has been received, the next sample converted is stored in the TBIM as the first word in the data set. Consecutive samples are stored in the TBIM until the entire data set is completed. At this point, the TransducerChannel returns to discarding samples until the next trigger is received. The sampling operation of a TransducerChannel when in the halted state is at the discretion of the manufacturer.

An actuator operating in the free-running sampling mode shall apply the data set received before the trigger in accordance with its End-of-data-set operating mode.

5.8.1.3 Free-running with pre-trigger

The following two subclauses describe the operation of a sensor when operating in the free-running mode with or without buffers enabled.

An actuator may not be operated in the free running with pre-trigger mode.

5.8.1.3.1 Free-running with pre-trigger without buffers enabled

In the free-running sampling mode, a sensor is measuring some physical parameter autonomously and continuously when in the operating state. The data being acquired and converted is stored until a trigger is received or the pre-trigger count (see 6.1.1.5) is reached. After the number of samples that have been stored reaches the pre-trigger count, the next sample acquired shall cause the oldest sample to be discarded and the new sample stored. After a trigger has been received, the next sample converted is stored in the TBIM as the next word in the data set. Consecutive samples are stored in the TBIM until the entire data set is completed. The data set shall be complete when the number of samples equal to the *data set size* minus the *pre-trigger count* has been acquired after the trigger is received. At this point the TransducerChannel returns to discarding samples until the next trigger is received or the data set is read. After the data set is read, the TransducerChannel shall start storing samples again while waiting for another trigger. In this mode, the data set may only be read once. Subsequent reads before the next trigger will return zero octets.

The sampling operation of a TransducerChannel when in the halted state is at the discretion of the manufacturer.

5.8.1.3.2 Free-running with pre-trigger and buffers enabled

In the free-running sampling mode, a sensor is measuring some physical parameter autonomously and continuously when in the operating state. The data being acquired and converted is stored until a trigger is received or the pre-trigger count (see 6.1.1.5) is reached. After the number of samples that have been stored reaches the pre-trigger count, the next sample acquired shall cause the oldest sample to be discarded and the new sample stored. Once a trigger has been received, the next sample converted is stored in the TBIM as the next word in the data set. Consecutive samples are stored in the TBIM until the entire data set is completed. The data set shall be complete when the number of samples equal to the data set size minus the pre-trigger count has been acquired after the trigger is received. The TransducerChannel shall switch to the next empty buffer and start acquiring data samples for the next data set. If no empty buffers are available, it shall discard any samples acquired until a buffer becomes available. A buffer shall be considered available after it has been read. The sampling operation of a TransducerChannel when in the halted state is at the discretion of the manufacturer.

5.8.2 Buffered operation mode

A sensor or actuator may be capable of being operated in buffered or non-buffered modes. In the buffered mode, one buffer is available to be read from a sensor or applied to an actuator output while other buffers are available to be filled. A characteristic of a TransducerChannel operating in the buffered mode is that the data available to be read or applied is always the data that was available in a buffer before the previous trigger. In the non-buffered mode, only a single buffer is available to store a

data set. The buffering capabilities of a given TransducerChannel are described by the *Buffered attribute* in the TransducerChannel TEDS. See 7.4.2.29 for the details of this attribute.

A sensor that is in the halted state and before it is placed into the operating state for the first time after initialization shall return zero data octets to any read command. If a buffered sensor has buffers that are full or partially full when it is returned to the halted state, it shall return the contents of the oldest buffer on receipt of a read command. Upon receipt of subsequent read commands, the contents of the remaining buffers shall be returned. After the contents of each buffer that had unread data in it when the **Halt** command was received have been read, the TransducerChannel or TransducerChannel proxy shall return zero data octets to subsequent read commands.

All buffers shall be cleared when the TransducerChannel transitions from the halted to the operating state.

5.8.3 End-of-data-set operation mode

This mode only applies to actuators. The *End-of-data-set operation attribute* in the TransducerChannel TEDS (7.4.2.30) defines the possible operations that an actuator may perform when it reaches the end of a data set. There are two possible operating modes described in the subsequent subclauses.

NOTE—These operation modes describe ways to allow actuators to smoothly transition from one data set to another. While this is desirable in most cases, there are instances when a rapid transition to another condition is required, i.e., emergency shutdown. For these cases, a second actuator TransducerChannel in the same TBIM may be used which executes the transition to the new condition. The mechanism for switching control of the output from one actuator to the other is internal to the TBIM and not covered in the standard.

5.8.3.1 Hold

An actuator operating in the hold mode shall utilize all of the samples in a data set and then continue to utilize the last sample in the data set until a new trigger is received.

NOTE—The hold mode is suited to an actuator containing a valve or other mechanical positioner. This type seems to also fit bistable-output logic-type output devices but is also frequently associated with the trigger-initiated sampling mode.

5.8.3.2 Recirculate

An actuator operating in the recirculate mode shall apply all of the samples in a data set to the output, then return to the beginning of the data set and repeat the application of that data set until another trigger is received or the TransducerChannel is disabled. When returning to the beginning of a data set or to a new data set, the appropriate sample interval shall be maintained. When another trigger is received, the TransducerChannel shall switch to the new data set when it reaches the end of the current data set.

5.8.4 Streaming mode

In the streaming mode, a sensor shall be armed to acquire data when the trigger is received. An actuator operating in the streaming mode shall be armed to apply data when a trigger is received. Sampling will begin at the appropriate time based upon the synchronization signal. When in this mode, a sensor shall internally stimulate its trigger logic at the appropriate time to cause the sensor to continue to acquire samples with a uniform sample interval. An actuator being operated in the streaming mode shall simulate a trigger when it receives a write data command.

After the first trigger, a sensor shall transmit a new data set as soon as it is acquired. This operation continues, without subsequent triggers, until the sensor is put into the halted state.

An actuator operating in the streaming mode shall begin applying the data written to it at the appropriate time after it has been received as determined by the sampling mode (trigger-initiated or free-running). An actuator operating in the buffered mode shall immediately switch to the next filled buffer after applying all of the data in the current buffer. If another filled buffer is not available, the actuator shall take the action prescribed by the setting of the End-of-data-set operations attribute as described in 5.8.3. If this action is to *recirculate*, the unit shall not look for a new filled buffer until it completes reapplying the current buffer. If that action is to *hold*, the unit shall switch to the new buffer as soon as it has been received and stored in memory.

When operated in the streaming mode, the system has no way of knowing whether or not all messages from a given TBIM have been received correctly. To correct this condition, the TBC shall periodically transmit a sequence number either to the TBIM or globally. This sequence number shall then be returned by the TBIM in the next message that it transmits. If that sequence number has already been used when transmitting a message from this TransducerChannel, the TBIM shall increment the sequence number and set the Predicted Sequence Number flag before transmitting the message. If transmitting a number of messages from different TransducerChannels within a TBIM, the same sequence number shall be used since one of its purposes is to identify the message with a particular time interval. For systems where the TBC does not periodically transmit a sequence number, the TBIM shall generate a Predicted Sequence Number locally and use that to provide an identifier for messages being transmitted across the bus.

A TBIM that may be operated in the streaming mode shall contain a 16-bit register that is used to store the sequence number. This register is set by management commands in the Data Link Layer or a lower layer in the communications protocol stack.

5.8.5 Edge-to-report operating mode

This operating mode applies to event sensors only.

An event sensor may be capable of operating in one of three modes. In the falling transitions mode, it reports all falling transitions. In the rising transition mode, it reports all rising transitions. See 5.4.2.3 and 5.4.2.4 for the definition of falling and rising transitions. In the all-transitions mode, it reports both rising and falling transitions. The *Edge-to-report attribute* field in the TransducerChannel TEDS defines the capabilities of the event sensor with regard to this operating mode. When defaults are included, there are six possible values to this attribute. See 7.4.2.32 for details.

5.8.6 Actuator-halt mode

This attribute applies to actuators only. This mode defines what an actuator does when a **Halt** command is received. The *Actuator-halt attribute* in the TransducerChannel TEDS (see 7.4.2.33) defines the possible operations.

5.8.6.1 Halt immediate

The actuator shall hold the current state of its output until it is returned to the operating state or put into the sleep state.

5.8.6.2 Halt at the end of the data set

The actuator shall finish applying the current data set and then hold the last output value of its output until it is returned to the operating state or put into the sleep state.

5.8.6.3 Ramp to a predefined state

The actuator shall use a manufacturer-defined process to ramp the output to a predefined state.

5.9 Triggering

A trigger is a signal applied to a TransducerChannel or set of TransducerChannels to cause them to take a particular action.

5.9.1 Trigger methods

There are two methods recognized by this standard to initiate a trigger. They are explicit triggers commanded by the TBC and events within a TBIM that may be used as triggers.

5.9.1.1 Trigger messages

Trigger messages are sent from the TBC to one or more TransducerChannels on a transducer bus. A trigger message is defined by the Trigger Protocol in 8.5. Trigger messages may require acknowledgement when addressed to an event sensor. Any other use of the trigger acknowledgement capability in the trigger protocol is not covered by this standard. A trigger message may be addressed to any of the following:

- A TransducerChannel
- A TransducerChannel Proxy
- A TBIM
- An AddressGroup
- The global address

A trigger message addressed to a specific TransducerChannel applies to one TransducerChannel on one TBIM.

A TransducerChannel proxy is an addressable resource within a single TBIM that is capable of *representing* one or more TransducerChannels within that TBIM. A trigger command addressed to a TransducerChannel proxy triggers each TransducerChannel represented by that TransducerChannel proxy.

A trigger message addressed to a TBIM triggers all TransducerChannels that are implemented on that TBIM.

The system user may define AddressGroups. When the system is set up, each TransducerChannel to be included in an AddressGroup is programmed to respond to the AddressGroup identifier for that AddressGroup. A trigger message issued to that AddressGroup triggers all members of that AddressGroup.

A global trigger applies to all TransducerChannels in all TBIMs on a given transducer bus. The system issues a global trigger by issuing a trigger message to the global address.

Regardless of the address mode, a TransducerChannel shall only honor the trigger when triggering is enabled for that TransducerChannel.

5.9.1.2 Events used as triggers

Events within a TBIM may be used as triggers. Events of this nature are not Trigger Messages, and they shall not be acknowledged via the Trigger Protocol. An event used as a trigger may be formally implemented as an Event Sensor.

NOTE—For event sensors used as triggers in systems where it is important for the system to be able to determine the time of the event, it is necessary to augment the event sensor with a time interval sensor or time-of-day sensor to be able to determine the time of the event accurately.

5.9.2 Nominal trigger logic

A functional block diagram of the logic within a TBIM that is related to triggering is shown in Figure 5.

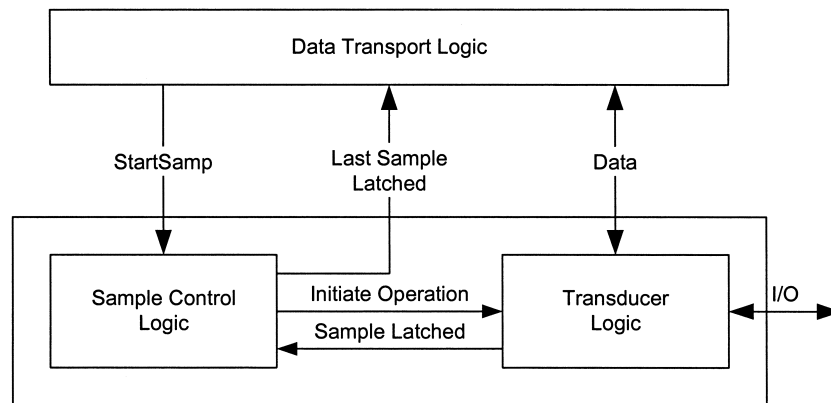


Figure 5—Simple TransducerChannel functional blocks

The Data Transport Logic responds to trigger messages. Upon receipt of a trigger message, provided that the TransducerChannel trigger is enabled, the Data Transport Logic issues a StartSamp signal.

The StartSamp signal shall cause a traditional sensor to acquire a new data set. For the simplest sensors, this may mean a digital latch is updated or an analog-to-digital converter begins a conversion. More complex sensors may take multiple samples per trigger, spaced apart incrementally in time or some other dimension.

The StartSamp signal shall cause an event sensor to begin monitoring the event of interest. The occurrence of the event shall cause the Sample Control logic to generate the Last Sample Latched signal.

The StartSamp signal shall cause an actuator TransducerChannel to apply a data set to its output. For the simplest actuators, this may mean a digital latch is updated or a digital-to-analog converter begins a conversion. More complex actuators may apply multiple samples per trigger, spaced apart incrementally in time or some other dimension.

The Sample Control logic is responsible for orchestrating the sequence of operations needed to acquire or apply a data set. The Transducer Logic block represents all signal conditioning, data conversion, and buffering logic. The Initiate Operation signal is used to initiate the process of data conversion on a single sample, and the Sample Latched signal indicates that a sample has been acquired or output.

For TransducerChannels implemented under the Nominal Triggering model, the determination of the time at which the sample was latched becomes the sole responsibility of the NCAP or host processor based on information provided by the TBIM. There are two variations. The method used shall be specified in the *Source for the time of sample* field in the TransducerChannel TEDS (see 7.4.2.24).

This time calculation gives the time of the last sample in the data set. The following calculation describes how to calculate the time of any other sample in the data set.

N is the number of samples in the data set

t_{si} is the sample interval

$$T_i = T_N - (N - i) * t_{si} \quad \text{for } i = 1 \text{ to } N$$

When the enumerated value of *Source for the time of sample* indicates that the delay between application of trigger and the subsequent latching of the sample is constant, and when the TBIM lacks embedded hardware resources to support time tagging, the time of sample shall be computed from the time at which the trigger message was issued by adding an adjustment equal to the *Incoming propagation delay through the data transport logic* field of the TransducerChannel TEDS. This method neglects the effects of propagation delays on the transducer bus.

Calculation:

T_{tm} is the time of day at which the trigger message was sent by the TBC.

t_{pdi} is the value of the *Incoming propagation delay through the data transport logic* field found in the TransducerChannel TEDS.

T_1 is the time of day when the first sample in the data set was latched.

$$T_1 = T_{tm} + t_{pdi}$$

This time calculation gives the time of the first sample in the data set. The following calculation describes how to calculate the time of any other sample in the data set.

N is the number of samples in the data set

t_{si} is the sample interval

$$T_i = T_1 + (i - 1) * t_{si} \quad \text{for } i = 1 \text{ to } N$$

For other possible configurations of the trigger logic see Annex J.

5.9.3 Trigger logic based on event recognition

The functional block diagram in Figure 6 depicts a TransducerChannel that is triggered by an event sensor. An event sensor may have additional embedded actuators to implement features such as a programmable analog threshold and hysteresis or a programmable digital pattern.

The event sensor has a special hardware relationship to one or more TransducerChannels. This shall be identified as a ChannelGroup in the Meta TEDS. A trigger message issued to the event sensor causes the event sensor to begin monitoring. The occurrence of the event, which may cause the Data Transport Logic to issue a trigger acknowledgment (if appropriate), may also be used to trigger the associated TransducerChannel(s).

The manufacturer may not honor a trigger message addressed to the TransducerChannels so configured.

5.9.4 Over-triggering a TransducerChannel

Irrespective of the time needed to read the TransducerChannel data, the TBC is expected to wait for at least the duration of the *TransducerChannel sampling period* between successive triggers. The *TransducerChannel sampling period* is in the TransducerChannel TEDS (see 7.4.2.20).

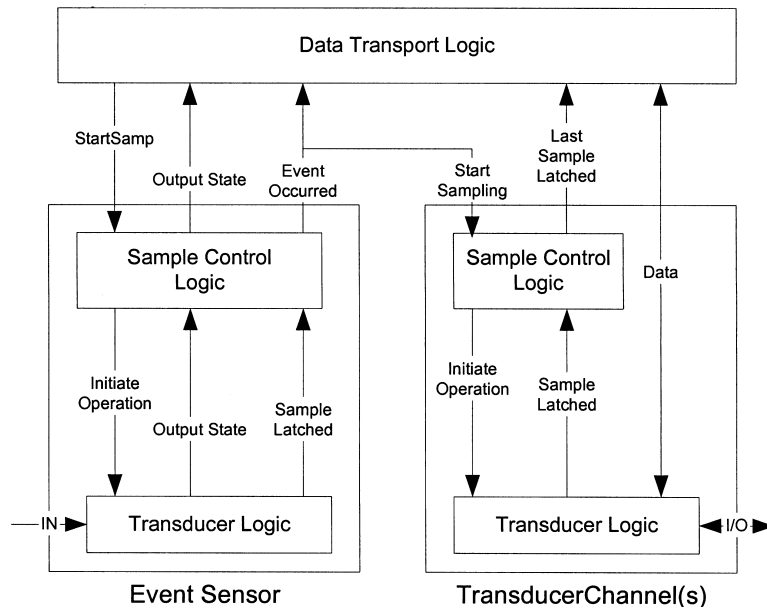


Figure 6—Event sensor output used as a trigger

Table 6—Response to over-triggering

Operating mode		
Streaming	Buffered	Response to the trigger
Yes	Don't care	Ignore all except the first trigger.
No	Yes	Accept the trigger as a command to start on another buffer when it completes the current buffer. A uniform interval between samples shall be maintained.
No	No	Ignore triggers until previous operation is complete.

The term *over-triggering* refers to issuing a second trigger before the *TransducerChannel* sampling period has expired. If this happens, the *TransducerChannel* may not have completely processed the previous trigger. Table 6 lists the response to over-triggering based upon the operating mode of the *TransducerChannel* as described in 5.8.

When in the non-streaming buffered mode, the unit treats a trigger received before all samples in a data set have been latched as a pre-trigger. In other words, it accepts the trigger as a command to start on another buffer when it completes the current buffer in such a way that all samples be acquired or applied while maintaining a uniform sample interval. If more than one extra trigger is received while processing a buffer, they are ignored. If this occurs, the time of a sample cannot be derived from the time that a trigger was sent.

5.10 Synchronization

A fundamental requirement for the large-scale acquisition of data from distributed multipoint sensor arrays is the inclusion of an intrinsic capability for the system to provide time-synchronous triggering of the various data converters. The individual sampling of the spatially distributed sensors must be coordinated within rather precise time intervals to assure that the array-wide data points are acquired at essentially the same instant, and thus represent an accurate temporal *snapshot* of the measured

parameters. There are several features built into this standard that are intended to aid in accomplishing this end. The following subclauses describe these features.

5.10.1 Methods of obtaining synchronicity

The basic tool that is used to obtain synchronization is the trigger. Since triggers may be addressed to groups of TransducerChannels, the members of the group are sampled synchronously within the limits of the propagation time of the bus and the variation of the delay between the receipt of the trigger by the TBIM and its application in the TransducerChannel.

5.10.1.1 Using triggers

The AddressGroup and global triggers, as described in 5.9.1.1, are the simplest way of achieving synchronous sampling of the data. This method works well with TransducerChannels that acquire or process a single sample for each trigger, but is less effective with TransducerChannels that have data sets that contain more than one sample. It also has limitations when used over long cables or with extremely tight synchronization requirements.

5.10.1.2 Using a delayed trigger

The problem of different delays associated with different lengths of cable between the TBC and the TBIMs and different delays within the TBIM may be addressed by adding the delayed trigger capability within the TBIMs. This is described in more detail in J.1. To make this feature of the TBIMs useful, this requires that the TBC have a method of determining the delay associated with each TBIM on the bus. The *Reflect* message described in 10.3.4.11 is used to implement this capability.

5.10.1.3 Using the synchronization signal

The synchronization signal provides a synchronous clock that is the same frequency for all TBIMs on the bus. The use of the synchronization signal makes the use of AddressGroup or global triggers with TransducerChannels that have a data set size greater than one more useful. However, it does not directly address the different delays associated with long cables.

5.10.2 Synchronization signal

The TBC shall generate a synchronization clock and transmit it on the transducer bus. The TBIMs may receive this clock, buffer it and use it to generate the appropriate clocks within the TBIM. The details of how this clock is transmitted by the TBC to the TBIMs are given in 10.4.2.

5.11 CommunicationsChannels

This standard recognizes two classes of CommunicationsChannels. One class of CommunicationsChannels is open to all users by some method that is determined in the Data Link and Physical layers of the protocol stack. This CommunicationsChannel is known as the TBIM CommunicationsChannel. The other class of CommunicationsChannels is dedicated to a particular TransducerChannel or TBC and that is the only device that may transmit using this CommunicationsChannel. These are called Data CommunicationsChannels.

To further differentiate between communications capabilities, Data and Physical CommunicationsChannels are defined in the standard. Data CommunicationsChannels are defined in the standard without regard for how they are physically implemented. Data CommunicationsChannels are mapped

to physical CommunicationsChannels in the Data Link or Physical layers of the protocol stack. Unless the context clearly indicates that a physical CommunicationsChannel is being specified, all references to a CommunicationsChannel refer to Data CommunicationsChannels.

5.11.1 TBIM CommunicationsChannel

The TBIM CommunicationsChannel is always assigned as Data CommunicationsChannel zero. The data link or physical layers of the protocol stack provide the mapping of this function into the physical realization of the CommunicationsChannel.

5.11.2 Data CommunicationsChannels

Data CommunicationsChannels are primarily used for streaming data and are assigned to TransducerChannel numbers. The data link or physical layers of the protocol stack provide the mapping of this function into the physical realization of the CommunicationsChannel.

5.12 Status

Status registers shall be implemented for the TBIM and for each implemented TransducerChannel in a TBIM. The returned status for the TBIM represents the state of the TBIM as a whole. In many cases, a bit in the TBIM status represents the logical *OR* of corresponding bits in all implemented TransducerChannels.

Certain of these status bits are duplicated in the *Status octet* in the Reply protocol. See 8.7.3 for a listing of the bits in the *Status octet*.

Each bit in the status register represents the presence or absence of a particular condition. A one in the appropriate bit position shall represent the presence of a condition.

Each status register is used in conjunction with an associated service request mask (see 5.13.1) to control which status bits will be used to generate a service request.

Status bits defined by this standard are tabulated in Table 7. Some status bits are *reserved* for future versions of this standard. Some bits are optional and the TBIM manufacturer may choose not to implement them. Bits designated as *open to manufacturers* may be used to report conditions not represented by bits that are already defined. New bit definitions in the TBIM status registers shall reflect conditions within the TBIM as a whole. Status bits labeled *reserved*, and unimplemented *optional* and *open to manufacturers* status bits, shall be reported as zero when read.

Bits in any status register may be cleared in one of several ways as shown in the following list:

- Some bits shall be cleared immediately when the condition they report goes away. Any bit in the TBIM status register that is defined as the *OR* of the corresponding bit in the TransducerChannel status registers shall be cleared in this manner.
- Other bits shall be cleared when the condition they report goes away *and* a status register read is performed.
- Status bits may also be changed by a change in operating states. The operating states are described in 5.3.1.

The first way shall apply unless the bit definition explicitly says that it is cleared when read.

Table 7—Status bits

Bit	TBIM status bits	TransducerChannel status bits	Required/optional
0	Service request	Service request	Required
1	Trigger acknowledged	Trigger acknowledged	Required
2	Invalid command	Reserved	Required
3	Command rejected	Command rejected	Required
4	Missed data or event	Missed data or event	See 5.12.5
5	Data/event	Data/event	See 5.12.6
6	Hardware error	Hardware error	See 5.12.7
7	Not operational	Not operational	Required
8	Bad frame received	Reserved	Required
9	Data available	Data available	Optional
10	Busy	Busy	Optional
11	Failed calibration	Failed calibration	Optional
12	Failed self-test	Failed self-test	Optional
13	Data over range or under range	Data over range or under range	Optional
14	Corrections disabled	Corrections disabled	Optional
15	Consumables exhausted	Consumables exhausted	Optional
16	Reserved	Not-the-first-read-of-this-data-set	Required
17	Protocol error	Protocol error	—
18	Reserved	Reserved	—
19	Reserved	Reserved	—
20	Reserved	Reserved	—
21	Reserved	Reserved	—
22	Reserved	Reserved	—
23	Reserved	Reserved	—
24	Open to manufacturers	Open to manufacturers	—
25	Open to manufacturers	Open to manufacturers	—
26	Open to manufacturers	Open to manufacturers	—
27	Open to manufacturers	Open to manufacturers	—
28	Open to manufacturers	Open to manufacturers	—
29	Open to manufacturers	Open to manufacturers	—
30	Open to manufacturers	Open to manufacturers	—
31	Open to manufacturers	Open to manufacturers	—

During power-on or reset initialization, data transport may be held off longer than the hold-off times specified in the TEDS. The TBIM shall assure that reads of status registers return an accurate representation of the TBIM's state. The TBIM shall hold off a read of any status register for which this is not the case.

The status register shall be accessed by the **Read status** command defined in 6.1.2.5. The returned status shall be 4 octets wide. In Table 7, the column "TBIM status bits" defines the information that is returned if the TBIM is addressed. If a TransducerChannel is addressed, the bits are as defined under the column titled "TransducerChannel status bits."

5.12.1 Service request bit

The *TransducerChannel service request* bit of any TransducerChannel shall be set when that TransducerChannel is requesting service, and shall be cleared when read unless a condition requiring service persists. Service Request masks are used to define conditions for which a TransducerChannel requests service.

See 5.13.1 for a description of the service Request mask.

The *TBIM service request* bit shall be set whenever the TBIM is requesting service, as defined by the TBIM masks. It shall be cleared when read, unless a condition requiring service persists.

A *TransducerChannel service request* bit shall be implemented for each TransducerChannel in a TBIM. A *TBIM service request* bit shall be implemented for each TBIM.

The *service request bit* will be cleared by a change of operating state if the status bit that caused it to be set is cleared.

5.12.2 Trigger acknowledged bit

The *TransducerChannel trigger acknowledged* bit shall be set by the TransducerChannel when it acknowledges the trigger. After global triggers, AddressGroup triggers, or triggers not requiring an acknowledgment, the appropriate *TransducerChannel trigger acknowledged* bit(s) shall be set by the TBIM for each effected TransducerChannel when it would have acknowledged the trigger. The *TBIM trigger acknowledged* bit shall be the OR of all *TransducerChannel trigger acknowledged* bits within the TBIM.

The *TransducerChannel trigger acknowledged* bit shall be cleared when read. The *TBIM trigger acknowledged* bit shall be set as long as any TransducerChannel within the TBIM has its *TransducerChannel trigger acknowledged* bit set.

The *TransducerChannel trigger acknowledged* bit shall be cleared when the TransducerChannel transitions from the halted to the operating state. The operating states are described in 5.3.1.

NOTE—While the TransducerChannel does not accept triggers in the halted state, the bit shall remain set until it is read or otherwise cleared if it was set when the TransducerChannel transitioned from the operating state to the halted state.

5.12.3 Invalid command bit

The *TBIM invalid command* bit shall be set whenever the TBIM detects an unimplemented command or a read or write to an unimplemented function. It shall be cleared when read.

This bit is not affected by a change of operating state.

5.12.4 Command rejected bit

The *Command rejected* bit shall be set whenever the TBIM detects a valid command that cannot be executed because of the current mode of operation of the TBIM or TransducerChannel. It shall also be set if an argument to the command is not acceptable for any reason. It shall be cleared when read.

Command rejected status bit is mandatory and shall be implemented for the TBIM and all TransducerChannels.

This bit is not affected by a change of operating state.

5.12.5 Missed data or event bit

The *TransducerChannel missed data or event* bit shall be set coincident with a data sample time of an event sensor or sensor with the sampling mode set to free-running if the *TransducerChannel* is in the operating state but is not currently triggered. The exception to the previous statement is that this bit shall not be set before the first trigger on any such *TransducerChannel* after the bit has been cleared. This bit shall be cleared when read.

The *TBIM missed data or event* bit shall be set when any *TransducerChannel missed data or event* bit is set.

This status bit shall be implemented for any event sensor or any sensor capable of operating with the sampling mode set to free-running. It shall also be implemented in any *TBIM* that contains such a sensor or event sensor.

The *TransducerChannel missed data or event* bit shall be cleared when the *TransducerChannel* transitions from the halted to the operating state. The operating states are described in 5.3.1.

NOTE—While the *TransducerChannel* is not acquiring or consuming data in the halted state, the bit shall remain set until it is read or otherwise cleared if it was set when the *TransducerChannel* transitioned from the operating state to the halted state.

5.12.6 Data/event bit

The *TransducerChannel data/event* bit shall be set at the data sample time for a sensor in the operating state when the sampling mode is set to free-running or when an event sensor detects an event. It shall be cleared when read. The *TBIM data/event* bit shall be set when any of the *TransducerChannel data/event* bits are set.

This status bit shall be implemented for any event sensor or any sensor capable of operating with the sampling mode set to the free-running state. It shall also be implemented in any *TBIM* that contains such a sensor or event sensor.

The *TransducerChannel data or event* bit shall be cleared when the *TransducerChannel* transitions from the halted to the operating state. The operating states are described in 5.3.1.

NOTE—While the *TransducerChannel* is not acquiring or consuming data in the halted state, the bit shall remain set until it is read or otherwise cleared if it was set when the *TransducerChannel* transitioned from the operating state to the halted state.

5.12.7 Hardware error bit

The *TransducerChannel hardware error* bit shall be set when the condition it reports becomes valid. It shall be cleared when read provided the condition it reports is no longer valid. An event sensor shall set this bit if consistency checks are made and fail. The *TBIM hardware error* bit shall be set when any of the *TransducerChannel hardware error* bits are set. The *TBIM* manufacturer may determine any additional criteria for hardware errors.

The *TBIM hardware error* bit in the returned *TBIM* status and the *TransducerChannel hardware error* bit shall be implemented for any *TBIM* that contains an event sensor if consistency checks are made (see 5.4.2).

This bit is not affected by a change of operating state.

5.12.8 Not operational bit

The *TransducerChannel not operational* bit shall be set when the TransducerChannel fails to comply with the manufacturers specifications. This bit shall be set during warm-up or any other condition when the TransducerChannel does not comply with specifications or is not operational.

The *TBIM not operational* bit shall be the OR of all of the *TransducerChannel not operational* bits. It shall also be set if the TBIM is not operational due to any other condition. The *TBIM not operational* bit shall be cleared when the condition causing it to be set is removed.

This bit is not affected by a change of operating state.

5.12.9 Bad frame received bit

A frame has been received from the transducer bus with checksum and/or CRC errors.

This bit is not affected by a change of operating state.

5.12.10 Data available bit

The *TransducerChannel data available* bit shall be set whenever the TransducerChannel has data available to be read. The *TBIM data available* bit shall be set when any TransducerChannel in the TBIM has data available to be read. They shall remain set as long as the condition persists.

The *TransducerChannel data available* bit shall be cleared when the TransducerChannel transitions from the halted to the operating state. The operating states are described in 5.3.1.

NOTE—While the TransducerChannel is not acquiring data in the halted state, the bit shall remain set until it is read or otherwise cleared if it was set when the TransducerChannel transitioned from the operating state to the halted state.

5.12.11 Busy bit

The *TransducerChannel busy* bit shall be set whenever a TransducerChannel cannot support read or write access of TransducerChannel data over the data transport. Commands that may cause the TransducerChannel to become busy include the **Run diagnostics** command, **Calibrate Transducer-Channel** command, **Reset** command, **Zero TransducerChannel** command, or any other command that interferes with normal operations. This bit shall be set only while the condition that it reports persists. The *TBIM busy* bit shall be the OR of all of the *TransducerChannel busy* bits in a TBIM.

This bit is not affected by a change of operating state.

5.12.12 Failed calibration bit

The *TransducerChannel failed calibration* bit and the *TBIM failed calibration* bit shall be set when a “calibration” check (see 6.1.1.4 or 6.1.1.8) fails to produce the expected results. It shall be cleared when it is read. The *TBIM failed calibration* bit shall be the OR of all of the *TransducerChannel failed calibration* bits in a TBIM.

This bit is not affected by a change of operating state.

5.12.13 Failed self-test bit

The *TransducerChannel failed self-test* bit and the *TBIM failed self-test* bit shall be set when a self-test fails to provide the expected results (see 6.1.9). It shall be cleared when read. The *TBIM failed self-test* bit shall be the OR of all of the *TransducerChannel self-test calibration* bits in a TBIM.

This bit is not affected by a change of operating state.

5.12.14 Data over range or under range bit

The *TransducerChannel data over range or under range* bit shall be set when the TransducerChannel detects an over range or under range condition. The *TBIM data over range or under range* bit shall be the OR of all of the *TransducerChannel data over range or under range* bits in a TBIM. This status bit shall be cleared when read.

An overrange condition exists when the output of the TransducerChannel represents a value greater than or more positive than the value given in the *Upper range limit* field of the TransducerChannel TEDS (see 7.4.2.6). An underrange condition exists when the output of the TransducerChannel represents a value less than or more negative than the value given in the *Lower range limit* field of the TransducerChannel TEDS (see 7.4.2.5).

The *TransducerChannel data over range or under range* bit shall be cleared when the TransducerChannel transitions from the halted to the operating state. The operating states are described in 5.3.1.

NOTE—While the TransducerChannel is not acquiring or consuming data in the halted state, the bit shall remain set until it is read or otherwise cleared if it was set when the TransducerChannel transitioned from the operating state to the halted state.

5.12.15 Corrections disabled bit

The *Corrections disabled* bit shall be set in any TBIM or TransducerChannel that is capable of correcting the data when that capability is disabled. It shall be implemented for any TransducerChannel or TBIM with this capability. This bit shall be set to zero for any TransducerChannel or TBIM not implementing this capability.

This bit is not affected by a change of operating state.

5.12.16 Consumables exhausted bit

The *TransducerChannel consumables exhausted* bit shall be set when the TransducerChannel can no longer obtain any consumables that it requires to continue operation. The *TBIM consumables exhausted* bit shall be the OR of all of the *TransducerChannel consumables exhausted* bits in a TBIM. This status bit shall only be set while the condition persists.

This bit is not affected by a change of operating state.

5.12.17 Not-the-first-read-of-this-data-set bit

The *Not-the-first-read-of-this-data-set* bit of any TransducerChannel shall be set when a data set is read more than once. This bit shall be cleared when read.

This bit is not affected by a change of operating state.

5.12.18 Protocol error bit

A frame has been received from the transducer bus with illegal fields that are not defined in the upper layer protocols.

5.13 Service request logic

Service requests are sent from a TBIM to the TBC when some condition exists within the TBIM that requires attention. They are roughly analogous to interrupts in computers. However, unlike computer interrupts, service requests have no independent mechanism for communicating with the TBC that the condition exists. The *TBIM service request* bit is contained in the status octet in every reply message or may be read using the **Read status** command. This means that the TBC is not able to respond within some minimum time to a service request.

5.13.1 Service request masks

The TBIM shall contain a service request mask register for the TBIM and for each implemented TransducerChannel in a TBIM. These registers are 4 octets wide. Writing a one to any bit position in the service request mask register allows the service request bit to be set when the corresponding bit in the status register is set. See Figure 7 for details.

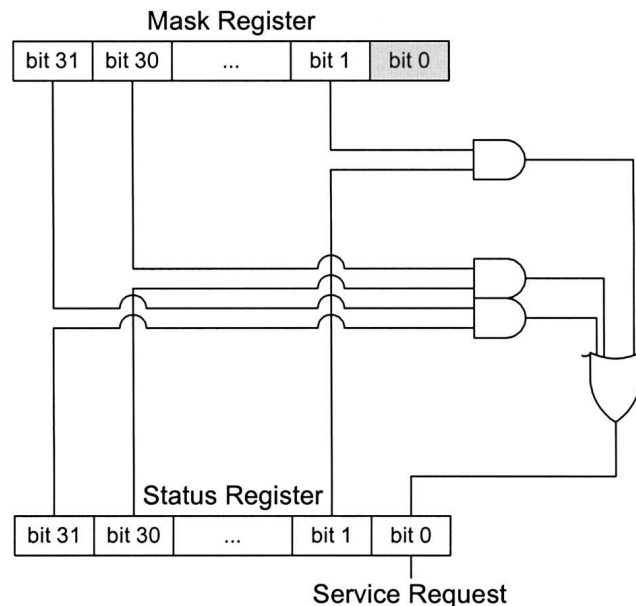


Figure 7—TBIM or TransducerChannel service request masking

The service request mask register bit positions correspond one-to-one with the bit positions in the status register, as defined in Table 7. The value placed into the bit 0 position of the service request mask register is not used as this corresponds to the service request bit. Since the service request bit directly generates the service request, it cannot be masked. The default power up value for the service request mask registers is all ones (i.e., all status bits can generate service requests). A **Reset** command shall not affect the value of this register.

5.13.2 Service requests

The service request signal is used in conjunction with the status registers and the service request mask registers to indicate exceptional conditions in the TBIM. The TBC typically reads all the TransducerChannels' status to determine which TransducerChannels are requesting service, and for what reason. The TBC is not required to respond to a service request immediately.

Figure 8 shows the logic used to generate the *TBIM service request*.

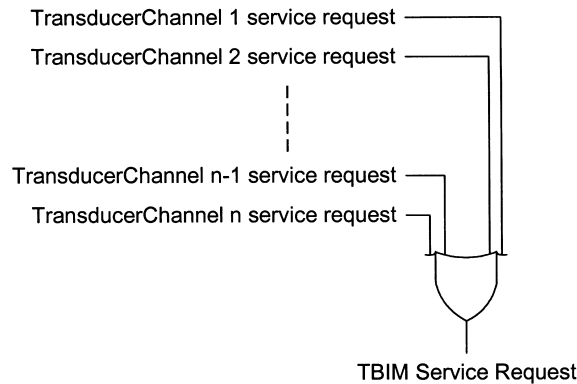


Figure 8—TBIM service request generation

5.14 Hot-swap capability

Hot-swap capability exists within the system on two levels. It shall be possible to disconnect a TBIM from the bus or to connect a TBIM to the bus without powering down the bus and without damage to either the TBIM being inserted or anything else connected to the bus. Transient impact upon transmissions in progress when a TBIM is added or removed from the bus is allowed but no permanent damage is allowed.

NOTE—To detect TBIMs being added to the bus, the TBC needs to issue an **Initiate discovery** command and each TBIM on the bus that has not been set up responds with its UUID. Detecting a TBIM that has been removed from the bus requires that the TBC send a command that requires a reply, such as a **Read status** command, and to wait for the TBIM to respond. If the TBIM does not respond within a time out period, the TBIM has been removed from the bus.

6. Commands

Commands are divided into two categories, standard and manufacturer-defined. Regardless of the category the command is divided into two octets. The most significant octet shall be used to define the class of the command. The least significant octet, called the *function*, shall identify the specific command within the class. For example, if the most significant octet defines the **Read TEDS block** class of commands, the least significant octet then specifies the TEDS to be read. Table 8 lists the various command classes.

A TBIM may generate a reply to a command under either of two circumstances. The first circumstance is when the command itself requires a reply. An example of this is a **Query data block** command. The second circumstance where a reply is generated is when the *Acknowledgment Expected* flag is set in the received command. In this second case, the acknowledgment is included in the response to the

command if the command itself requires a response. If the command does not require a response but the *Acknowledgment Expected* flag is set in the received command, then a reply shall be generated. In either case, the Reply Protocol (see 8.7) shall be used to transmit the response. If an acknowledgement is required, it shall be issued after the completion of the operation required by the command.

The information described as being passed is passed through the Command Services Protocol, as described in 8.6, if the command is being transmitted or received. The following paragraphs describe what should be in the command dependent octets in 8.6. If a reply to a command is being transmitted or received, the Reply Protocol (see 8.7) is utilized.

In the tables in this section, some enumerations are *reserved* for future versions of this standard. Some enumerations are optional and the TBIM manufacturer may choose not to implement them. Enumerations designated as *open to manufacturers* may be used to designate conditions not described by the standard.

6.1 Standard commands

The control function allows commands to be sent to the TBIM as a whole, or to each TransducerChannel thereof, which affect their state or operation. The list of standard command classes is given in Table 8.

The TBIM shall respond to all unimplemented commands by setting the *TBIM invalid command* bit in the status register. See 5.12.3 for a complete description of this bit.

Table 8—Standard command classes

Class	Category	Address				
		TBIM	TransducerChannel		Global	Group
			Normal	Proxy		
0	Reserved	—	—	—	—	—
1	Initialization	See Table 9	See Table 9	See Table 9	See Table 9	See Table 9
2	Operational	See Table 16	See Table 16	See Table 16	See Table 16	See Table 16
3	Query TEDS	See Table 20	See Table 20	See Table 20	No	No
4	Read TEDS block	See Table 20	See Table 20	See Table 20	No	No
5	Write TEDS block	See Table 20	See Table 20	See Table 20	No	No
6	Update TEDS	See Table 20	See Table 20	See Table 20	No	No
7	Set operating mode	Yes	Yes	Yes	Yes	Yes
8	Read operating mode	No	Yes	No	No	No
9	Run diagnostics	Yes	Yes	Yes	Yes	Yes
10–127	Reserved	—	—	—	—	—
128–255	Open for manufacturers	—	—	—	—	—