



IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats

IEEE Instrumentation and Measurement Society

Technical Committee on Sensor Technology

IEEE Std 1451.0™-2024
(Revision of 1451.0-2007)

STANDARDS

IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats

Developed by the

Technical Committee on Sensor Technology
of the
IEEE Instrumentation and Measurement Society

Approved 15 February 2024

IEEE SA Standards Board

Abstract: The common functions, network services, transducer services, and Transducer Electronic Data Sheet (TEDS) formats for members of the IEEE 1451 family of standards to be interoperable in both network interface and transducer interface are provided. The common functions and characteristics that are to be performed by a network capable application processor (NCAP), which is a 1451 server, and all devices that implement the NCAP are defined. A set of network services that includes a set of commands and replies to access transducer data and TEDS data from the NCAP (1451 server) by the Internet of Things (IoT) applications (APPs) that are 1451 clients are also defined. Application programming interfaces (APIs) of network services are outlined to facilitate communications between the APPs (1451 clients) and NCAPs. Furthermore, the common functions and characteristics that are to be performed by a transducer interface module (TIM) and all devices that implement the TIM are discussed. A set of transducer services that includes a set of commands and replies to facilitate the setup and control of the TIM, as well as reading and writing the data used by the NCAP, are outlined. APIs of transducer services are also discussed to facilitate communications between the NCAPs and TIMs. Finally, the formats for a TEDS and methods to access the TEDS are specified, along with the common functions and characteristics that are to be performed by an APP and all devices that implement the APP.

Keywords: actuator, application programming interface, communication protocol, IEEE 1451.0™, network-capable application processor, sensor, smart transducer, transducer channel, transducer electronic data sheet, transducer interface module, universal unique identification, wireless interface

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

Copyright © 2024 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 26 June 2024. 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, Incorporated.

PDF: ISBN 979-8-8557-0691-8 STD26916
Print: ISBN 979-8-8557-0692-5 STDPD26916

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

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.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA, and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive 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 comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](#). An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, 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 order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#). For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#). Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the [IEEE SA Patent Policy](#).

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 by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Participants

At the time this standard was completed, the Common Functionality and TEDS Working Group had the following membership:

Eugene Song, *Chair*
Kang B. Lee and John Schmalzel, *Vice Chairs*
Cheng-Jen Chen, *Secretary*
Stan Woods, *Editor*
Riccardo Brama, *Leader of IEEE-SA OPEN 1451.0 Project*

Reza Abrishambaf	Nishi Hiroaki	Michael Tocher
Jorgen Bække	Victor Huang	Russel Trafford
Robin Brooks	Gopalakrishna Kuppa	Kim Fung Tsang
Michal Brychta	Jiana Li	Peter Waher
Marco Carratu	Yu Cheng Liu	Hao Wang
Yu Chen	William Miller	Jin Wang
Grazia D'Elia	Jay Murphy	Darold Wobschall
Xinzheng Feng	Vincenzo Paciello	Yang Wei
Angelo Ferraro	Markus Rentschler	Jun Wu
Evan Franklin	Vincent Randal	Wentao Zhang
Giada Giorgi	Antonio Espirito Santo	Ke Zhou
	Robert Stemp	

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

Mihaela Albu	Darren Hoppins	Eugene Song
Boon Chong Ang	Victor Huang	Robert Stemp
Jorgen Baekke	Piotr Karocki	Thomas Strasser
Riccardo Brama	John Kay	Kim Fung Tsang
Bill Brown	Kang B. Lee	Benton Vandiver
Dietmar Bruckner	James McBride	John Vergis
Cheng-Jen Chen	Chris Mullins	Stephen Webb
Zhiman Chen	Rajesh Murthy	Darold Wobschall
Ritwik Chowdhury	Arthur Neubauer	Jun Wu
Rodney Cummings	Hiroaki Nishi	Ruqiang Yan
Wenbin Dai	Mfanasibili Nkonyane	Z. J. Yang
Patrick Diamond	Vincenzo Paciello	Yu Yuan
Lee Eccles	Venkatesha Prasad	Oren Yuen
Antonio Espirito Santo	Sergio Rapuano	Janusz Zalewski
A. Ferraro	Oleg Roizman	Yining Zhang
Jean-Sebastien Gagnon	John Schmalzel	Daidi Zhong
Werner Hoelzl	Veselin Skendzic	Nan Zhou

When the IEEE SA Standards Board approved this standard on 15 February 2024, it had the following membership:

David J. Law, *Chair*
TBD, *Vice Chair*
Gary Hoffman, *Past Chair*
Alpesh Shah, *Secretary*

Sara R. Biyabani
Ted Burse
Stephen Dukes
Doug Edwards
J. Travis Griffith
Guido R. Hiertz
Ronald W. Hotchkiss
Hao Hu

Yousef Kimiagar
Joseph L. Koepfinger*
Howard Li
Xiaohui Liu
John Haiying Lu
Kevin W. Lu
Hiroshi Mano
Paul Nikolich

Robby Robson
Jon W. Rosdahl
Mark Siira
Lei Wang
F. Keith Waters
Sha Wei
Philip B. Winston
Don Wright

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 1451.0-2024, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats.

This standard is intended to provide a basis for all future members of the IEEE 1451 family of standards that use network and transducer (sensor and/or actuator) interfaces. It should also be adopted by the existing members of the IEEE 1451 family of standards as they are revised in the future to provide the highest degree of compatibility and interoperability among the members of the family. This standard is the revision of IEEE Std 1451.0™-2007 to address the needs of new technologies, such as cyber-physical systems (CPS) and Internet of Things (IoT) for smart cities and smart grids. This standard does not apply to IEEE Std 1451.4™, which only provides a size-constrained TEDS and a mixed-mode transducer interface. This standard also does not apply to IEEE Std 1451.7™, which only provides a specific TEDS and a radio frequency identification (RFID) interface. IEEE Std 1451.0™-2024 is based on and intended to replace IEEE Std 1451.0-2007; the IEEE 1451.0-2024 version is not backward compatible with IEEE Std 1451.0-2007.

NOTE—The standard uses the term “TEDS” to refer to single or multiple instances.¹

The functional relationships between this standard and the other members of the family are shown in Figure 1. The functional relationships mean that services or interfaces exist between the standards. The solid lines mean required or mandatory services or interfaces. The dashed lines mean optional services or interfaces.

- IEEE Std 1451.2™² defines a digital interface for connecting transducers to microprocessors. A TEDS and its data formats are described. It defines an electrical wireline interface, read and write logic functions to access the TEDS, and a wide variety of transducers.
- IEEE Std 1451.5™ defines a wireless interface for sensors. It specifies radio-specific protocols for this wireless interface. It defines communication modules that connect the wireless transducer interface module (WTIM) and the network capable application processor (NCAP) using the radio-specific protocols. It also defines the TEDS for the radio-specific protocols.
- IEEE Std 1451.4™ defines the protocol and interface that allows analog transducers to communicate digital information with an IEEE 1451 object. It also defines the format of the TEDS. The standard does not specify the transducer design, signal conditioning, or the specific use of the TEDS.
- IEEE Std 1451.7™ defines the transducer to RFID systems communication protocols and TEDS formats.
- IEEE Std 21451™-001 defines a set of transducer signal treatment services.
- IEEE Std 21451™-002 establishes operating aspects and additional TEDS based on the IEEE 1451 family of standards for low-power smart sensors and actuators, concerning the signal conditioning electronics and the processing unit operation.
- IEEE Std 1451.1.4™ defines the eXtensible Messaging and Presence Protocol interface based on IEEE 1451.0 network services.
- IEEE Std 1451.1.5™ defines the Simple Network Management Protocol interface for IEEE 1451.0-based sensor networks.
- IEEE Std 1451.1.6™ defines the Message Queuing Telemetry Transport interface based on IEEE 1451.0 network services.

¹ Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

² Information on references can be found in Clause 2.

- IEEE Std 1451.5.5™ defines a smart sensor interface using Long-Range protocol based on IEEE 1451.0 transducer service.
- IEEE Std 1451.5.6™ defines a smart sensor interface using Sigfox protocol based on IEEE 1451.0 transducer service.
- IEEE Std 1451.5.10™ defines a smart sensor interface using Narrow-band Internet-of-Things protocol based on IEEE 1451.0 transducer service.
- IEEE Std 1451.99™ defines a method for data sharing, interoperability, and security of messages over a network, where sensors, actuators and other devices can interoperate, regardless of underlying communication technology.

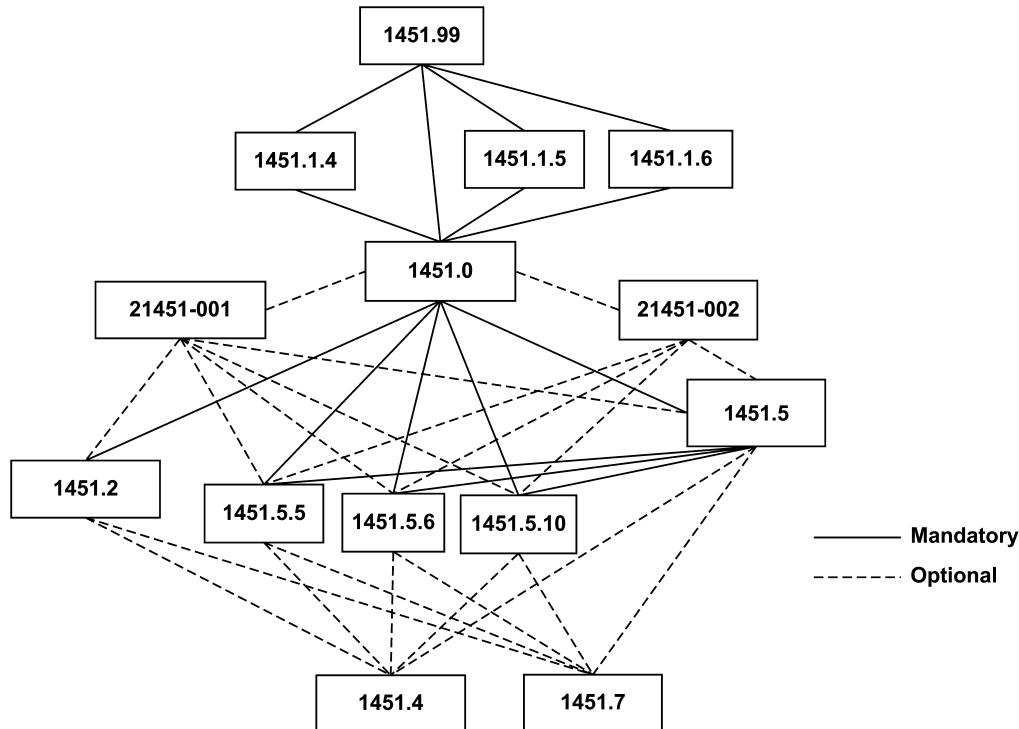


Figure 1—Functional relationships between IEEE 1451.0 and other family members

The purpose of this family of standards is to allow manufacturers to build elements of a system that are interoperable. To accomplish this goal, the IEEE 1451 family of standards divides the parts of a system into three general categories of devices: Network Capable Application Processor (NCAP), Transducer Interface Module (TIM), and Application (APP). The NCAP (1451 server) functions as a gateway between the APP (1451 client) and the TIM (1451 sensors and/or actuators). The NCAP is a processor-based device that has two interfaces: network and transducer. The physical network interface to the wide area network or APP is not specified in any of this family of standards. The physical transducer interface between the NCAP and the TIMs is defined in the remaining members of the family of standards. This standard defines a set of network services and a set of transducer services. Different manufacturers may build the NCAPs, APPs, and TIMs, and if they comply with this standard, they should be interoperable.

This standard describes the functions performed by a TIM. Provisions are made for a high level of addressing that is independent of the physical medium-level and low-level protocols that are used to implement communications. The standard defines the common characteristics of all devices that implement the TIMs. The timing of the acquisition or processing of the data samples is described. Methods of grouping the outputs from multiple transducers within one TIM are defined. Common status words are also defined.

A standard set of commands and replies is defined to facilitate the setup and control of the TIMs, as well as to read and write the data used by the system. Commands and replies are also provided for reading and writing the TEDS that supplies the system with the operating characteristics that are needed to use the transducer modules. A method of adding manufacturer unique commands is included.

This standard also defines the common functions and characteristics that are to be performed by an NCAP, as well as all devices that implement the NCAP. It defines a set of network services that includes a set of commands and replies to access transducer data and TEDS data from the NCAP by the APP.

In addition, this standard provides formats for the TEDS. Several TEDS are defined in the standard. Four of these TEDS are required, and the remaining TEDS are optional. Some TEDS are provided to allow the user to define information and store it in the TEDS. These TEDS could be applied for sensor or/and actuator, TIM, and NCAP. This standard provides areas that are “open to manufacturers.” Any use of these areas, however, may compromise the “plug-and-play” potential of NCAPs and TIMs.

The IEEE 1451 family of standards for smart transducer interfaces for sensors and actuators defines communication interfaces specifications for sensor networks and systems. As shown in Figure 2, the communication architecture for the IEEE 1451 family of standards defines two levels of interfaces between sensors/actuators and applications—a transducer interface and a network interface. The transducer interface is a device-level interface that communicates sensors/actuators data between the TIM and the NCAP. In the IEEE 1451 family, it includes standards for mixed-mode transducer interface (1451.4), wireline interface (1451.2), wireless interfaces (1451.5 or 1451.5.X), and RFID-to-sensor interface (1451.7). At the network level, the family consists of network interface standards for 1451.1.4 (or 21451-1-4), 1451.1.5 (or 21451-1-5), and 1451.1.6 (21451-1-6) network interfaces, as well as the 1451.99 framework for the harmonization of 1451-based sensor networks with other non-1451 IoT networks or systems. All these interfaces are based on the core of the IEEE 1451 family of standards, IEEE Std 1451.0, which defines a common function and messaging, common data format, and metadata such as the TEDS that enable the access of sensors and actuators data and information and pass them to the IoT, Industrial Internet of Things (IIoT), and CPS clients or applications via various 1451.1.X network interfaces, through both IEEE 1451 and non-IEEE 1451 networks and systems.

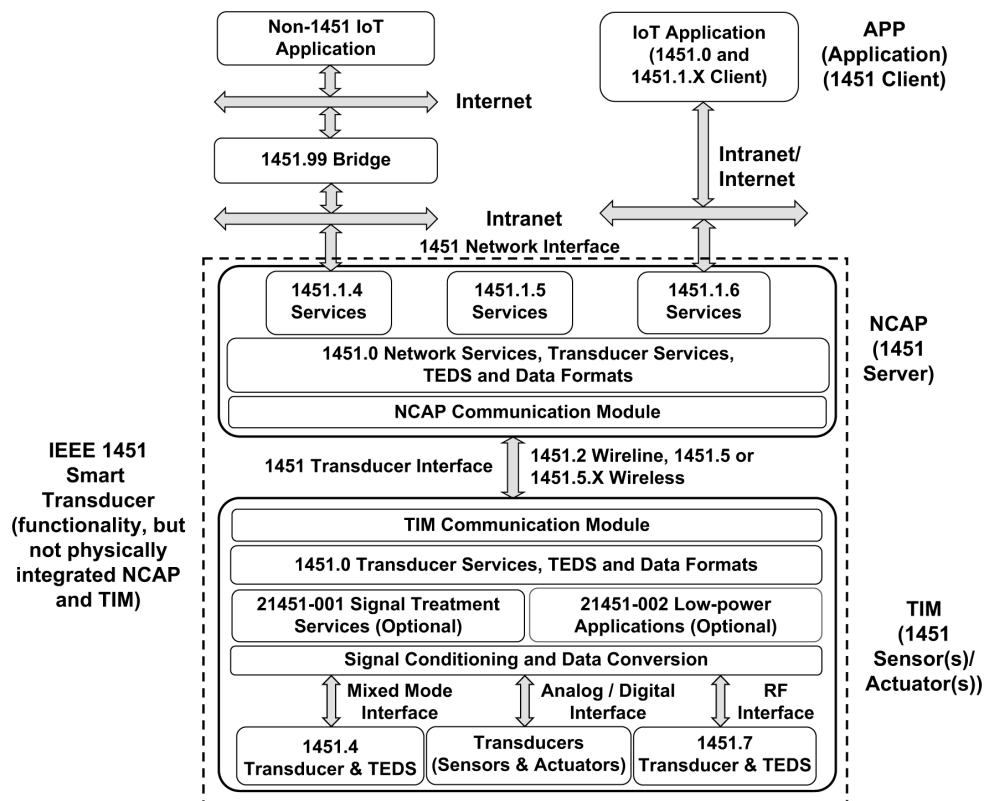


Figure 2—Communication architecture for IEEE 1451 family of standards

Figure 3 shows the clauses relevant to creating IEEE 1451.0-compliant APPs, NCAPs, and TIMs.

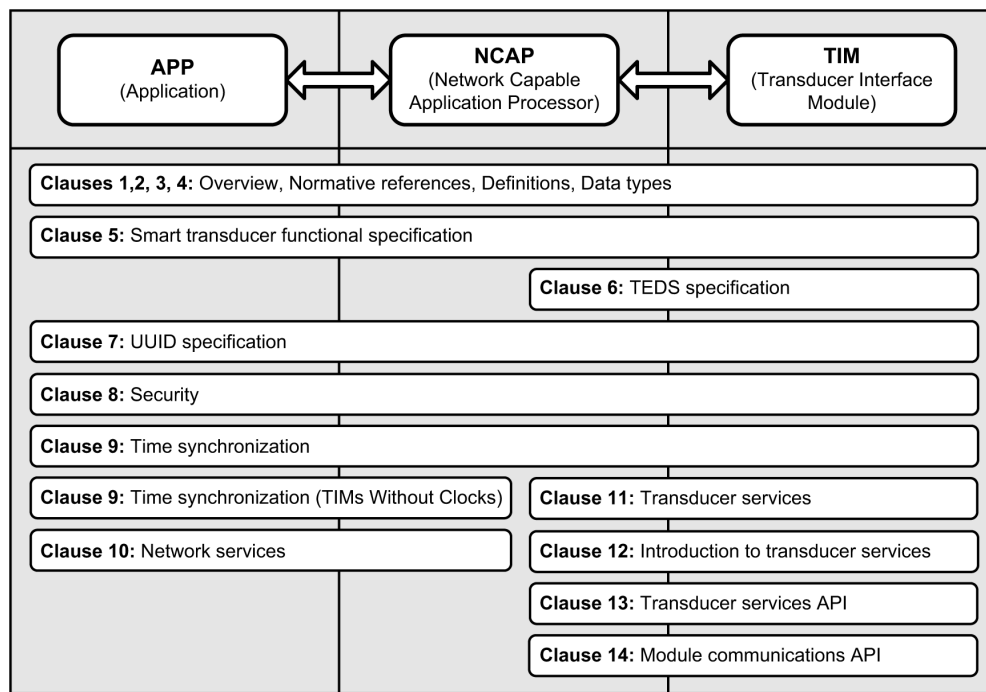


Figure 3—Normative clauses for IEEE 1451 APPs, NCAPs, and TIMs

Figure 4 shows suggested annexes for further understanding the normative clauses in IEEE Std 1451.0.

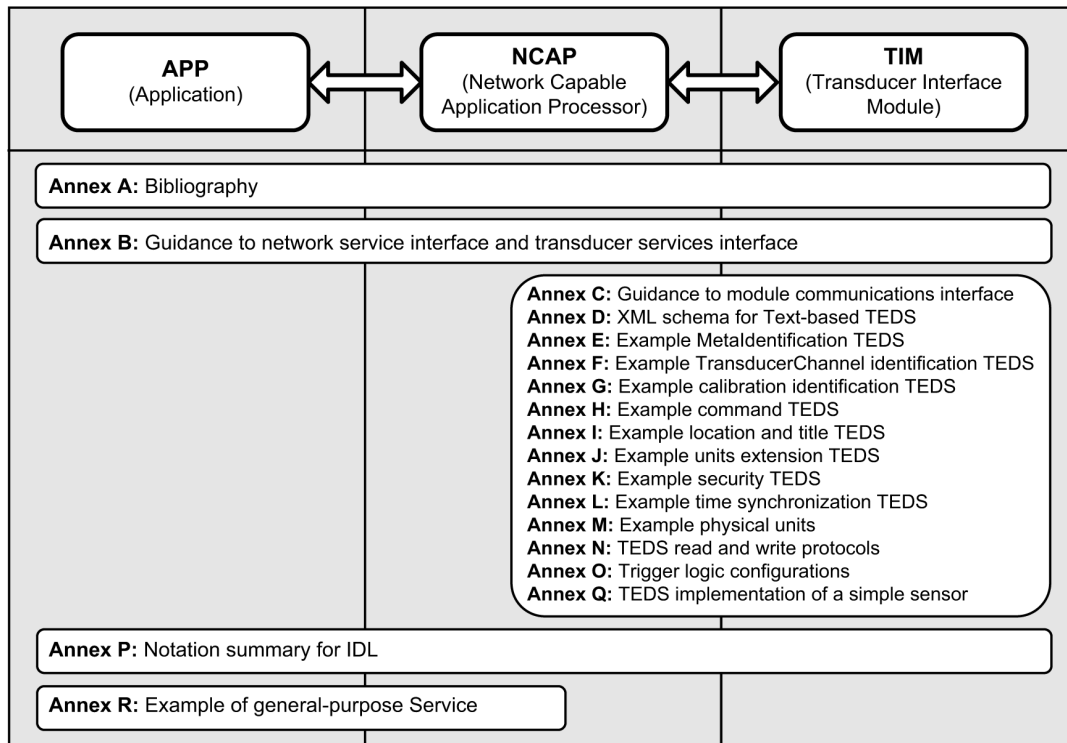


Figure 4—Informative annexes for IEEE 1451 APPs, NCAPs, and TIMs

Contents

1. Overview	19
1.1 Scope	22
1.2 Purpose	22
1.3 Word usage	22
1.4 Conformance	22
2. Normative references	24
3. Definitions, acronyms, and abbreviations	25
3.1 Definitions	25
3.2 Acronyms and abbreviations	29
4. Data types	29
4.1 Unsigned octet integer	29
4.2 Unsigned 16-bit integer	30
4.3 Signed 32-bit integer	30
4.4 Unsigned 32-bit integer	30
4.5 Unsigned 48-bit integer	30
4.6 Signed 64-bit integer	30
4.7 Single-precision real	31
4.8 Double-precision real	31
4.9 String	31
4.10 Boolean	31
4.11 TimeRepresentation	32
4.12 Data types for associated applications	33
4.13 Physical Units	33
4.14 Arbitrary octet array	34
4.15 String array	35
4.16 Boolean array	35
4.17 Array of 8-bit signed integers	35
4.18 Array of 16-bit signed integers	35
4.19 Array of 32-bit signed integers	35
4.20 Array of 8-bit unsigned integers	35
4.21 Array of 16-bit unsigned integers	36
4.22 Array of 32-bit unsigned integers	36
4.23 Array of single-precision real numbers	36
4.24 Array of double-precision real numbers	36
4.25 Array of TimeDuration data types	36
4.26 Array of TimeInstance data types	36
5. Smart transducer functional specification	37
5.1 IEEE 1451 family reference model	37
5.2 IEEE 1451 communication models	43
5.3 Plug-and-play capability	46
5.4 Addresses	46
5.5 Common characteristics	48
5.6 Transducer Electronic Data Sheets	50
5.7 TransducerChannel type descriptions	55
5.8 Embedded TransducerChannels	58
5.9 TransducerChannel groups	58
5.10 TransducerChannel proxy	59

5.11 Attributes and operating modes	60
5.12 Triggering.....	65
5.13 Universal unique identification (UUID)	72
5.14 Security	72
5.15 Time synchronization	72
5.16 Status	72
5.17 Service request logic.....	79
5.18 Hot-swap capability	80
6. TEDS specification.....	80
6.1 General format for TEDS	80
6.2 Order of octets in numeric fields	82
6.3 TEDS identification header	82
6.4 Meta-TEDS.....	83
6.5 TransducerChannel TEDS	94
6.6 Calibration TEDS	119
6.7 Frequency Response TEDS	137
6.8 Transfer Function TEDS.....	140
6.9 Text-based TEDS	150
6.10 End User Application Specific TEDS.....	155
6.11 User's Transducer Name TEDS	156
6.12 Manufacturer-defined TEDS	158
6.13 PHY TEDS	159
6.14 Security TEDS.....	159
6.15 Geographic Location TEDS	161
6.16 Time Synchronization (TimeSync) TEDS.....	162
7. Universal unique identification	163
7.1 Overview of UUID	163
7.2 UUID specification.....	163
8. Security.....	165
8.1 Security requirements and recommendations for IEEE 1451.0-based sensor networks	165
8.2 Security of IEEE 1451.0 and P1451.1.X-based WAN Interface	171
8.3 Security of IEEE 1451.0 and (1451.5 or P1451.5.X)-based WLAN interface	172
9. Time synchronization	173
9.1 Time synchronization of IEEE 1451.0 and P1451.1.X-based user network or WAN	175
9.2 Time synchronization of IEEE 1451.0- and (1451.5 or P1451.5.X)-based wireless LANs (WLANs).....	177
9.3 Time synchronization of IEEE 1451.0 and 1451.2-based serial LANs (SLANs).....	178
10. Network services	180
10.1 Discovery services.....	182
10.2 Transducer access services	190
10.3 TEDS access services	215
10.4 Event notification services.....	219
11. Transducer services	230
11.1 Transducer service message structures	230
11.2 Transducer service command-and-reply messages.....	236

12. Introduction to the IEEE 1451.0 transducer services API	266
12.1 API goals	266
12.2 API design decisions.....	267
12.3 IEEE1451 Dot0.....	269
13. Transducer services API.....	279
13.1 IEEE1451Dot0::TransducerServices::TimDiscovery	280
13.2 IEEE1451Dot0::TransducerServices::TransducerAccess	281
13.3 IEEE1451Dot0::TransducerServices::TransducerManager	288
13.4 IEEE1451Dot0::TransducerServices::TedsManager	295
13.5 IEEE1451Dot0::TransducerServices::CommManager	298
13.6 IEEE1451Dot0::TransducerServices::AppCallback	299
14. Module communications API	301
14.1 IEEE1451Dot0::ModuleCommunication::Comm	302
14.2 IEEE1451Dot0::ModuleCommunication::P2PComm	306
14.3 IEEE1451Dot0::ModuleCommunication::NetComm.....	310
14.4 IEEE1451Dot0::ModuleCommunication::Registration.....	320
14.5 IEEE1451Dot0::ModuleCommunication::P2PRegistration	322
14.6 IEEE1451Dot0::ModuleCommunication::NetRegistration	323
14.7 IEEE1451Dot0::ModuleCommunication::Receive	326
14.8 IEEE1451Dot0::ModuleCommunication::P2PReceive	326
14.9 IEEE1451Dot0::ModuleCommunication::NetReceive.....	328
Annex A (informative) Bibliography	330
Annex B (informative) Guidance to network service interface and transducer services interface	332
B.1 Simple polled measurement.....	332
B.2 Non-blocking read example	333
B.3 Generic SendCommand() mechanism.....	334
B.4 APP, NCAP, and TIM two-way communications	334
Annex C (informative) Guidance to module communications interface	339
C.1 Module communications API discussion	339
C.2 Symmetric APIs	341
C.3 Implementation choices	341
C.4 Implementation examples	341
C.5 Node communication parameters.....	342
C.6 The destination identifier “destId” parameter	342
C.7 The communication session “commId” parameter	343
C.8 The message transaction identifier “msgId” parameter.....	343
C.9 Memory constrained implementations	343
C.10 IEEE 1451.X or P1451.5.X communication state machines.....	344
C.11 Communication sequence	347
Annex D (informative) XML schema for text-based TEDS.....	349
D.1 Introduction to text-based TEDS	349
D.2 Schema.....	350
D.3 Include file “SmartTransducerDataModel.xsd”.....	350
D.4 Include file “TextTEDS.xsd”.....	357
D.5 MetaIdentificationTEDS.xsd	360
D.6 TransducerChannelIdentificationTEDS.xsd	361
D.7 CalibrationIdentificationTEDS.xsd	362
D.8 CommandsTEDS.xsd.....	363

D.9 LocationAndTitleTEDS.xsd	368
D.10 UnitsExtensionTEDS.xsd	369
D.11 GeographicLocationTEDS.xsd	370
D.12 SecurityTEDS.xsd	371
D.13 TimeSyncTEDS.xsd	372
Annex E (informative) Example Meta-Identification TEDS.....	374
E.1 Introduction.....	374
E.2 Example instance document.....	374
Annex F (informative) Example TransducerChannelIdentification TEDS	375
F.1 Introduction	375
F.2 Example instance document.....	375
Annex G (informative) Example Calibration Identification TEDS	376
G.1 Introduction	376
G.2 Example instance document	376
Annex H (informative) Example Commands TEDS	377
H.1 Introduction	377
H.2 Example situation	377
H.3 Example instance document	377
H.4 Example instance document	378
H.5 Example instance document	379
Annex I (informative) Example Location and Title TEDS	381
I.1 Introduction.....	381
I.2 Example situation.....	381
I.3 Example instance document.....	381
Annex J (informative) Example Units Extension TEDS	383
J.1 Introduction.....	383
J.2 Example situation.....	383
J.3 Example instance document.....	383
Annex K (informative) Example security TEDS.....	384
K.1 Introduction	384
K.2 Example of security TEDS of IEEE 1451.5 802.11.....	384
Annex L (informative) Example time synchronization TEDS	387
L.1 Introduction.....	387
L.2 Example of TimeSync TEDS of IEEE P1451.1.6.....	387
Annex M (informative) Example Physical Units	389
M.1 Examples.....	392
M.2 System considerations.....	394
M.3 Conclusions.....	395
M.4 Acknowledgments.....	395
Annex N (informative) TEDS read and write protocols.....	396
N.1 TEDS access	396
N.2 First step in a TEDS access.....	396
N.3 Writing a TEDS	396
N.4 Reading a TEDS	397

Annex O (informative) Trigger logic configurations.....	398
O.1 Trigger logic augmented with an embedded time delay actuator	398
O.2 Trigger logic augmented with TimeInstance sensor	399
O.3 Trigger logic augmented with time interval sensor.....	400
Annex P (informative) Notation summary for IDL	403
P.1 Key features of IDL.....	403
P.2 Example of IDL notation with explanation	403
Annex Q (informative) TEDS implementation of a simple sensor.....	407
Q.1 Meta-TEDS.....	407
Q.2 TransducerChannel TEDS	410
Q.3 Calibration TEDS	417
Q.4 User’s transducer name TEDS.....	419
Q.5 Required commands	421
Annex R (informative) Example of general-purpose service—authentication.....	426

IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats

1. Overview

This standard introduces the concepts of a transducer interface module (TIM) (IEEE 1451 sensors and/or actuators), a network capable application processor (NCAP) (IEEE 1451 server), and an IoT application (APP) (IEEE 1451 client). A TIM is a module that contains the interface, signal conditioning, analog-to-digital and/or digital-to-analog conversion, and, in many cases, the transducer. A TIM may range in complexity from a single sensor or actuator to units containing many transducers (sensors and actuators). An NCAP is the hardware and software that provides the gateway function between the TIMs and APPs. An APP is a software module running on a device or a device that performs network communications with the NCAP to access transducer data and TEDS, and processes transducer data for various IoT applications. Other members of the standards family provide the communications interfaces between APPs and multiple NCAPs and the communication interfaces between an NCAP and one or more TIMs. Three types of transducers are recognized by this standard. They are sensors, event sensors, and actuators.

A transducer is denoted “**smart**” in this context because of the following features:

- It is described by a machine-readable Transducer Electronic Data Sheet (TEDS).
- The control and data associated with the transducer are via network and transducer interfaces.
- Triggering, status, and control are provided to support the proper functioning of the transducer channel.
- A smart transducer provides functions, such as sensing and/or actuating, signal and data conversion, intelligent data processing, network communication with an application or outside world, self-identification, and self-description based on TEDS.

An NCAP controls a TIM by means of a transducer interface medium. The NCAP mediates between the TIM and the APP and may provide local intelligence.

This standard defines a set of network services that provide communications between the APP (IEEE 1451 client) and the NCAP (IEEE 1451 server). It also defines an application programming interface (API) for NCAP to communicate with TIMs. The API is also provided between the IEEE 1451.0 layer and the underlying physical communications layers usually referred to in this standard as “IEEE 1451.X” or “P1451.5.X”. These API definitions are provided for systems that have visible interfaces. For TIMs and NCAPs with a single set of hardware and software without regard to distinguishing separate interfaces between IEEE 1451.0 functionality and IEEE 1451.X or P1451.5.X functionality, the API is optional as long as the messages at visible interfaces conform to the rest of the standard. The definition of these APIs is to facilitate modular design to the extent that multiple suppliers can provide different functionality and yet have the various parts integrate seamlessly.

This standard defines TIMs that can be plugged into a system and can be used without having to add special drivers and profiles or make any other changes to the system. This process is referred to as a “plug-and-play” operation. The primary features that enable a plug-and-play operation are the TEDS and the command set. A TIM may be added to or removed from an active IEEE 1451 physical layer 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 defines the common functions and characteristics that are to be performed by an NCAP that is a 1451 server, and all devices that implement the NCAP. It defines a set of network services that include a set of commands and replies to access transducer data and TEDS data from the NCAP for APPs.

This standard is organized as follows:

Clause 1: “Overview” provides the scope of this standard.

Clause 2: “Normative 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 an APP, an NCAP, a TIM, and all TransducerChannels in a TIM.

Clause 6: “TEDS specification” specifies the transducer electronic data sheet structure and content.

Clause 7: “Universal unique identification” specifies a global identifier for each TIM, NCAP, and APP.

Clause 8: “Security” specifies the existing standardized security methodology for IEEE 1451-based transducer networks including network interface and transducer interface.

Clause 9: “Time synchronization” specifies existing standardized time synchronization methodologies for IEEE 1451-based transducer networks including IEEE 1451 user network WAN and 1451 LAN.

Clause 10: “Network services” specifies a set of network services for the NCAP that provides these network services, including message structures and a set of command/reply message pairs. Message structures are used to encapsulate the information being passed between an APP and NCAPs. A set of command/reply message pairs provides methods to access NCAPs. Commands provide the command syntax and the expected replies.

Clause 11: “Transducer services” specifies a set of transducer services for the TIMs that provides a set of transducer services, including message structures and a set of command/reply message pairs. Message structures are used to encapsulate the information being passed between an NCAP and TIMs. A set of command/reply message pairs provides methods to access TIMs. Commands provide the command syntax and the expected replies.

Clause 12: “Introduction to the IEEE 1451.0 transducer service API” gives the common features of the two APIs.

Clause 13: “Transducer services API” gives the API that an application would use to utilize this standard.

Clause 14: “Module communications API” gives the API that this standard would use to communicate to the TIMs using the communications features of a physical interface defined by another member of the IEEE 1451 family of standards.

Annex A: “Bibliography” provides references for additional information about topics referred to in this document.

Annex B: “Guidance to network service interface and transducer services interface” gives examples of the use of this interface for measurement and control applications that interact with the IEEE 1451.0 layer using the transducer services interface.

Annex C: “Guidance to module communication interface” gives additional guidance for the logical communication between the NCAP and TIMs or between TIMs using the Module Communication API.

Annex D: “XML Schema for text-based TEDS” gives the basic schemas for the Text-based TEDS defined in this standard.

Annex E: “Example MetaIdentification TEDS” gives an example of a possible MetaIdentification TEDS.

Annex F: “Example TransducerChannel identification TEDS” gives an example of a possible Transducer Channel Identification TEDS.

Annex G: “Example calibration identification TEDS” gives an example of a possible Calibration Identification TEDS.

Annex H: “Example commands TEDS” gives an example of a possible Commands TEDS.

Annex I: “Example location and title TEDS” gives an example of a possible Location and Title TEDS.

Annex J: “Example units extension TEDS” gives an example of a possible Units Extension TEDS.

Annex K: “Example security TEDS” gives an example of possible security TEDS.

Annex L: “Example time synchronization TEDS” gives an example of possible TimeSync TEDS.

Annex M: “Examples of physical units” gives a series of examples of implementations of Physical Units using the representation specified in this standard.

Annex N: “TEDS read and write protocols” describes processes that may be used to write or read the TEDS.

Annex O: “Trigger logic configurations” shows some possible configurations of the trigger logic allowed in this standard.

Annex P: “Notation summary for IDL” is intended to give guidance on the use of IDL notation in this standard.

Annex Q: “TEDS implementation of a simple sensor” is an example of a simple sensor implemented using the structures defined in this standard.

Annex R: “Example of general purpose service—authentication” is an example of using the general-purpose service defined in this standard.

1.1 Scope

This standard develops a set of common functionality and TEDS formats for the family of IEEE 1451 smart transducer interface standards. This functionality is independent of the physical communications media and includes the common network services and transducer services required to control and manage smart transducers and TEDS. It defines a set of implementation-independent APIs. This standard also defines universal unique identification, time synchronization, and security for the IEEE 1451 standards family. This project does not specify signal conditioning and conversion, physical media, or how the transducer data and TEDS data are used in applications.

1.2 Purpose

Several standards in the IEEE 1451 family share certain characteristics, but no common set of network services, transducer services, and TEDS formats exists that facilitates interoperability among these standards. The purpose of this standard is to define a set of common network services, transducer services, and TEDS formats for IEEE 1451 smart transducers to achieve smart transducer interoperability and facilitate interoperability among the 1451 standards family. IEEE Std 1451.0-2007 has been revised to address the needs of new technologies, such as cyber-physical systems (CPS) and Internet of Things (IoT) for smart cities and smart grids.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).^{1, 2}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

1.4 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 an open opportunity for continued technical improvement and differentiation.

¹ The use of the word *must* is deprecated and shall not be used when stating mandatory requirements, *must* is used only to describe unavoidable situations.

² The use of *will* is deprecated and shall not be used when stating mandatory requirements, *will* is only used in statements of fact.

TIM (IEEE 1451 sensors and/or actuators) implementation shall be deemed in conformance with this standard provided the following requirements are met:

- The TIM supports the required functional specifications identified in Clause 5.
- The TIM supports the transducer service message structures specified in Clause 11.
- The TIM supports the required commands and replies specified in Clause 11.
- The TIM supports required TEDS that have the format and content specified in Clause 6.
- The TIM supports the Universal Unique Identification (UUID) defined in Clause 7, security defined in Clause 8, and time synchronization defined in Clause 9.
- The TIM supports one of the communications protocols and transducer physical media defined by another member of the IEEE 1451 family of standards.

NOTE—Several features are highly desirable and are supported by this standard, but they are not practical to make into hard requirements. It is desirable for the sense element of a sensor to be an integral part of the TIM, but for sensing elements like structural strain gages 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 TIM, but there are systems where the environment and/or the physical size make this impractical so the standard allows the TEDS to be located remotely from the TIM.

An NCAP (IEEE 1451 server) implementation shall be deemed in conformance with this standard provided the following requirements are met:

- The NCAP supports the required functional specifications identified in Clause 5.
- The NCAP supports the UUID defined in Clause 7, security defined in Clause 8, and time synchronization defined in Clause 9.
- The NCAP supports the transducer service message structures specified in Clause 11.
- The NCAP supports the required commands and replies specified in Clause 11.
- The NCAP supports one of the communications protocols and transducer physical media defined by another member of the IEEE 1451 family of standards (e.g., IEEE Std 1451.5) when there is a separable physical interface between NCAP and TIM.
- The NCAP supports the network services defined by Clause 10.
- The NCAP supports one of the communication protocols defined by another member of the IEEE 1451 family of standards (e.g., IEEE P1451.1.6).

APP (IEEE 1451 client) implementation shall be deemed in conformance with this standard provided the following requirements are met:

- The APP supports the required functional specifications identified in Clause 5.
- The APP supports the network services defined by Clause 10.
- The APP supports one of the communication protocols defined by another member of the IEEE 1451 family of standards (e.g., IEEE P1451.1.6).
- The APP supports the UUID defined in Clause 7, cybersecurity defined in Clause 8, and time synchronization defined in Clause 9.

1.4.1 Byte and bit ordering (endianness)

In this standard, the default byte transfer and communication ordering (Endianness) is the most significant byte of a word first and the least significant byte last. This ordering applies to neither the storage of the bytes nor the data manipulations within the NCAP or TIM. Many different ordering schemes are utilized by various microcomputer and microcontroller platforms that for the purposes of this standard do not specify a particular ordering. This specification of Big-Endian concerns only the communication of data between various system components (i.e., NCAP, TIM, and other interfacing components).

The default communication bit ordering is also specified as the most significant bit first and the least significant bit last. This specification only applies to the external communication of data between NCAP and TIM and between other system components (e.g., between APP and NCAP) if any bit data information communications take place. The internal NCAP and TIM bit manipulations and storage orders are not within the scope of this standard and are not specified.

When not otherwise stated in figures, tables, and diagrams, the same order is applied, and most significant bytes and bits are always aligned starting from the top leftmost character down to the bottom rightmost.

As an illustrative example, refer to 11.1.1.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they need to be understood and used, so each referenced document is cited in the text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

ANSI X3.4-1986 (Reaff 1992), Coded Character Sets—7-bit American National Standard Code For Information Interchange.³

Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation, 26 November 2008.⁴

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

IEEE Std 802.3™-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 1451.2™-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 1451.4™-2004, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats.

³ ANSI publications are available from the American National Standards Institute (<https://www.ansi.org/>).

⁴ Documents on the eXtensible Markup language can be downloaded from <https://www.w3.org/TR/2008/REC-xml-20081126/> or ordered from the World Wide Web Consortium, c/o MIT, 32 Vassar Street, Room 32-G515, Cambridge, MA 02139 USA.

⁵ IEEE publications are available from The Institute of Electrical and Electronics Engineers (<https://standards.ieee.org/>).

⁶ The IEEE standards or products referred to in Clause 2 are trademarks owned by The Institute of Electrical and Electronics Engineers, Incorporated.

IEEE Std 1451.5™-2007, IEEE Standard for a Smart Transducer Interface for Sensors and Actuator—
Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats.

IEEE Std 1451.7™-2010, IEEE Standard for Smart Transducer Interface for Sensors and Actuators--
Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer
Electronic Data Sheet Formats.

IEEE Std 1588™-2019, IEEE Standard for a Precision Clock Synchronization Protocol for Networked
Measurement and Control Systems.

ISO 639:1988-04-01 (E/F), Codes for the Representation of Names of Languages.⁷

ISO/DIS 19136, Geographic information—Geography Markup Language.⁸

ISO/IEC 14750:1999-03-15, Information technology—Open Distributed Processing—Interface Definition
Language.⁹

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.¹⁰

1451.X: Any member of the IEEE 1451 smart transducer standards family based on the IEEE 1451.0 transducer interface.

P1451.1.X: Any member of the IEEE 1451 smart transducer standards family based on the IEEE 1451.0 network interface.

P1451.5.X: Any member of the IEEE 1451 smart transducer standards family based on the IEEE 1451.0 transducer interface and P1451.5.X wireless media.

actuator: A transducer that accepts a data sample or samples and converts them into an action. The action may be completely contained within the TIM or may change something outside of the TIM.

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

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

⁷ ISO publications are available from the International Organization for Standardization (<https://www.iso.org/>) and the American National Standards Institute (<https://www.ansi.org/>).

⁸ DS publications are available from Dansk Standard (<https://www.ds.dk/d/>).

⁹ IEC publications are available from the International Electrotechnical Commission (<https://www.iec.ch>) and the American National Standards Institute (<https://www.ansi.org/>).

¹⁰ *IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>. An IEEE account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

application (APP): A software module (IEEE 1451 client) running on a device that performs network communications with the NCAP (IEEE 1451 server) to access transducer data and the TEDS, as well as to processes transducer data for various IoT applications.

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.

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

ControlGroup: Manufacturer specifications that define the inherent relationships between the TransducerChannels of a multichannel TIM. This ControlGrouping information is not normally used by the TIM itself. This information is used to identify TransducerChannels that are used to control some characteristic of another TransducerChannel. For example, a ControlGroup could be used to identify an actuator that is used to set the threshold of an analog event sensor.

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

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

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

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).

digital interface: A communications media and a protocol for transferring information by binary means only.

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

embedded transducer: A device that behaves as a transducer from the point of view of the NCAP even though nothing outside of the TIM is sensed or changed. Embedded transducers are useful for setting or reading the operating parameters of other transducers.

enumeration: 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 to have a specific meaning. Values without meaning are declared to be unused or reserved for future use. Enumeration is the process of declaring the encoding of human interpretable information in a manner convenient for digital electronic machine storage and interchange. Any subclause that defines a TEDS data field to be enumerated shall contain a table that defines the meaning of the data field for each numeric value possible. The meanings encoded in each data field shall be specific and unique to that data field and only to that data field. The value becomes meaningless if not associated with the data field and its defining table.

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.”

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

interoperability: The ability of two or more systems or components to exchange information and to use the information that has been exchanged (IEEE Std 610.12™-1990 [B8]¹¹).

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

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

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.

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

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

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

network capable application processor (NCAP): A device between the TIMs and the Application. The NCAP performs network communications with the Application via the network interface, network communications with TIM via the transducer interface, and data conversion or other processing functions. NCAP is an IEEE 1451 server that provides network services for Applications (IEEE 1451 clients), and a client that can access transducer data from TIMs via the transducer interface.

not-a-number (NaN): As defined in IEEE Std 754™-1985,¹² a bit pattern of a single-precision or double-precision real number data type that is a result of an invalid floating point operation. For single-precision real numbers, the bit pattern shall have an exponent of 255 (decimal) and a nonzero mantissa. The sign is not considered in determining whether a value is not a number.

octet: A group of 8 bits. (In the United States, an octet is usually referred to as a byte.)

packet: A block of information that is to be passed by the physical layer between devices in a single transmission.

sample latched: The term is used in a sensor to signal 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.

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

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

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

smart transducer: A transducer that provides functions beyond those necessary for generating a correct representation of a sensed or controlled quantity:

¹¹ The numbers in brackets correspond to those of the bibliography in Annex A.

¹² Information on references can be found in Clause 2.

- A smart transducer consists of transducers (sensors and/or actuators), signal conditioning and data conversion, an application processor or data processing unit, and a network communication interface and TEDS.
- A smart transducer provides functions, such as sensing and/or actuating, signal and data conversion, intelligent data processing, network communication with an application or outside world, self-identification, and self-description based on TEDS.
- From a sensor network perspective, a smart transducer can be constructed or deployed as an NCAP and a number of TIMs. See 5.1.

synchronization signal: For the purposes of this standard, a signal is transmitted by the NCAP to provide clock or time synchronization signals to a TIM or group of TIMs. Synchronization signals may not be available with all physical layer standards. Low-cost and performance TIMs may not implement a receiver for the synchronization signal.

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

transducer interface module (TIM): A module that contains the TEDS, logic to implement the transducer interface, the transducer(s) or connection to the transducer(s), and any signal conversion or signal conditioning. A TIM can contain sensors and/or actuators.

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

TransducerChannel number: A 16-bit number assigned to an individual TransducerChannel within a TIM by the manufacturer.

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

Transducer Electronic Data Sheet (TEDS): An electronic data sheet describing a TIM or a TransducerChannel, or an NCAP. The structures of multiple TEDS are described in this standard. Note that the term “TEDS” is used to refer to single or multiple instances in this standard.

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

transfer function: The function that defines the response of a TransducerChannel in both amplitude and frequency.

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

VectorGroup: Manufacturer specifications that define the inherent relationships between the TransducerChannels of a multichannel TIM. This VectorGrouping information is not normally used by the TIM itself. This information is normally used by NCAP applications to properly compose human readable displays or in formulating other computations. For example, VectorGroupings may be used to indicate which TransducerChannels represent each of the three vector axes of a three-axis vector measurement.

virtual TEDS: A TEDS that is stored permanently in a location other than the TIM or NCAP.

wireless transducer interface module (WTIM): A transducer interface module with a wireless communication interface. *See: transducer interface module.*

3.2 Acronyms and abbreviations

AD	anno domini
A/D	analog to digital
ADC	analog-to-digital converter
CRS	Coordinate Reference System
D/A	digital to analog
DTD	document type declaration, ref. eXtensible Markup Language (XML)
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
LAN	local area network
MJD	modified Julian date
MQTT	message queuing telemetry transport
MSB	most significant bit
NaN	not a number
NCAP	network capable application processor
OMG	Object Management Group
PHY	physical layer
SI	International System of Units, reference <i>The International System of Units (SI)</i> (Taylor [B13])
SLAN	serial local area network
SNMP	Simple Network Management Protocol
TAI	International Atomic Time
TCP	Transmission Control Protocol
TEDS	Transducer Electronic Data Sheet
TIM	transducer interface module
TLV	type/length/value
UDP	User Datagram Protocol
USASCII	U.S. American standard code for information interchange (ANSI X3.4-1986)
UTC	universal time coordinated also known as <i>universal coordinated time</i>
UUID	universal unique identification
VPN	virtual private network
WAN	wide area network
WLAN	wireless area network
WPAN	wireless personal area network
WTIM	wireless transducer interface module
Xdc	transducer
XML	eXtensible Markup Language
XMPP	eXtensible Messaging and Presence Protocol

4. Data types

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

4.1 Unsigned octet integer

Symbol: UInt8

Size: 1 octet

IDL: typedef octet UInt8;

This data type represents positive integers from 0 to 255.

4.2 Unsigned 16-bit integer

Symbol: UInt16

Size: 2 octets

IDL: `typedef unsigned short UInt16;`

A UInt16 may take any value between 0 and 65 535.

4.3 Signed 32-bit integer

Symbol: Int32

Size: 4 octets

IDL: `typedef long Int32;`

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

4.4 Unsigned 32-bit integer

Symbol: UInt32

Size: 4 octets

IDL: `typedef unsigned long UInt32;`

This data type is used to represent positive integers from 0 to $4\,294\,967\,295$.

4.5 Unsigned 48-bit integer

Symbol: UInt48

Size: 6 octets

IDL: `typedef unsigned long UInt48;`

This data type is used to represent positive integers from 0 to $281\,474\,976\,710\,655$.

4.6 Signed 64-bit integer

Symbol: Int64

Size: 8 octets

IDL: `typedef long Int64;`

This data type is used to represent a signed integer from $-9\,223\,372\,036\,854\,775\,808$ to $9\,223\,372\,036\,854\,775\,807$.

4.7 Single-precision real

Symbol: Float32

Size: 4 octets

IDL: `typedef float Float32;`

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

4.7.1 Floating-point NaN in TEDS

According to IEEE Std 754-1985, a single-precision number with an exponent of 255 and a fractional portion (mantissa) that is nonzero is NaN regardless of the sign bit. The recommended value for use in a TEDS field for NaN is 0x7FFFFFFF (hex).

4.8 Double-precision real

Symbol: Float64

Size: 8 octets

IDL: `typedef double Float64;`

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

4.9 String

Symbol: `_String`

Size: variable

IDL: `typedef string _String; // Leading '_' to escape reserved IDL keyword`

The use of text strings, as type string as a basic type, shall conform with the use of OMG standards for IDL. A null character shall be used as the string terminator and is included in the string length specified in a text-based TEDS.

The XML used in 6.9 is the text-based TEDS. The controlling document for XML is the W3C Recommendation *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. All text strings in the TEDS shall conform to this recommendation or to a future update to it.

4.10 Boolean

Symbol: `_Boolean`

Size: 1 octet

IDL: `typedef boolean _Boolean; // Leading '_' to escape reserved IDL keyword`

The type `_Boolean` is a basic type that is defined in the OMG standards for IDL.

For the purposes of this standard, a bit or octet with a nonzero value is considered True. A zero value represents the False state of the variable.

4.11 TimeRepresentation

This abstract class defines the time representation. It is subclassed into TimeInstance. The definition of the two arguments is given in Table 1.

```
IDL: struct TimeRepresentation {
    UInt48 secs;
    UInt32 nsecs;
};
```

Table 1—TimeRepresentation data structure

Parameter	Type	Description
secs	UInt48	seconds —This unsigned 48-bit number represents the positive number of seconds from the beginning of the epoch (normally 0 hours on 1 January 1970). The secs field is the integer portion of the timestamp in units of seconds.
nsecs	UInt32	nanoseconds —The nsecs field is the fractional portion of the timestamp in units of nanoseconds. The nsecs field is always less than 10^9 .

This standard uses the time representation specified in IEEE Std 1588-2019.

4.11.1 TimeInstance

Time values are represented by this struct in the IEEE 1451.0 layer. This struct is appropriate when trying to represent a time value and not time duration.

```
IDL: struct TimeInstance {
    UInt48      secs;
    UInt32      nsecs;
};
```

The TimeInstance is based on an epoch that is defined as starting at midnight 1 January 1970. The definition of the two arguments is the same as shown in Table 1.

Example:

+2.000000001 seconds is represented by secs field = 0000 0000 0002₁₆ and nsecs field = 0000 0001₁₆.

4.11.2 TimeDuration

This time representation is used to specify a time interval rather than a time value. TimeDuration is the time interval expressed in nanoseconds, multiplied by 2^{+16} . Positive or negative time intervals outside the maximum range of this data type shall be encoded as the largest positive and negative values of the data type, respectively.

```
IDL: Int64 TimeDuration
```

Example:

2.5 ns is expressed as 0000 0000 0002 8000₁₆.

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

4.12 Data types for associated applications

The data types in this clause are not used in this standard but are provided to allow the applications using this standard to pass data in these formats.

4.12.1 8-bit signed integer

Symbol: Int8

Size: 1 octet

IDL: `typedef char Int8;`

This datatype represents integers from –128 to 127.

4.12.2 16-bit signed integer

Symbol: Int16

Size: 2 octets

IDL: `typedef short Int16;`

This data type represents integers from –32 768 to 32 767.

4.13 Physical Units

Symbol: UNITS

Size: 11 octets

IDL: `struct Units {
 UInt8 interpretation;
 UInt8 radians;
 UInt8 steradians;
 UInt8 meters;
 UInt8 kilograms;
 UInt8 seconds;
 UInt8 amperes;
 UInt8 kelvins;
 UInt8 moles;
 UInt8 candelas;
 UInt8 Units Extension TEDS Access Code
};`

Physical Units are a binary sequence of 10 octets that encode Physical Units as described in Table 2 and Table 3. Each field 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 examples of Physical Units, see Annex M. For further information, see Hamilton [B2].

Table 2—Physical Units data type structure

Field	Description	Data type	Number of octets
1	Physical Units interpretation—see Table 3	UInt8	1
2	$(2 * \text{<exponent of radians>}) + 128$	UInt8	1
3	$(2 * \text{<exponent of steradians>}) + 128$	UInt8	1
4	$(2 * \text{<exponent of meters>}) + 128$	UInt8	1
5	$(2 * \text{<exponent of kilograms>}) + 128$	UInt8	1
6	$(2 * \text{<exponent of seconds>}) + 128$	UInt8	1
7	$(2 * \text{<exponent of amperes>}) + 128$	UInt8	1
8	$(2 * \text{<exponent of kelvins>}) + 128$	UInt8	1
9	$(2 * \text{<exponent of moles>}) + 128$	UInt8	1
10	$(2 * \text{<exponent of candelas>}) + 128$	UInt8	1

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 2 through 10.

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

Table 3—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 (for example, a bit vector) and has no unit. Fields 2–10 shall be set to 128. The “digital data” type applies to data that do 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 (for example, hardness). Fields 2–10 are reserved and shall be set to 128.
6–255	—	Reserved.

4.14 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 comprising one or more primitive data types, arrays of primitive data types, or smaller OctetArrays.

4.15 String array

Symbol: StringArray

Size: varies

This data type comprises an arbitrary number of string data types (see 4.9), treated as an aggregate entity.

4.16 Boolean array

Symbol: BooleanArray

Size: varies

This data type comprises an arbitrary number of `_boolean` data types (see 4.10), treated as an aggregate entity.

4.17 Array of 8-bit signed integers

Symbol: Int8Array;

Size: varies

This data type comprises an arbitrary number of octets, treated as an aggregate entity made up of 8 bit signed integers (Int8).

4.18 Array of 16-bit signed integers

Symbol: Int16Array;

Size: varies

This data type comprises an arbitrary number of 16-bit signed integers (Int16), treated as an aggregate entity.

4.19 Array of 32-bit signed integers

Symbol: Int32Array;

Size: varies

This data type comprises an arbitrary number of 32-bit signed integers (Int32), treated as an aggregate entity.

4.20 Array of 8-bit unsigned integers

Symbol: UInt8Array;

Size: varies

This data type comprises an arbitrary number of octets, treated as an aggregate entity made up of 8 bit unsigned integers (UInt8).

4.21 Array of 16-bit unsigned integers

Symbol: UInt16Array;

Size: varies

This data type comprises an arbitrary number of 16-bit unsigned integers (Int16), treated as an aggregate entity.

4.22 Array of 32-bit unsigned integers

Symbol: UInt32Array;

Size: varies

This data type comprises an arbitrary number of 32-bit signed integers (Int16), treated as an aggregate entity.

4.23 Array of single-precision real numbers

Symbol: Float32Array;

Size: varies

This data type comprises an arbitrary number of single-precision real numbers as specified in IEEE Std 754-1985, treated as an aggregate entity.

4.24 Array of double-precision real numbers

Symbol: Float64Array;

Size: varies

This data type comprises an arbitrary number of double-precision real numbers as specified in IEEE Std 754-1985, treated as an aggregate entity.

4.25 Array of TimeDuration data types

Symbol: TimeDurationArray;

Size: varies

This data type comprises an arbitrary number of TimeDuration data types as specified in 4.11.2, treated as an aggregate entity.

4.26 Array of TimeInstance data types

Symbol: TimeInstanceArray;

Size: varies

This data type comprises an arbitrary number of TimeInstance data types as specified in 4.11.1, treated as an aggregate entity.

5. Smart transducer functional specification

This clause of the standard provides a description of the functions expected to be in an IEEE 1451 smart transducer. The details of these functions are provided in Clause 6 through Clause 14.

5.1 IEEE 1451 family reference model

The reference model shown in Figure 5 and Figure 6 shows the relationships among the various IEEE 1451 standards family members and consists of three components: APP (1451 client), NCAP (1451 server), and TIM. However, they are to be used to help define what is in each standard in the family. It is not intended to imply a required implementation of devices that comply with the standard. Figure 5 shows a functional view of the IEEE 1451 reference model. Figure 6 shows a network communication view of the IEEE 1451 reference model.

As shown in Figure 6, the NCAP communicates with TIMs by means of a transducer interface creating the TIMs LAN. Through this LAN network, a single NCAP can reach a multiplicity of TIMs. Despite a TIM, shown on the right, not being able to reach the WAN, it is able to offer a set of 1451.0 transducer services to the NCAP.

An NCAP communicates with APPs via the WAN that employs one or more network interfaces whose characteristics are abstracted by each one of the ancillary P1451.1.X network services implementing 1451.0 network services. Please note that the standard technologies, such as the one described in IEEE Std 802.2 and in IEEE Std 802.3-2002, are referenced as WAN notwithstanding their local reach.

By means of this architecture, different P1451.1.X networks can exchange 1451.0-compliant network service messages in a uniform way despite bridges that may be needed to accommodate WAN interface hardware peculiarities. It is beyond the scope of this standard to describe how this WAN bridging can be realized.

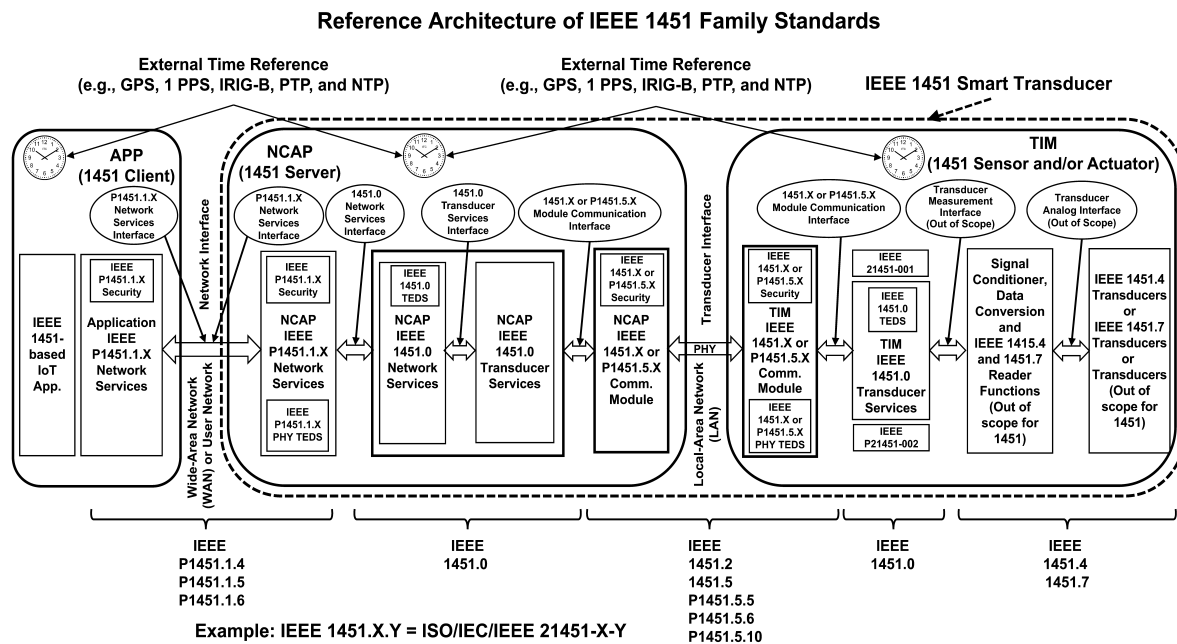


Figure 5—IEEE 1451.0 reference model

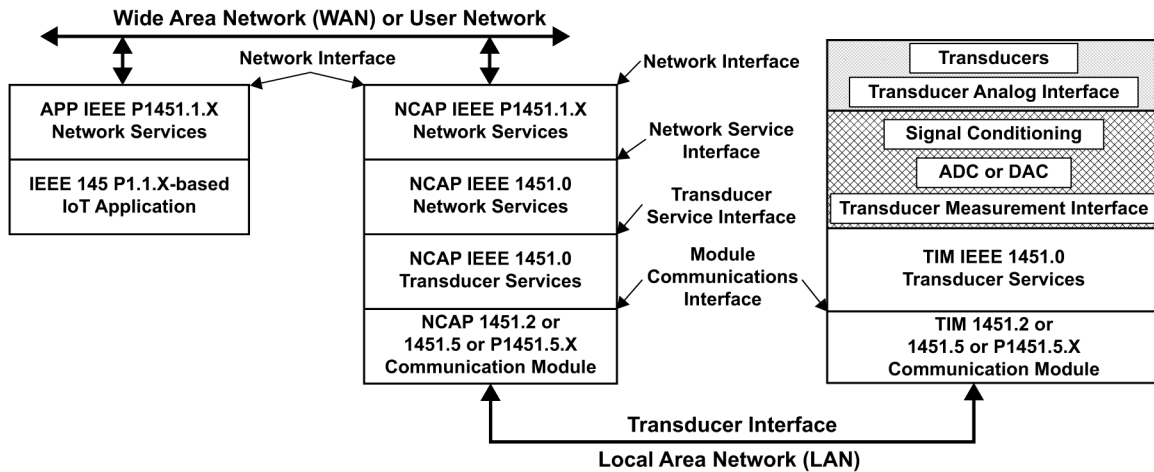


Figure 6—Alternative view of the reference model

5.1.1 APP (1451 Client)

APP consists of the IEEE P1451.1.X network services as a network interface and the IEEE 1451 standard-based IoT applications. The APP should communicate with the NCAP based on the IEEE P1451.1.X network services via a user network or WAN that the user desires; it could be wired or wireless Ethernet/Intranet/Internet/VPN.

NOTE—P1451.1.X could be P1451.1.4, P1451.1.5, or P1451.1.6.

5.1.2 APP network interface

The network interface of APP consists of several network interfaces, including P1451.1.4 network interface, P1451.1.5 network interface, and P1451.1.6 network interface. The APP P1451.1.X network interfaces include P1451.1.X network services that implement IEEE 1451.0 network services (client).

5.1.3 Network physical interface (network PHY)—user network or wide-area network (WAN)

The network physical interface between the NCAP and the APP is left up to the other members of the standards family to define. It is not described in this standard. Functions needed across this interface such as synchronization, if applicable, are described but not their implementation. The network physical interface may include Ethernet, and wireless networks including cellular (e.g., LTE, 5G/6G) networks.

5.1.4 NCAP (1451 Server)

NCAP is a device between the TIMs and the APP. The NCAP performs network communications with the APP via the network interface, network communications with TIM via the transducer interface, and data conversion or other processing functions.

NCAP is a 1451 server that provides network services for APPs, and a client that can access transducer data from TIMs via the transducer interface and services. The NCAP consists of IEEE P1451.1.X network services for P1451.1.X network interfaces, IEEE 1451.0 network services, 1451.0 transducer services for a 1451.X or P1451.5.X transducer interface, and an NCAP 1451.X or P1451.5.X communication module.

5.1.5 NCAP network interface

The NCAP network interface of the user network or WAN consists of several network interfaces, including P1451.1.4 network interface, P1451.1.5 network interface, and P1451.1.6 network interface. The NCAP P1451.1.X network interfaces include P1451.1.X network services that implement the IEEE 1451.0 network services.

5.1.5.1 IEEE P1451.1.4 network interface

This P1451.1.4 standard defines a method for transporting IEEE 1451.0 network services messages over a network using P1451.1.4 network services to establish session initiation, secure communication, and characteristic identification between networked client and server devices using device Meta identification information based on the IEEE 1451.0 TEDS.

5.1.5.2 IEEE P1451.1.5 network interface

This P1451.1.5 standard defines a method for transporting IEEE 1451.0 network services messages over a network using P1451.1.5 network services to establish session initiation, secure communication, and characteristic identification between networked client and server devices using device Meta identification information based on the IEEE 1451.0 TEDS.

5.1.5.3 IEEE P1451.1.6 network interface

This P1451.1.6 standard defines a method for transporting IEEE 1451.0 network services messages over a network using P1451.1.6 network services to establish session initiation, secure communication, and characteristic identification between networked client and server devices using device Meta identification information based on the IEEE 1451.0 TEDS.

5.1.6 NCAP IEEE 1451.0 network services interface

This network services interface is defined in this standard in terms of an application program interface or API. See Clause 10 for the details of this interface.

5.1.7 NCAP IEEE 1451.0 network services

IEEE 1451.0 network services support a set of services (command-and-reply messages) for IEEE 1451 smart transducers to communicate transducer data and TEDS information to and from a WAN or user network via various network communication protocols, such as P1451.1.5, P1451.1.6, and 1451.1.4. IEEE 1451.0 network services could be implemented using HTTP, Web services, and TCP/UDP/IP even though not specified in IEEE Std 1451.0. The purpose of network services is to achieve smart transducer data interoperability in the network interface.

5.1.8 NCAP IEEE 1451.0 transducer services interface

This transducer services interface is defined in this standard in terms of an application program interface or API. See Clause 13 for the details of this interface.

5.1.9 NCAP IEEE 1451.0 transducer services

The transducer service defines the transducer functions and services provided to the NCAP communications module and the NCAP transducer interfaces. These functions and services provided by this communication module are defined in this standard. These functions and services include the command-and-reply set and the TEDS. Features are included in the standard to support capabilities such as triggering and synchronous sampling of sensors, but the detailed implementation is left up to the other standards in the family.

5.1.10 NCAP module communication interface

The module communications interface is defined in this standard in terms of an API that is used to pass information across the interface between the NCAP IEEE 1451.0 transducer services and the NCAP IEEE 1451.X communication module. See Clause 14 for details of this interface. The physical aspects of this interface are not described in the standard. That is left up to the manufacturers.

The NCAP module communication interface consists of several 1451.X or P1451.5.X module communication interfaces. The 1451.X and P1451.5.X transducer interfaces implement and provide IEEE 1451.0 transducer services.

NOTE—

- 1451.X could be 1451.2 or 1451.5.
- P1451.5.X could be P1451.5.5, P1451.5.6, or P1451.5.10.

5.1.10.1 NCAP IEEE 1451.2 serial module communication interface

The NCAP IEEE 1451.2 serial module communications interface is defined in this standard in terms of an API that is used to pass information across the interface between the NCAP IEEE 1451.0 transducer services and the NCAP IEEE 1451.2 communication module. See Clause 14 for details of this interface. The physical aspects of this interface are not described in the standard.

5.1.10.2 NCAP IEEE 1451.5 or P1451.5.X wireless module communication interface

The NCAP IEEE 1451.5 or P1451.5.X wireless module communications interface is defined in this standard in terms of an API that is used to pass information across the interface between the NCAP IEEE 1451.0 transducer services and the NCAP IEEE 1451.5 or P1451.5.X communication module. See Clause 14 for details of this interface. The physical aspects of this interface are not described in the standard.

5.1.11 NCAP communication module

The NCAP communication module includes several 1451.X or P1451.5.X communication modules.

5.1.11.1 NCAP IEEE 1451.2 serial communication module

The NCAP IEEE 1451.2 serial communication module is not defined in this standard but is defined in IEEE Std 1451.2-1997 serial communication module, which provides the low levels of the communications protocol stack and any other services that the NCAP requires to function in the system.

5.1.11.2 NCAP IEEE 1451.5 or P1451.5.X wireless communication module

The NCAP IEEE 1451.5 wireless communication module is not defined in this standard but is defined in IEEE Std 1451.5-2007, which provides the low levels of the communications protocol stack and any other services that the NCAP requires to function in the system.

The NCAP IEEE P1451.5.X wireless communication module is not defined in this standard but is defined in IEEE P1451.5.5, P1451.5.6, and P1451.5.10, respectively, which provides the low levels of the communications protocol stack and any other services that the NCAP requires to function in the system, respectively.

5.1.12 Transducer physical interface (Transducer PHY)—Wireline LAN or Wireless LAN

The transducer physical interface between the NCAP and the TIM is left up to the other members of the standards family to define. It is not described in this standard. Functions needed across this interface such as synchronization, if applicable, are described but not their implementation. The transducer physical interface may include a wireline interface, including 1451.2, or wireless interfaces, including 1451.5, P1451.5.5, P1451.5.6, and P1451.5.10 networks.

5.1.13 Transducer interface module

The TIM consists of TIM IEEE 1451.X or P1451.5.X transducer interfaces and communication modules, which will implement and provide 1451.0 transducer services and messages. The TIM also contains the TEDS, logic to implement the transducer interface, the transducer(s) or connection to the transducer(s), and any signal conversion or signal conditioning. A TIM can contain sensors and/or actuators.

5.1.14 TIM communication module

The TIM communication module includes several 1451.X or P1451.5.X communication modules.

5.1.14.1 TIM IEEE 1451.2 serial communication module

The TIM IEEE 1451.2 serial communication module is logically the complement of the NCAP IEEE 1451.0 and 1451.2 communication module but not identical since some services required in the TIM are different than the services required in the NCAP.

The TIM IEEE 1451.2 serial communication module is not defined in this standard but is defined in IEEE Std 1451.2-1997, which provides the low levels of the communications protocol stack and any other services that the TIM requires to function in the system.

5.1.14.2 TIM IEEE 1451.5 or P1451.5.X wireless communication module

The TIM IEEE 1451.5 or P1451.5.X wireless communication module is logically the complement of the NCAP IEEE 1451.5 or P1451.5.X wireless communication module but not identical since some services required in the TIM are different than the services required in the NCAP.

The TIM IEEE 1451.5 wireless communication module is not defined in this standard but is defined in the IEEE 1451.5-2007, which provides the low levels of the communications protocol stack and any other services that the TIM requires to function in the system.

The TIM IEEE P1451.5.X wireless communication module is not defined in this standard but is defined in the IEEE P1451.5.5, P1451.5.6, and P1451.5.10, respectively, which provides the low levels of the communications protocol stack and any other services that the TIM requires to function in the system.

5.1.15 TIM module communication interface

The TIM module communications interface is defined in this standard in terms of an API that is used to pass information across the interface between the TIM IEEE 1451.0 transducer services and the TIM IEEE 1451.X communication module. See Clause 14 for details of this interface. The physical aspects of this interface are not described in the standard. That is left up to the manufacturers.

The TIM module communication interface consists of several 1451.X or P1451.5.X module communication interfaces. The TIM 1451.X and P1451.5.X transducer interfaces will implement and provide IEEE 1451.0 transducer services.

5.1.15.1 TIM IEEE 1451.2 serial module communication interface

The TIM IEEE 1451.2 module communications interface is defined in this standard in terms of an API that is used to pass information across the interface between the TIM IEEE 1451.0 transducer services and the TIM IEEE 1451.2 communication module. See Clause 14 for details of this interface. The physical aspects of this interface are not described in the standard.

5.1.15.2 TIM IEEE 1451.5 or P1451.5.X wireless module communication interface

The TIM IEEE 1451.5 or P1451.5.X wireless module communications interface is defined in this standard in terms of an API that is used to pass information across the interface between the TIM IEEE 1451.0 transducer services and the TIM IEEE 1451.5 or P1451.5.X communication module. See Clause 14 for details of this interface. The physical aspects of this interface are not described in the standard.

5.1.16 TIM IEEE 1451.0 transducer services

This TIM IEEE 1451.0 transducer services are performed in this function. Most transducer services performed by the NCAP transducer services have counterparts within this block.

5.1.17 Transducer measurement API

This interface is left up to the manufacturer and is outside of the scope of this standard.

5.1.18 Signal conditioner and data conversion functions

The details of the signal conditioning and data conversion functions are outside of the scope of this standard.

5.1.19 Transducer analog interface

This interface is the physical interface between the signal conditioning and data conversion functions and is outside of the scope for all members of the IEEE 1451 family of standards except IEEE Std 1451.4-2004.

5.1.20 Transducer

The transducer is considered outside of the scope by all members of the IEEE 1451 family of standards.

5.1.21 TEDS

The TEDS are defined in this standard along with the access methods for them. However, some information that relates to the physical interface is defined in other standards in the IEEE 1451 family. These TEDS are defined in the other standards. See Clause 6 for detailed descriptions of the TEDS.

5.1.22 The use of IEEE Std 1451.4-2004 with this reference model

The IEEE 1451.0 layer is defined for the transfer of digital information between modules in a system. The IEEE 1451.0 TEDS describe the entire TIM, including the transducer, signal conditioner, and data converters. The analog information that is being described as transported using these interfaces is not covered by any of this family of standards. However, IEEE Std 1451.4-2004 adds a TEDS in parallel with the analog output from a transducer. TEDS give the characteristics of the analog information that may be obtained using an IEEE 1451.4 sensor. The use of IEEE 1451.4 transducers is not required by any other member of the IEEE 1451 family of standards. However, it is allowed as shown in this reference model. Using an IEEE 1451.4 device with an IEEE 1451.0-compatible device requires the appropriate signal conditioner for the transducer and an IEEE 1451.4 TEDS Translator. The TEDS Translator is required to combine the contents of the IEEE 1451.4 TEDS with the characteristics of the signal conditioner to allow the IEEE 1451.0 TEDS to describe the overall TIM. Both sets of TEDS are described in the standards for the family members, but the details of how to combine the TEDS are not covered in any of the standards.

IEEE 1451.4 transducers are commonly used with interfaces that do not comply with the other members of the IEEE 1451 family of standards. In these cases, this reference model does not apply.

5.1.23 The use of IEEE Std 1451.7-2010 with this reference model

The IEEE 1451.0 layer is defined for the transfer of digital information between modules in a system. The IEEE 1451.0 TEDS are used to describe the entire TIM, including the transducer, signal conditioner, and data converters. The analog information that is being described as transported using these interfaces is not covered by any of this family of standards. However, IEEE Std 1451.7-2010 adds a TEDS in parallel with the analog output from a transducer. These TEDS give the characteristics of the analog information that may be obtained using an IEEE 1451.7 transducer. The use of IEEE 1451.7 transducers is not required by any other member of the IEEE 1451 family of standards. However, it is allowed as shown in this reference model. Using an IEEE 1451.7 device with an IEEE 1451.0-compatible device requires the appropriate signal conditioner for the transducer and an IEEE 1451.4 TEDS Translator. The TEDS Translator is required to combine the contents of the IEEE 1451.7 TEDS with the characteristics of the signal conditioner to allow the IEEE 1451.0 TEDS to describe the overall TIM. Both sets of TEDS are described in the standards for the family members, but the details of how to combine the TEDS are not covered in any of the standards.

IEEE 1451.7 transducers are commonly used with interfaces that do not comply with the other members of the IEEE 1451 family of standards. In these cases, this reference model does not apply.

5.2 IEEE 1451 communication models

The IEEE 1451 communications models include two models: client–server and publish–subscribe.

5.2.1 Client–Server communication model

Synchronous and asynchronous client–server models shown in Figure 7(a) are command–reply communication models. The client shall send the command to the server. The server shall receive and process the commands from the client and then send the corresponding replies back to the client. The client–server communication model supports both a synchronous communication model (blocking) without the callback service and an asynchronous communication model (non-blocking) with the callback service.

Figure 7(b) shows synchronous and asynchronous client–server communications between the APP and the NCAP based on network services. APPs (1451 clients) shall connect to and communicate with one NCAP in a synchronous or an asynchronous (optional) way. The NCAP provides a set of network services for the APP to access transducer data and TEDS. The synchronous client–server models shall be applied to the communication between the APP and the NCAP. The asynchronous client–server model with the callback service is optional for client–server communications between APP and NCAP. Please see the asynchronous details in 10.2.

Figure 7(c) shows synchronous and asynchronous client–server communication between the NCAP (client) and TIM (server) based on transducer services. Normally, one TIM/WTIM shall only register to one NCAP (client). One NCAP (client) shall connect and communicate with multiple TIMs/WTIMs that registered to it in a synchronous or asynchronous (optional) way. The asynchronous client–server model with the callback service is optional for client–server communications between NCAP and TIM. Please see the asynchronous details in 13.6.

Figure 7(d) shows synchronous and asynchronous client–server communications between the APP and the NCAP based on network services, and the client–server communication between the NCAP (client) and TIM (server) based on transducer services. The asynchronous client–server communication model with the callback service is optional for communications between APP and NCAP, as well as between NCAP and TIM.

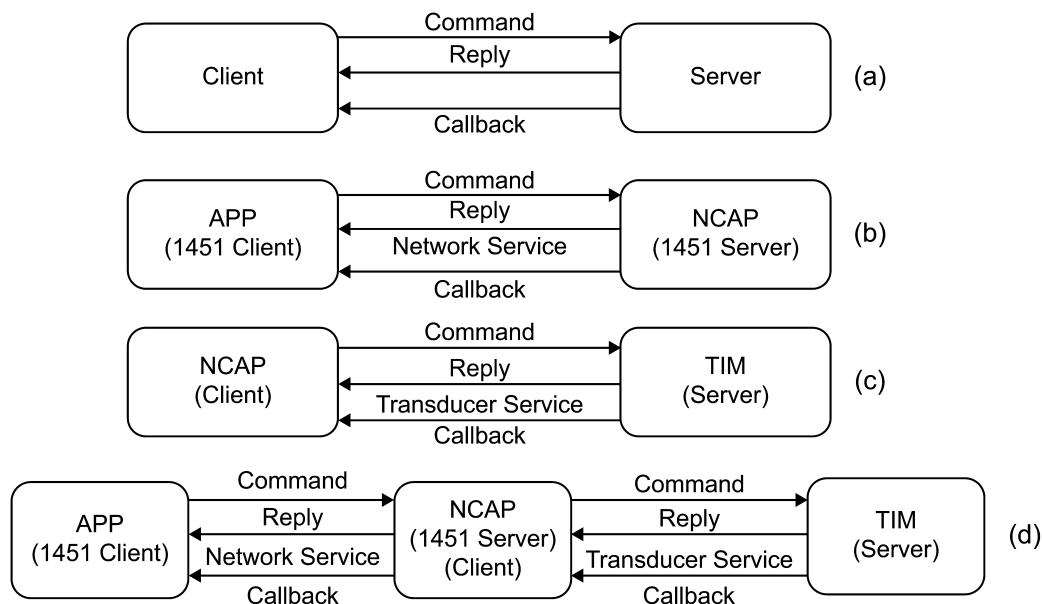


Figure 7—Synchronous and asynchronous client–server communication model

5.2.2 Publish–Subscribe communication model

As shown in Figure 8(a), the publish–subscribe communication model consists of a publisher, multiple subscribers, and service messages. The subscribers shall subscribe the service messages from the publisher. Publish–subscribe could be unicast, multicast, or broadcast. Figure 8(b) shows publish–subscribe between the NCAP and APPs. Figure 8(c) shows publish–subscribe between the TIM and NCAP, which should be applied for transducer services that have only a reply message without the command message, such as the TIM Announcement defined in 11.2.8.2.

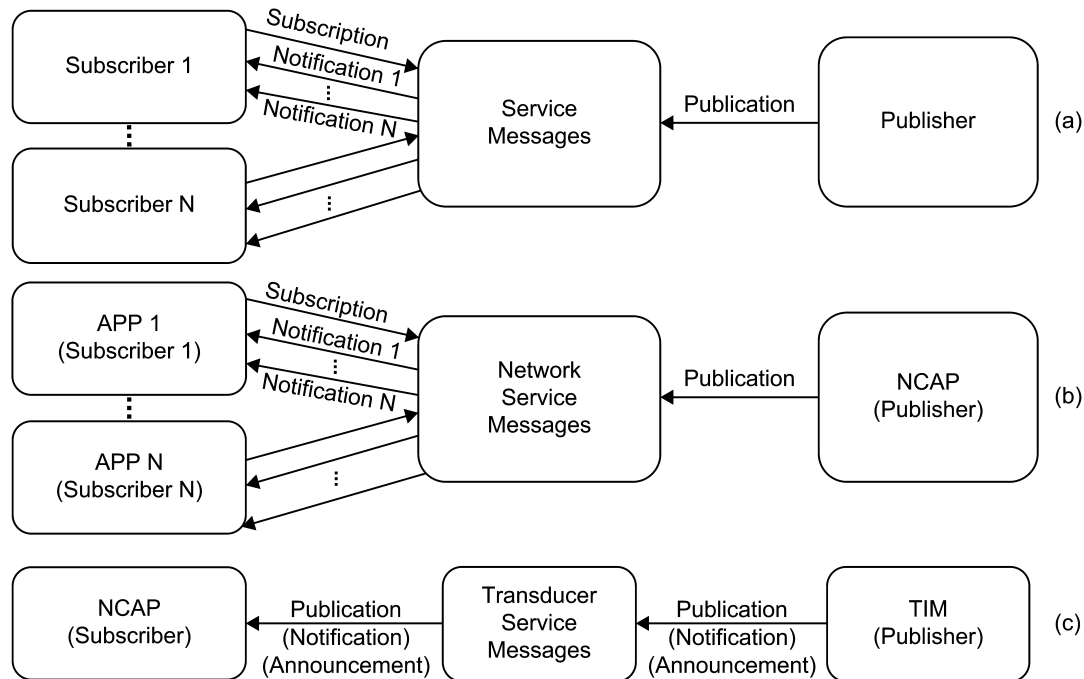


Figure 8—Publish–Subscribe communication model

An event-notification communication shown in Figure 9(a) is one type of publish–subscribe model in which the service messages are events. An event publisher, which is a generator of events, publishes event notifications to those event subscribers that have subscribed to the events. Figure 9(b) shows event publish–subscribe between the NCAP and APPs. Figure 9(c) shows event publish–subscribe between the TIM and NCAP, which should be applied for the transducer event notification service defined in 11.2.7.6.

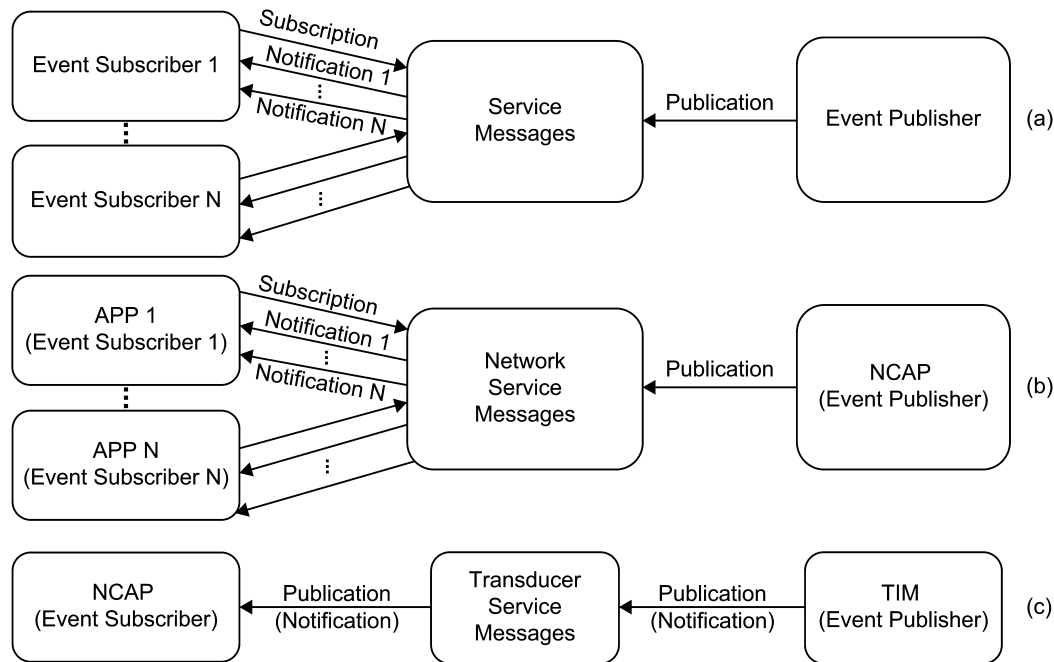


Figure 9—Example of event-notification model

5.3 Plug-and-play capability

A TIM and NCAP that are built to this standard shall be able to be connected using a compatible physical communications media and shall 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 they impact the plug-and-play capability of the TIM or NCAP.

5.4 Addresses

Two levels of addressing are used in this standard. One level of addressing is associated with the physical layer implementation. The details of this address are covered in the members of the IEEE 1451 family that define the communications media. The module communications API links this address to the “destId” with the discoverDestination call described in 14.3.1. This level of addressing allows messages (see Clause 10) to be sent between a TIM and an NCAP. For more details about this address, see the standard for the physical layer communications.

The second level of addressing is the TransducerChannel number for a given TransducerChannel within the TIM. These 16-bit numbers are used in command messages as described in 11.1.3. As the Destination TransducerChannel number or in reply messages (see 11.1.4) as a source TransducerChannel number. This level is used to specify to the TIM how the message should be directed or to tell the NCAP where it came from within the TIM. Rules are associated with the interpretation of a TransducerChannel number that are given in Table 4. 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 Clause 11 along with the definition of the commands.

Table 4—Rules for TransducerChannel number usage

Address class	Value	Description
Global	0xFFFF	Global addressing is a special GroupAddress that pertains to all TransducerChannels on the TIM.
AddressGroup	0x8000<A≤0xFFFE	There are two types of AddressGroups as defined in Table 5. The Bit Mapped AddressGroups are used when a few AddressGroups are needed and it is desirable to send a command to multiple AddressGroups at the same time. Binary AddressGroup are used when a large number of different AddressGroups are required. Address 0x8000 is a nonfunctional address.
TransducerChannel	1≤A≤0x7FFF	An address with the most significant bit set to zero. The remaining 15 bits identify the TransducerChannel for which the message is intended.
TIM	0	An address within the message of zero indicates that the message is intended for the TIM and not an individual TransducerChannel.

Table 5—Address groups

Address class	Value	Description
Bit Mapped	0x8000<A≤0xBFFF	The most significant bit is set to one, the next most significant bit is zero, and at least one other bit is set to a one. The pattern in the two most significant bit is set to indicate that this is a group address. The other bit or bits identify the group(s).
Binary	0xC000≤A≤0xFFFE	The two most significant bits are set to one, and the remaining 14 bits form a binary pattern identifying a single address group.

5.4.1 Global addresses

Commands addressed to the Global address shall be received and honored by all TransducerChannels on the addressed TIM(s). If the received command is not implemented by a TIM or TransducerChannel, the command shall be ignored, and no error shall be generated.

5.4.2 AddressGroup addresses

Commands issued to an AddressGroup shall be honored by all TransducerChannels that have been initialized as members of that AddressGroup. See the AddressGroup definition command in 11.2.3.3 for how to assign or clear an AddressGroup assignment. If the received command is not implemented by a TIM or TransducerChannel, the command shall be ignored and the command rejected bit shall be set in the TIM or TransducerChannel.

The use of AddressGroups across multiple TIMs requires that a method of addressing multiple TIMs be established before TransducerChannels can be assigned to AddressGroups shown in Table 5. See 13.1.3 and 13.2.4 for the details of the method required to make these assignments.

5.4.2.1 Bit-mapped AddressGroup addresses

Each AddressGroup is represented by having the most significant bit in the AddressGroup assignment set to one, the next most significant bit set to zero, and one other bit set in the TransducerChannel number field of the address class. The one, zero pattern in the two most significant bits of the TransducerChannel number field indicates that the address is a bit-mapped group address. Using bitwise OR logic, one command may be issued to multiple AddressGroups simultaneously. This allows for 14 different AddressGroups.

5.4.2.2 Binary AddressGroup addresses

Each AddressGroup is represented by having the two most significant bits in the AddressGroup assignment set to one. The remaining 14 bits in the TransducerChannel number field of the address class identify the AddressGroup. This allows for 16 383 different AddressGroups.

5.4.3 TransducerChannel number

Command-and-reply messages that have a number between one and 0x7FFF, inclusive, in the TransducerChannel number field are addressed to a TransducerChannel. They shall be received and honored by the TransducerChannel to which they are addressed. If the received command is not implemented by the TransducerChannel, the command shall be ignored, and an error shall be generated as specified for the individual command.

5.4.4 TIM addresses

Command-and-reply messages that have zero in the TransducerChannel number field are addressed to a TIM. They shall be received and honored by the TIM to which it is addressed by the “destId.” They are intended for the interface module and not for any particular TransducerChannel within the TIM. If the received command is not implemented by a TIM, the command shall be ignored, and an error shall be generated as specified for the individual command.

5.5 Common characteristics

Subclauses 5.5.1 through 5.5.4 define the operating characteristics that are common to all TransducerChannel types.

5.5.1 Operating states

A high-level state transition diagram for a TransducerChannel is provided in Figure 10. There are two basic operating states for a TransducerChannel after it comes out of initialization. The TransducerChannel enters the “Transducer Idle” state when it comes out of initialization. A TransducerChannel accepts most commands when it is in the Transducer Idle state. A TransducerChannel enters the Transducer Operating state upon receipt of a TransducerChannel Operate (see Table 92) command and remains in this state until the TransducerChannel Idle (see 11.2.5.2) command is received.

As shown in Figure 11, a TIM has three states. A TIM is placed in the TIM Initialization state by a reset (see 11.2.8.1) command or by a power-up event. Once it completes the initialization process, it transitions to the TIM Active state. A TIM may be placed in the TIM Sleep state by the TIM Sleep (see 11.2.7.2) command. The only command that a TIM shall accept when in the TIM Sleep state is the Wake-up (see 11.2.6.1) command that returns it to the TIM Active 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 11.

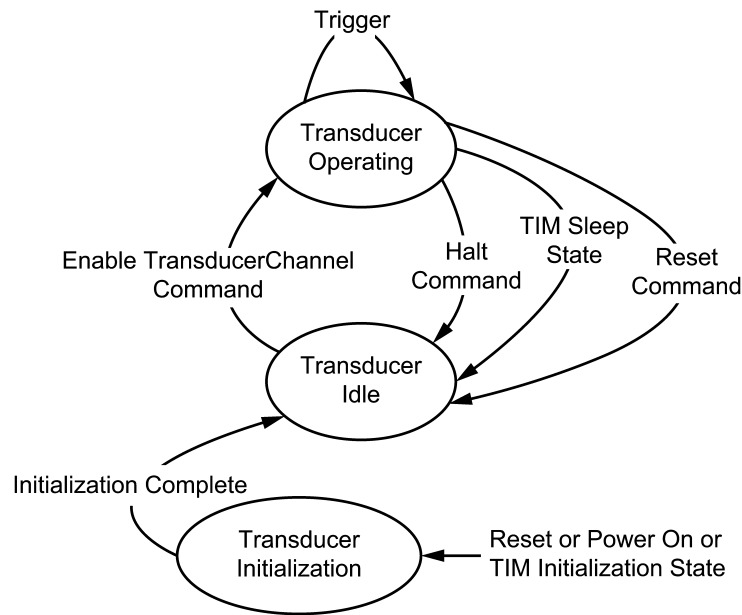


Figure 10—TransducerChannel operating states

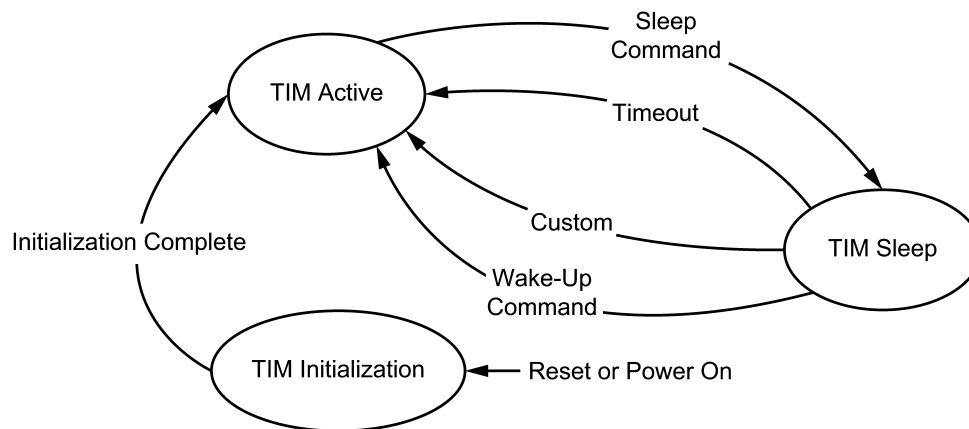


Figure 11—TIM operating states

5.5.2 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.

5.5.3 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 no requirements are imposed on the scope of the diagnostic logic.