INTERNATIONAL STANDARD

ISO/IEC 24753

First edition
2011-09-01

Information technology — Radio frequency identification (RFID) for item management — Application protocol: encoding and processing rules for sensors and batteries

Technologies de l'information — Identification par radiofréquence (RFID) pour gestion d'objets — Protocole d'application: règles de codage et de traitement pour capteurs et batteries

Reference number
ISO/IEC 24753:2011(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 24753 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

Introduction

The technology of radio frequency identification (RFID) is based on non-contact electronic communication across an air interface. The structure of the bits stored in the memory of the RFID tag is invisible and accessible between the RFID tag and the interrogator only by the use of the appropriate air interface protocol, as specified in the corresponding part of ISO/IEC 18000. Since the initial publication of ISO/IEC 18000, it has become possible to add sensors to the RFID tag using various physical methods, but always using the air interface protocol as a consistent means of communicating between the RFID tag and the interrogator.

For sensor information, functional commands from the application and responses from the interrogator are processed in a standard way. This allows equipment to be interoperable and, in the special case of the sensor attached to or integrated within an RFID tag, enables configuration parameters to be encoded in one system's implementation with the resultant sensory information to be read at a later time in a completely different and unknown system's implementation. The data bits stored on each RFID tag and sensor must be formatted in such a way as to be reliably read at the point of use if the sensor is to fulfil its basic objective. The integrity of this is achieved through the use of an application protocol, for example as supported by the functional commands specified in ISO/IEC 15961 and as specified in ISO/IEC 24791.

Manufacturers of radio frequency identification equipment (interrogators, RFID tags, etc.), manufacturers of sensors, and users of RFID technology supporting sensors each require a publicly available application protocol. This International Standard specifies the sensor encoding and processing rules, which are independent of any of the air interface standards defined in the various parts of ISO/IEC 18000. As such, the sensor encoding and processing rules are consistent components in the RFID system that can, independently, evolve to support additional air interface protocols and different types of sensors.

This International Standard specifies the overall process and methodologies developed to format and process sensory information in a standardized manner and provide an interface with the appropriate air interface protocol.

The transfer of sensory information and other related data to and from the application is supported by the use of the object identifiers standard, as defined in this International Standard.

Information technology — Radio frequency identification (RFID) for item management — Application protocol: encoding and processing rules for sensors and batteries

1 Scope

This International Standard defines a minimum application protocol to support sensors and the monitoring of batteries in conjunction with RFID tags utilizing the air interface as defined in the ISO/IEC 18000 series.

This application protocol for sensors applies to RFID tags irrespective of their operating frequency. This application protocol is agnostic to how the sensor(s) are connected to or integrated within the RFID tag; however, the communication between the interrogator and the sensor(s) is always through the RFID tag. This will allow the interrogator and application to understand a compliant sensor's characteristics and process its information without prior knowledge of that sensor. This will allow sensors to announce their sense activity and the units of measurement to the interrogator.

This International Standard provides common encoding rules for identifying sensors, their functions, and their delivered measurements. It also defines the process rules to support the following functions:

— selecting and de-selecting a particular sensory function when more than one is supported by the RFID tag;

NOTE The measurement of time or battery life can be considered as separate sensory functions.

— setting sensor parameters both initially and ongoing;

— starting and stopping the monitoring function of a sensor;

— accessing sensor data; and

— carrying out basic processing of sensor data and interpreting this into a format that is meaningful for an application.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 18000-6, *Information technology — Radio frequency identification for item management — Part 6: Parameters for air interface communications at 860 MHz to 960 MHz*

ISO/IEC 19762-1, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 1: General terms relating to AIDC*

ISO/IEC 19762-3, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 3: Radio frequency identification (RFID)*

IEEE 1451.7, *Standard for Smart Transducer Interface for Sensors and Actuators — Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer Electronic Data Sheet Formats*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762-1, ISO/IEC 19762-3 and the following apply.

3.1.1
calibration
process used to determine the information that resides in the Calibration TEDS to support correction

3.1.2
data-set
collection of samples acquired by a sensor (or applied by an actuator) in response to a trigger command

3.1.3
event sensor
sensor that detects a state change in the physical world

NOTE The fact that a change of state has occurred and/or the instant in time of the change of state, not the state value, is the “measurement”.

3.1.4
Network Capable Application Processor
NCAP
device between the transducer modules and the network for an IEEE 1451 sensor

NOTE The NCAP performs network communications, communications, and data conversion or other processing functions.

3.1.5
Sensor Address Map
SAM
defined memory area of an RFID tag that specifies the logical location of the sensor memory

3.1.6
sensor driver
mechanism to transfer sensor-related data between the implementation of ISO/IEC 24753 and the RFID tag

3.1.7
smart transducer
transducer that provides functions beyond those necessary for generating a correct representation of a sensed or controlled quantity

NOTE This functionality typically simplifies the integration of the transducer into applications in a networked environment.

3.1.8
timestamp
unambiguous representation of some instant in time

3.1.9
transducer
sensor or actuator device that converts energy from one domain into another

3.1.10
Transducer Electronic Data Sheet
TEDS
electronic data sheet describing a transducer

3.1.11
trigger
signal or command that is used to start an action

3.2 Abbreviations

ASIC	Application-specific Integrated circuit
LSB	Least Significant Bit
MSB	Most Significant Bit
NCAP	Network Capable Application Processor
RTC	Real Time Clock
SAM	Sensor Address Map
SI	international system of units, reference *The International System of Units (SI)*
TEDS	Transducer Electronic Data Sheet
UTC	Co-ordinated Universal Time

4 Conformance

To claim conformance with this International Standard, an implementation shall support one or both of the following:

a) All the processes that are required to support all aspects of full function sensors for configuration and interpretation of sensor data.

b) All the processes that are required to support all aspects of simple sensors for configuration and interpretation of sensor data.

5 Basic Model

5.1 Logical interface model

5.1.1 General

The processes defined in this International Standard are implemented between the application and the air interface protocol. This International Standard performs similar functions for sensory data as ISO/IEC 15962 does for item-related data. The relationship and basic functions of the standards are illustrated in Figure 1 — Basic application interface model.

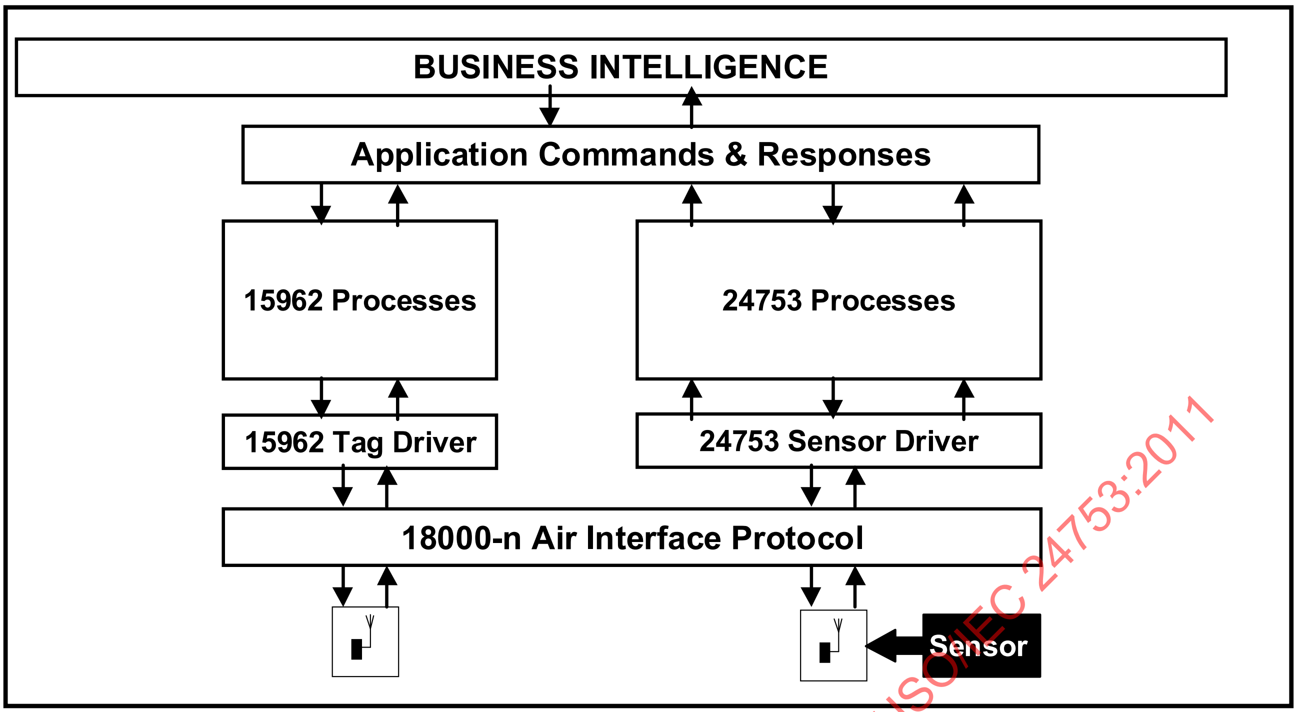

Figure 1 — Basic application interface model

Each of the component parts of the model relevant to sensors and batteries is described below.

5.1.2 Functional application commands

A set of functional application commands are required to enable the application to identify what sensor functions are supported, to access data from sensors, to access the status of the battery power, and to reset values such as alarm values for the sensor activity. The structure of these commands may be derived from the set of object identifiers used, or from future standards as these are developed.

5.1.3 ISO/IEC 24753 processes

This International Standard specifies a set of encoding and processing rules that address the following:

— Reading sensor identifiers

— Reading sensor characteristics and converting this from the bit based encoding on the sensor to a format more appropriate for business processing

— Reading (and encoding when permitted by the sensor) configuration parameters

— Reading event data captured by the sensor and converting this from the bit based encoding on the sensor to a format more appropriate for business processing

— Reading alarm data provided by the sensor

These processes are applied in different manners depending on the whether the sensor is a full function sensor (as specified in IEEE 1451.7) or a simple sensor (as specified in ISO/IEC 18000-6).

5.1.4 Sensor driver

The sensor driver provides two main functions:

— It provides the rules for identifying how a particular air interface protocol supports sensors. Typically, this would be one of the following: the sensor is integrated within the RFID ASIC or attached through a serial or parallel port, or internal analogue to digital conversion incorporated into the RFID ASIC. A sensor driver may define that an air interface supports one of these types, or if more than one option is possible, it will need to clearly differentiate the type of interface between the sensor and the RFID tag.

— It provides facilities that accept the application commands and converts them to a format that results in calls to command codes supported by the particular RFID tag. In some instances, the air interface protocol might have a specific command such as "read port n" whereas in other cases an air interface command might simply act as a transport mechanism to embed command understood by the sensor.

The description of the sensor drivers for particular RFID tags is provided in Annex A. The sensor driver is unique to the particular air interface type of RFID tag as specified in the appropriate part of ISO/IEC 18000. This is a logical representation, and a physical implementation could combine features of different logical tag drivers. The interrogator may support one or many sensor drivers.

Generally speaking, the sensor driver will be logically independent from the tag driver, as specified in ISO/IEC 15962, but physical implementations could combine the functionality of the tag driver and sensor driver.

5.1.5 ISO/IEC 18000 air interface functions

This International Standard can only be prescriptive on how the air interface functions insofar as the sensor driver needs to represent the capabilities of the air interface protocols. However, some attempts at standardisation can prove useful. Such standardisation can include having reasonably consistent:

— command structures

— response structures

— rules for defining ports or memory mapped locations

The prime requirement for consistency is with respect to the sensor and also between the application and this International Standard. So, any variation on the common approach for a particular air interface should be addressable through the rules of the sensor driver. As air interface protocols are developed to support sensors and batteries, consideration should be given to the manner that previous implementations have been incorporated into ISO/IEC 18000 standards.

5.1.6 The sensor

Full function sensors supported by this International Standard shall be compliant with IEEE 1451.7.

Simple sensors supported by this International Standard shall be as specified in ISO/IEC 18000-6.

In either case, the focus for this International Standard is on the data encoded about the sensor or generated by the sensor during its monitoring process. The functional processes of the sensor or the interface or integration method with the RFID tag component are beyond the scope of this International Standard.

No constraints are placed on what environmental features are monitored because this is a matter for sensor design to meet market requirements, with the exception that the unit of measure shall be basic SI units, derived SI units, or integer values (for example for counts). This open-ended approach enables this International Standard to maintain similar processes for sensor types that have yet to be developed, or to require minimal changes to support an increased variety of sensor capabilities.

5.2 The sensor information model for full function sensors

5.2.1 General

The sensor information model (Figure 2 — Sensor information model for full function sensors) shows the relationship between component processes and structures described later in this International Standard for full function sensors specified in IEEE 1451.7. A physical sensor is defined as one that monitors a particular environmental feature capable of being expressed in terms of an SI unit or derived SI unit. A given physical sensor may support a number of logical sensors, each of which specifies a method of event data output, e.g. maximum value, observed value below a threshold qualified by a timestamp, count of events observed that are above a threshold.

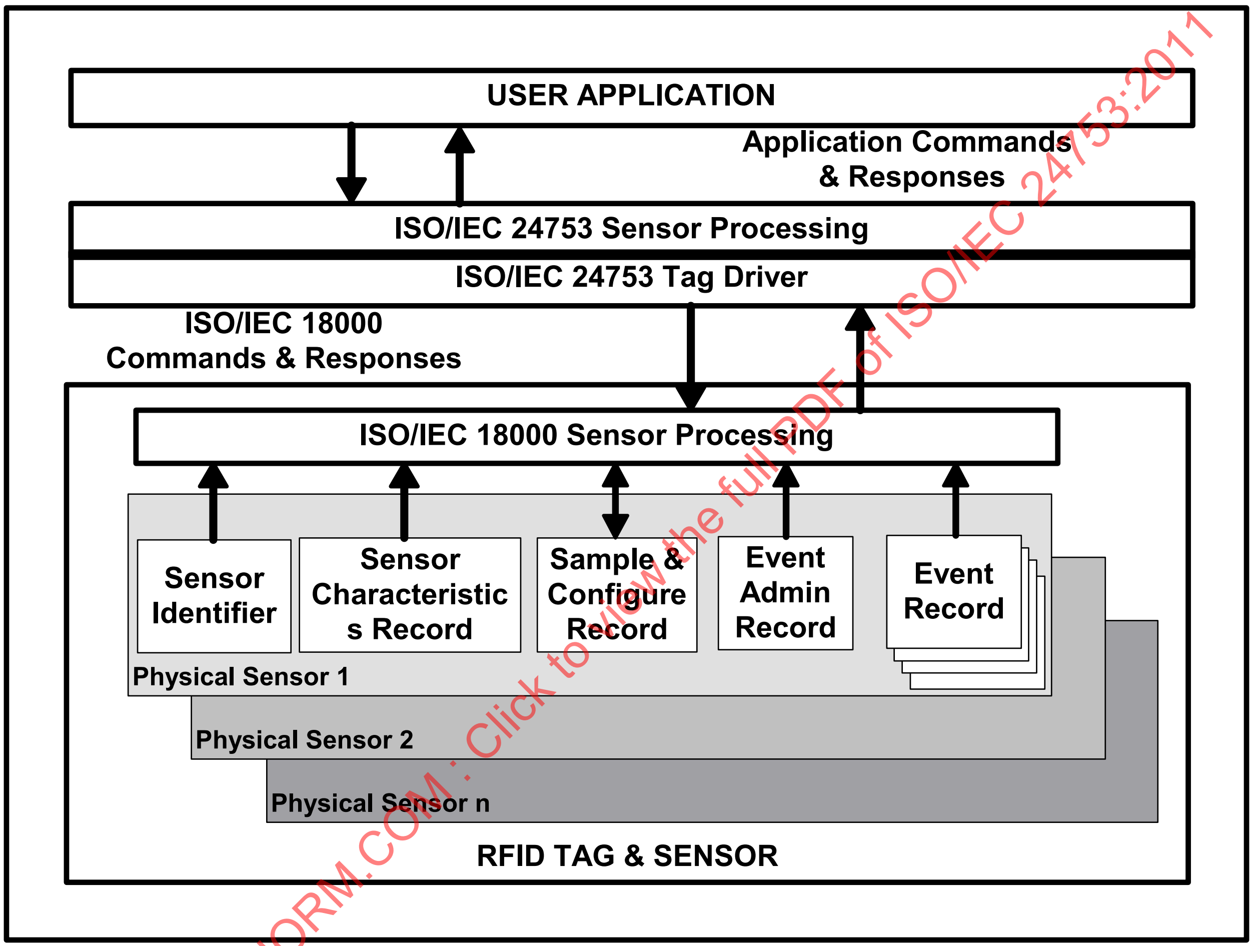
USER APPLICATION
Application Commands & Responses
ISO/IEC 24753 Sensor Processing
ISO/IEC 24753 Tag Driver
ISO/IEC 18000 Commands & Responses
ISO/IEC 18000 Sensor Processing
Sensor Identifier
Sensor Characteristics Record
Sample & Configure Record
Event Admin Record
Event Record
Physical Sensor 1
Physical Sensor 2
Physical Sensor n
RFID TAG & SENSOR

Figure 2 — Sensor information model for full function sensors

5.2.2 Sensor identifier

The sensor identifier enables the sensor manufacturer to uniquely identify a sensor to provide information about the sensor that is additional to that encoded in the sensor characteristic record (e.g. batch number, date of manufacturer, and any other feature that is potentially relevant to the unique sensor. The sensor identifier also provides a mechanism for RFID communications that may be used if more than one sensor can be incorporated on the RFID tag, or if sensor functions can be changed (e.g. through a port connection).

5.2.3 Sensor characteristics

There is a fundamental requirement to be able to identify the sensor characteristics in a consistent manner on the RFID tag, through the processes of this International Standard, and in turn from IEEE 1451.7 for full function sensors so that the application receives meaningful information. This requirement has to apply irrespective of air interface protocol and RFID tag architecture, how the sensory function is incorporated with the RFID tag, and even irrespective of the capabilities of the sensors. This does not mean that all encoding and processes have to be achieved in exactly the same manner, just that there shall be no ambiguity with the processing of sensor characteristics.

This International Standard converts the code structures specified in IEEE 1451.7 for the type of sensor, additional attribute features such as whether the sensor is recording a minimum value, a maximum value or an average value. In addition, all of the characteristics of the sensor are defined in a manner that enables their information to be processed unambiguously, so that the outputs from a temperature sensor cannot be confused with the outputs of a strain sensor.

5.2.4 Sampling and configuration

The capability of sensors will differ considerably from the simplest that only outputs a single item of data, through to more complicated devices where threshold values can be set to trigger monitoring of event data, or the delivery of an alarm. To enable sensors with different data capture and processing capabilities to be supported, similar configuration rules are applied in IEEE 1451.7, but where specific parameters are not relevant, they can be encoded in a manner that the sensor can ignore.

In its turn, this International Standard has process rules that enable the configuration parameters to be converted to an application level interpretation, or for the inverse procedure to be used to set configuration parameters on the sensor.

To ensure that the sampling and configuration values remain set by an authorised user, the air interface protocols may impose additional security features so that only those with permission to change the configuration in the sampling may do so.

5.2.5 Event administration record

The event records can vary from a simple single record to multiple records. In addition for the multiple records, the sensor can provide the option of rollover where the earliest records get overwritten when the memory becomes full. To manage all this, the sensor holds an event administration record that contains information about the size of memory and the number of sensor words that can be stored on the memory. It also contains information that enables the reconstruction of the observation history.

5.2.6 Event record

The event records encode the digital output from the sensor in a binary format specified in IEEE 1451.7. Each physical sensor (e.g. temperature °C, relative humidity %) supports one or more logical sensors that provide the application with a particular form of event record (e.g. current value, maximum value, data log of observations above a threshold, count of observations below a threshold). The flexibility of the logical memory structure enables more rapid access to some data with a more detailed subsequent RFID transaction for detailed data.

5.3 The sensor information model for simple sensors

5.3.1 General

A simple sensor provides limited functional support to determine whether the temperature or other environmental conditions have gone outside some allowable limits. These sensors are defined as factory programmed, which restricts parameter setting from a fully open systems application, but allows data to be captured using open system air interface commands and processes.

5.3.2 Simple sensor data block

The prime operating mode of a simple sensor is to provide the simple sensor data block using some delivery mechanism defined by the air protocol interface. The simple sensor data block is a short bit-based code that provides sensor characteristics, configuration and alarm data.

5.3.3 Other simple sensor memory

Implementation of the simple sensor may use additional memory as defined in 8.3, which specifies the structure of different record blocks. Access to this additional memory is through commands as defined in ISO/IEC 18000-6.

6 Real time clock (RTC)

6.1 General requirements

The RFID tag shall have an RTC and it shall be used as the source of UTC timestamps for sensor related data. Timestamps shall be based on the UTC time epoch beginning at 1970-01-01 00:00:00. At configuration of a sensor, the RTC shall be set to the current 32-bit UTC time precise to 1 second.

6.2 Presentation of time to the application

All time stamps shall be presented to the application in the UTC format of:

year-month-day hour:minute:second

More precisely the format defined by RFC 3339 is presented as yyyy-mm-ddThh:mm:ssZ.

6.3 Encoding of the time stamp

All time stamps shall be stored on the RFID tag as a 32-bit value precise to 1 second.

NOTE This time stamp can be achieved by taking the most significant 32-bit from the IEEE 1588 synchronised time stamp.

The number of 32 bit time stamps that are required to be stored on the RFID tag are kept to a minimum by using the ordinal number of the sample to reduce the use of memory and air interface transmission.

6.4 Converting between the two time presentations

During the decode process, the 32-bit timestamp needs to be converted into the UTC format to provide better human understanding for interpreting an observation at particular points in time. As a number of computer program languages have mechanisms to carry out such a conversion, specifying a detailed algorithm is beyond the scope of this International Standard. However, there are two points of guidance that are provided to assist with implementing this conversion:

— Different program languages use different epochs, and it essential that the epoch for this International Standard begins on 1st January 1970. If the program language uses a different epoch, then an offset value will need to be used to correct the output UTC date and time.

— Although not critical to overall precision, different conversion processes either take into account the leap second or ignore the leap second. Annex B.1 provides information about leap seconds, and sources of information for these.

Annex B.2 also includes references to internet-based conversion calculators that can be used to benchmark any program output, but obviously cannot be incorporated within the process of this International Standard.

For encoding, there are two options available. The first is to apply an inverse algorithm to take the UTC date and time and convert it to a 32-bit value. The second is to access this 32-bit value directly from a network service. Some services compliant with IEEE 1588 can deliver a 64-bit time stamp, where the MSB 32 bits identify time to a second. Services compliant with the Internet Simple Network Time Protocol, as defined in IETF RFC 1769, might also be able to provide the time as a 32-bit value that can be directly used. If encoders are part of a network that has access to the Internet, then time stamps (for example for configuration) can be delivered via the network rather than through the command structure. The choice of providing a program conversion so that UTC is addressed through the application commands, or using a network time based on the correct epoch, is a matter of implementation.

6.5 Setting the RTC

6.5.1 General

The current 32-bit UTC timestamp shall be transmitted across the air interface between the interrogator and the RFID tag to set the RTC during sensor configuration and at other times to maintain time synchronisation.

The 32-bit UTC timestamp may be taken from the interrogator if it supports UTC time as specified in this clause. Timestamps that are not in seconds and of a 32-bit format are not suitable. Timestamps that are based on local time are not acceptable and shall not be used.

If the interrogator is not capable of providing a timestamp in the specified format, then this should be provided using some accurate network-based UTC time.

The RFID tag shall implement a method to determine the memory address of the RTC.

6.5.2 Write-RTC command

This command is intended to be used to set the RTC by writing the current UTC time into the RTC within the RFID tag memory. If the air interface protocol supports it, this command may be encapsulated in an air interface broadcast command, which simply sends out a UTC timestamp to all RFID tags having an RTC within the reading zone. As the RTC has to be integrated with the tag compliant with a particular air interface, it should be possible to implement the RTC such that a response is not necessary to a broadcast command.

Write operations to the RTC may be restricted by the use of memory locking, permalocking, password authorisation, or other means implemented by the RFID tag. It shall not be permitted to write a UTC time of zero into the RTC nor set the RTC when an alarm condition is present other than the low battery alarm.

6.5.3 Read-RTC command

This command is intended to be used to read the current time of the RTC within the RFID tag memory. The current time may be obtained by directly reading from the RTC address. Read operations to the current time may be restricted by the use of memory locking, permalocking, password authorisation, or other means implemented by the RFID tag.

6.6 Time synchronisation

6.6.1 General

The RTC mechanism on an RFID tag might not be able to maintain an accurate record of time. This is partly a factor of the cost of providing additional accuracy and the impact on temperature variation to which the RTC is exposed. If the RFID tag implements sensor event records, then it shall also implement a time synchronisation record block to enable the application to reconstruct a more accurate time line.

The time synchronisation record block enables a UTC timestamp to be recorded against the most recent sample count value. Each recorded time synchronisation record shall comprise, in sequence:

— The prevailing time as recorded by the RFID tag. This may be a 32-bit RTC value maintained since the time of configuration, or the 16-bit sample count value. The 32-bit structure is recommended where time accuracy is important or where there is a large sample interval.

— The 32-bit UTC timestamp.

The number of time synchronisation records contained within the time synchronisation record block shall be vendor defined and consistent with the vendor defined timestamp accuracy at the application level. The timestamp accuracy at the application level shall be achieved provided that a sufficient time synchronisation update rate is maintained. The time synchronisation update rate should define a maximum time interval between updates necessary to maintain timestamp accuracy as well as a minimum time interval between updates necessary to generate a time synchronisation record. When the allocated memory for the time synchronisation block becomes full no further records are possible, but no error shall be indicated.

6.6.2 Reconciling UTC with the 32-bit RTC time

As both the RTC and UTC timestamps have the same common datum, the UTC timestamp at configuration, it is possible to determine the extent of the difference in time at each point when time synchronisation took place. The extent of any difference might vary over the sample period and because of different causes, the RTC could be 'fast' or 'slow' without any pattern. Any time correction can only be applied more accurately close to the time of synchronisation, and be less certain at a point midway between two time synchronisation events.

6.6.3 Reconciling UTC with the 16-bit sample count

In this simpler approach, the best level of synchronisation that can be achieved is to write the UTC timestamp against the most recent sample count value. This means that time synchronisation is also a factor of the size of the sample interval. The larger the sample interval, the more likely that the UTC timestamp will represent a time that is between the most recently recorded sample count value and the projected next (future) sample count value. This provides a simple indication:

— that the RTC and UTC are within the same time period as determined by the sample interval, or

— if the UTC indicates an earlier time than the sample count, then the RTC is 'fast', or

— if the UTC indicates a later time than sample count + 1, then the RTC is 'slow'.

It is only when the synchronisation is at these outer limits that any inaccuracy can be corrected.

7 Full function sensors

7.1 General

Full function sensors shall be compliant with IEEE 1451.7 Type 1. They contain five records:

1) Sensor identifier
2) Sensor characteristics record (Type 1)
3) Sampling and configuration record
4) Event administration record
5) Event records

The numbers in the list are used as record references for full function sensors in this International Standard. Each record is described in the following sub-clauses.

7.2 Sensor identifier

Each IEEE 1451.7 sensor is identified with a unique 64-bit identifier. This value is encoded and locked by the sensor manufacturer. It has two potential functions with respect to this International Standard:

— The unique sensor identifier may be used by an air interface protocol sensor driver as a means of precisely identifying a particular physical sensor.

— With supportive middleware access to the Internet, it can provide additional information about the sensor that is not possible to convey with the sensor characteristics record. For example, a unique sensor identifier may convey information about batch number, manufacturing data and further details for interpreting the data uncertainty and remaining battery life. A procedure to resolve these details is currently beyond the scope of this International Standard.

7.3 Sensor characteristics record (Type 1)

The sensor characteristics record, or more properly called the Transducer Electronic Data Sheet (TEDS) in IEEE 1451, defines all the capabilities of the sensor and this record is permanently encoded on the sensor by the sensor manufacturer. This International Standard only supports IEEE 1451.7 Type 1 sensors; it might need to be amended if a new Type is introduced.

This record is 128 bits long, and provides all the basic characteristics necessary to define the sensor functionality. The first 111 bits of the Type 1 sensor characteristics record are defined in the first edition of IEEE 1451.7. As any of the reserved 17 bits are assigned a function, this International Standard might need to be amended. The sensor characteristics record comprises 22 fields, which are identified by their ordinal number in this International Standard:

1	TEDS type
2	Sensor Type
3	Units extension
4	Sensor map
5	Data resolution
6	Scale Factor Significand
7	Scale Factor Exponent
8	Scale Offset Significand
9	Scale Offset Exponent
10	Data uncertainty
11	Sensor Reconfiguration Capability
12	Memory Rollover Capability
13	Air Interface Security Capability Code
14	Sensor Security Capability Code
15	Sensor Authentication Encryption Capability Map
16	Sensor Data Encryption Capability Map
17	Sensor Authentication Password/Key Size
18	Sensor Data Encryption Key Size
19	Random Number Sizes Supported
20	Continuing Authentication Capability Field
21	Data Encryption Capability Field
22	Clock Accuracy
23	RFU

Procedures to encode and decode the sample and configuration record are specified in 10.3.

7.4 Sampling and configuration record

The IEEE 1451.7 sampling and configuration record is a re-writable record encoded on the sensor. Permission to make changes may be restricted by invoking passwords and other features supported by the air interface protocol standards and the application commands.

The sampling and configuration record comprises of the following 13 fields, which are identified by their ordinal number:

1 UTC timestamp at configuration, or upon successfully beginning a mission.
2 Sample interval
3 Monitor delay
4 Alarm values set
5 Memory rollover enabled
6 Air Interface Security Function Code
7 Sensor Security Function Code
8 Sensor Authentication Encryption Function Code
9 Sensor Data Encryption Function Code
10 Security Timer Duration
11 Begin-End-Mission Authority
12 Upper alarm threshold
13 Lower alarm threshold

This record is of various pre-determined lengths depending on whether none, one, or both alarm levels are set.

Procedures to encode and decode the sample and configuration record are specified in 10.4.

7.5 Event administration record

The event administration record is a read-only record that identifies, for those multiple sample measurement types, the capacity of those event records and other associated parameters.

The event administration record comprises of the following 12 fields, which are identified by their ordinal number:

1 Code 10 Sample capacity
2 Code 11 Sample capacity
3 Code 12 Sample capacity
4 Code 13 Sample capacity
5 Sample count
6 Alarm triggered
7 Sample count at predetermined time
8 Sample count at critical event
9 Sample count of events outside either threshold
10 Sample count at the first threshold event
11 Password/key(s) Read-Lock and Write-Lock status flags
12 Mission in Progress

Procedures to encode and decode the event administration record are specified in 10.5.

7.6 Event records

The event records supported by a particular physical sensor are declared by bit values on the logical sensor map field of the sensor characteristics record. If a bit position is 1_2, then that measurement code is supported and used as the basis for the event record. Any measurement type is addressable by using a 4-bit code as defined in 7.3.

Procedures to encode and decode the event record are specified in 10.6.

8 Simple sensors

8.1 General

This clause specifies the basic functionality, record structures, sensor commands, and processes for simple sensors. Because of their intended functionality, their integrated design with the RFID ASIC and their sensitivity to power consumption, form-factor and cost, only a limited number of simple sensor types are specified. The basic output is a simple sensor data block, where a 32-bit string (or possibly longer) declares information about a simple sensor and reports basic sensor observations using a minimum number of data bits. Simple sensors include:

— Temperature sensor having a high and/or a low out-of-range temperature alarm.

— Relative Humidity sensor having a high and/or a low out-of-range relative humidity alarm.

— Impact sensor with an instance of_out-of-range impact alarm.

— Tilt sensor with an instance of_out-of-range tilt

Although called "simple sensors", the devices are required to support features common to any type of sensor device. The simple sensor has to monitor the environmental characteristic for which it is designed, take samples at defined intervals, compare and process against criteria, and report output in terms of the simple sensor data and additionally in a more detailed diagnostic structure. The following sub-clauses specify all the requirements.

8.2 Implementations

Simple sensors shall be implemented either as a memory mapped simple sensor (see 8.4) or as a ported simple sensor (see 8.5).

8.3 Record structures

Each simple sensor shall support a simple sensor data block. Details of the data blocks for the simple sensor types so far defined are in an annex in ISO/IEC 18000-6. Each of the simple sensors only requires a 32-bit structure, but future simple sensors might require a slightly larger data block size. As additional simple sensors are specified they will be added by amendment to this International Standard.

The ported simple sensor supports additional mandatory and optional records, as detailed in the list. An annex of ISO/IEC 18000-6 defines the requirements for processing these records if present on the ported simple sensor.

The sequence of records is as follows:

— Record 1: Simple sensor data block (mandatory for both implementations), with the following fields:

1 Sensor type
2 Measurement span
3 Accuracy
4 Sampling regime
5 High on-range limit
6 Low in-range limit
7 Monitor delay
8 High out-of-range alarm delay
9 Low out-of-range alarm delay
10 Alarms

- Record 2: Manufacturer record (mandatory only for the ported simple sensor), with the following fields:

  1 Data transmission size (bits)
  2 Password indicator and size
  3 Calibration record indicator
  4 Transfer packet size (in bytes)
  5 Diagnostic memory capacity (in terms of transfer packets)
  6 Configuration history indicator
  7 Time synchronisation type
  8 Number of time synchronisation records

- Record 3: Authorisation password record (optional for the ported simple sensor)

- Record 4: Calibration record (recommended for the ported simple sensor), which holds three values for temperature, 17 values for humidity, and none for impact and tilt

- Record 5: Sample and configuration record (mandatory only for the ported simple sensor), with the following fields:

  1 UTC timestamp at point of manufacture
  2 Number of configuration records
  3 UTC timestamp at configuration and activation
  4 Sampling regime
  5 High in-range limit
  6 Low in-range limit
  7 Monitor delay
  8 High out-of-range limit alarm delay
  9 Low out-of-range limit alarm delay

  The simple sensor may have the facility to store a sequence of Fields 3 to 9. This is declared the configuration history indicator (Field 6 of the Manufacturer record).

- Record 6: Event record (recommended for the ported simple sensor), with the following fields:

  1 Current sample count
  2 Packet count
  3 Sample count at observation
  4 Observed sample data

  Fields 3 and 4 may be repeated a number of times equivalent to the packet count.

- Record 7: Time synchronisation record (mandatory only for the ported simple sensor and only if the event record is present), with the following fields:

  1 RTC time or sample count value
  2 UTC timestamp

  These fields are may be repeated a number of times

The sequence of the set of records numbered 2 to 7 and the size of each memory is specified in logical terms leaving the physical structure a matter of implementation.

The numbers in the list are used as record references for simple sensors in this International Standard. Each record is described in the following sub-clauses.

8.4 Memory mapped simple sensor

8.4.1 Air interface access method

A memory mapped simple sensor is accessed via memory read and write operations to the RFID tag.

8.4.2 Simple sensor data block address

The RFID tag shall implement a method to determine the memory address of the simple sensor data block. Details of the data blocks for the simple sensor types so far defined are in an annex of ISO/IEC 18000-6. Each of these only requires a 32-bit structure, but future simple sensors might require a slightly larger data block size. As additional simple sensors are specified they will be added by amendment to this International Standard.

8.4.2.1 Write-Simple-Sensor-Data-Block Command

This command is intended to be used to configure the simple sensor by writing the entire simple sensor data block into the RFID tag memory. The processes defined in this International Standard ensure that various bit fields are logically structured in accordance with the capabilities of the sensor, as defined by the structure of the simple sensor data block.

The command is used in two distinctly different manners:

— Sensor manufacturers and RFID tag manufacturers may use it to set the first three fields of the record (sensor type, measurement span, and accuracy). These fields should be locked to ensure that the manufacturer settings are not changed. The remaining fields should be zero filled for subsequent configuration by the user or for the alarm states to be set by the sensor. The RFID tag shall not permit setting alarm conditions when processing this command.

— The command is used by users for operational control of the configuration parameters (fields 4 to 9). Access may be restricted by the need for password authorisation. The command shall have the following structure, applied to memory mapped and ported simple sensors to ensure a compliant structure in the air interface commands:

  i) 9 bits zero filled. This covers the three fields encoded by the manufacturer, and the sensor ignores this bit string because the fields are already locked.

  ii) 19 bits for the configuration parameters.

  iii) 4 bits zero filled. This covers the alarm bits, and the sensor ignores this bit string because the fields are reserved to be set automatically by the sensor.

Write operations to the simple sensor data block may be restricted by the use of memory locking, permalocking, password authorisation, or other means implemented by the RFID tag. The RFID tag may be implemented such that some parameter fields (e.g. sensor type, span, accuracy) are not writeable by an interrogator and these bits shall be ignored by the RFID tag when processing this command. The RFID tag shall not permit setting alarm conditions when processing this command.

8.4.2.2 Read-Simple-Sensor-Data-Block Command

This command is intended to be used to obtain the status of a memory mapped simple sensor by reading the simple sensor data block within the RFID tag memory. The simple sensor data block may be obtained by directly reading from the simple sensor data block address. Read operations to the simple sensor data block may be restricted by the use of memory locking, permalocking, password authorisation, or other means implemented by the RFID tag. Other mechanisms specified by an air interface protocol may exist that capture the simple sensor data block such as having a procedure that appends the simple sensor data block to the unique item identifier.

8.4.3 Memory mapped simple sensor operation

Configuring the memory mapped simple sensor shall reset all the alarms within the simple sensor data block, stop the RTC, and initialise the RTC to zero. The memory mapped simple sensor is disarmed when the RTC has a zero value. Writing the current time to the RTC shall then start the RTC running and arm the memory mapped simple sensor. Once armed, the sensor shall be monitored as configured and check for the event conditions required to declare an alarm condition. If an alarm condition other than low battery is declared, then the RTC shall be stopped and its current time retained as a timestamp for when the event occurred.

8.5 Ported simple sensor

8.5.1 Air interface access method

A ported simple sensor is accessed via dedicated sensor commands to the RFID tag.

8.5.2 Record structures

Various records are defined for inclusion on a mandatory or optional basis to support the processing, encoding and subsequent diagnostics of a simple sensor. These are as defined in 8.3.

8.5.3 "Payload" commands

Various simple sensor commands for ported simple sensors are specified in an annex of ISO/IEC 18000-6. These commands (as listed in Table 1 — Payload commands for ported simple sensors) are embedded in the ISO/IEC 18000-6 Type C *HandleSensor* command.

Table 1 — Payload commands for ported simple sensors

Code	Description
00001	Read-Simple-Sensor-Data-Block
00010	Read-Manufacturer-Record
00011	Write-Password
00100	Read-Calibration-Record
00101	Write-Sample-And-Configuration-Record
00110	Initialise-Sensor-Monitoring
00111	Read-Sample-And-Configuration-Record
01000	Read-Event-Record
01001	Write-UTC-Timestamp
01010	Read-Time-Synchronisation-Record
01011	Erase-Monitored-Data
01100	Activate-Simple-Sensor
01101	Deactivate-Simple-Sensor
01110 to 11111	RFU

The construction of the "payload" of these sensor commands and the deconstruction of their responses are specified in 9.3 and Clause 11 of this International Standard.

9 Processing functional application commands and responses

9.1 General

Application interface command may specify a set of functional commands and their associated responses used to configure sensors, initiate processes, and read sampling data from the sensors. The sensor commands can be applied in parallel to the data commands as specified in ISO/IEC 15961-1. This will allow interoperability between item-related data and sensor-related data.

Unlike the data-related commands, all the sensor commands can be more explicitly specified because ISO/IEC 18000-6 (for simple sensors) and IEEE 1451.7 (for full function sensors) prescribe the data content in the sensor memory. Because of this, each data element on a sensor has been assigned an object identifier to align with the command arguments.

The following root-OIDs apply:

1.0.24753.0.r.f	Simple sensors, where the {arc = r} refers to a record number, and {arc = f} refers to a field number
1.0.24753.0.126	Simple sensor command, where the next arc = command number (1..13), and others are reserved
1.0.24753.0.127	Simple sensor response, where the next arc = response number (1..13), and others are reserved
1.0.24753.7.r.f	IEEE 1451.7 full function sensors, the {arc = r} refers to a record number, and {arc = f} refers to a field number
1.0.24753.7.126	IEEE 1451.7 command, where the next arc = command number (1..21), and others are reserved
1.0.24753.7.127	IEEE 1451.7 response, where the next arc = response number (1..21), and others are reserved

NOTE: In ISO/IEC 15961-1, the object identifiers are, by necessity, defined by the application. Another difference is that the sensor commands do not have qualifying arguments that determine the sequence, compaction scheme, or whether the data is locked. Everything is prescribed by the relevant sensor record structure.

Using the object identifier structure also enables greater integration with this standard and ISO/IEC 24791-5 (for device interfaces) and 24791-2 (for data management).

This International Standard uses two slightly different approaches to the use of object identifiers. For simple sensors the data elements identified as arguments in the commands and responses are assigned object identifiers based on the root-OID of the command or response. This is because of the optional nature of some of the associated records and their fields. For full function sensors, the records and their inherent fields are mandatory. This enables each field to be assigned a unique object identifier. Object identifiers based on a root-OID for IEEE 1451.7 commands and responses are only applied where a field name is not applied (for example for a response code).

9.2 Processing full function sensors functional application commands and responses

9.2.1 General

The command and response names used in the following sub-clauses align with the same names as used in IEEE 1451.7. The 5-bit code that identifies the command and response is converted to and from the object identifier.

Many of the commands and responses contain fields for security functions that are not supported by ISO/IEC 18000-6, which is the only 18000 series air interface protocol standard that supports sensors. As theses fields are required for the payload sensor command detailed advice is provided in the sub-clauses below of how to address the issue.

In addition many of the IEEE 1451.7 commands have two additional arguments *Reader-Security-Token* and *New-Reader-RN* that are present at the end of the command. It is not possible to support these arguments by processes in this International Standard because they are outside the scope of being defined by the application. Most of the IEEE 1451.7 responses have the arguments *Sensor-Security-Token* and *New-Sensor-RN*. These arguments are created in the IEEE 1451.7 commands by the sensor driver, with the default value 0_2. Where these arguments are also in the IEEE 1451.7 responses, the sensor driver shall remove them prior to transferring to the processes defined in this International Standard.

9.2.2 Common command-based object Identifiers

The following command-based object identifiers apply to all commands except: Read-Sensor-Identifier (see 9.2.4).

Sensor-Address-Type [OID ref: 1 0 24753 7 126 {command code} 1]

Sensor interface:

00 = no sub-addressing
01 = use 7-bit sub-addressing
10 = use sensor type sub-addressing
11 = use sensor ID sub-addressing

App-interface: INTEGER (0..3)

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 {command code} 2]

Sensor interface and App-interface: BIT STRING (0, or 7, or 15, or 64)

If Sensor-Address-Type =	00	then this field is empty
	01	then this field is the 7-bit sensor sub-address
	10	then this field is the 15-bit concatenation of: TEDS-Type, Sensor-Type, and Units-Extension
	11	then this field is the 64-bit Sensor-Id

9.2.3 Common response-based object identifiers

The following response-based object identifiers apply to all responses.

Response-Code [OID ref: 1 0 24753 7 127 {response code} 1]

Sensor interface: 3-bit code, as defined in the IEEE 1451.7 Read-Sensor-Identifier response in the current versions of IEEE 1451.7. Only code 111 provides a full response

App-interface: INTEGER (0..7), where the following codes result in a truncated response caused by some sensor error:

0, 1, 3, 4, and 5 - truncated at this point
2 - truncated after the next field

Battery-Status-Code [OID ref: 1 0 24753 7 127 {response code} 2]:

Sensor interface and App-interface:

0 = Battery OK
1 = Battery low

9.2.4 Read-Sensor-Identifier

This command is used to return the different identifiers used for subsequent communication with the sensor if this is not achieved by using a port number on the RFID tag itself.

Sensor interface for command and response: 5-bit code = 00001

App-interface: Command = **[OID ref: 1 0 24753 7 126 1]**
Response = **[OID ref: 1 0 24753 7 127 1]**

9.2.4.1 Command-based object identifiers

The following command argument and object identifier apply:

Read-Sensor-Identifier-Parameter [OID ref: 1 0 24753 7 126 1 1]
Sensor interface and App-interface:
0 = respond with the sub-address (port number) and identifier
1 = response with the sub-address and Primary TEDS Fields 1,2,3

9.2.4.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Response-Code [OID ref: 1 0 24753 7 127 1 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 1 2]:
See 9.2.3 for details

Sub-Address [OID ref: 1 0 24753 7 127 1 3]:
Sensor interface: 7-bit code
App-interface: INTEGER (0..127)
This is the sub-address number as assigned by the sensor

9.2.4.3 Memory-based object identifiers in the response

The response contains, depending on the request parameter in the command, the following data elements:

— **Sensor-Id [OID ref: 1 0 24753 7 1]** (see 10.2 for processing)

— **TEDS-Type [OID ref: 1 0 24753 7 2 1]** (see 10.3.1 for processing)

— **Sensor-Type [OID ref: 1 0 24753 7 2 2]** (see 10.3.2 for processing)

— **Units-Extension [OID ref: 1 0 24753 7 2 3]** (see 10.3.3 for processing)

9.2.4.4 Process requirements

Other than mapping between sensor interface format and the app-interface format, no special processing is required.

9.2.5 Read-Primary-Characteristics-TEDS

This command is used to return the read-only Primary Characteristics TEDS record for a nominated full function sensor.

Sensor interface for command and response: 5-bit code = 00010

App-interface: Command = **[OID ref: 1 0 24753 7 126 2]**
Response = **[OID ref: 1 0 24753 7 127 2]**

9.2.5.1 Command-based object identifiers

The following command arguments and object identifier apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 2 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 2 2]
See 9.2.2 for details

Read-Primary-TEDS-Parameter [OID ref: 1 0 24753 7 126 2 3]
Sensor interface and App-interface:
0 = respond only with Primary TEDS
1 = Primary TEDS + Unique Identifier

9.2.5.2 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 7 127 2 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 2 2]:
See 9.2.3 for details

9.2.5.3 Memory-based object identifiers in the response

The response contains, depending on the request parameter in the command, the following data elements:

— **Sensor-Id [OID ref: 1 0 24753 7 1]** (see 10.2 for processing)

— **TEDS-Type [OID ref: 1 0 24753 7 2 1]** (see 10.3.1 for processing)

— **Sensor-Type [OID ref: 1 0 24753 7 2 2]** (see 10.3.2 for processing)

— **Units-Extension [OID ref: 1 0 24753 7 2 3]** (see 10.3.3 for processing)

— **Sensor-Map [OID ref: 1 0 24753 7 2 4]** (see 10.3.4 for processing)

— **Data-Resolution [OID ref: 1 0 24753 7 2 5]** (see 10.3.5 for processing)

— **Scale-Factor-Significand [OID ref: 1 0 24753 7 2 6]** (see 10.3.6 for processing)

— **Scale-Factor-Exponent [OID ref: 1 0 24753 7 2 7]** (see 10.3.7 for processing)

— **Scale-Offset-Significand [OID ref: 1 0 24753 7 2 8]** (see 10.3.8 for processing)

— **Scale-Offset-Exponent [OID ref: 1 0 24753 7 2 9]** (see 10.3.9 for processing)

— **Data-Uncertainty [OID ref: 1 0 24753 7 2 10]** (see 10.3.10 for processing)

— **Sensor-Reconfiguration-Capability [OID ref: 1 0 24753 7 2 11]** (see 10.3.11 for processing)

— **Memory-Rollover-Capability [OID ref: 1 0 24753 7 2 12]** (see 10.3.12 for processing)

9.2.5.4 Process requirements

The process requirements are to convert the fields listed in 9.2.5.3 of the 128-bit **Primary-Characteristics-TEDS** record into its constituent parts as defined by the Primary Sensor Characteristics TEDS (Type 1) record in IEEE 1451.7. The fields that are not listed are ignored.

9.2.6 Write-Sample-And-Configuration

This command is used to construct the Sample and Configuration as a bit string from the various component real values using the relevant conversion rules.

Sensor interface for command and response: 5-bit code = 00011

App-interface: Command = **[OID ref: 1 0 24753 7 126 3]**
Response = **[OID ref: 1 0 24753 7 127 3]**

9.2.6.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 3 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 3 2]
See 9.2.2 for details

9.2.6.2 Memory-based object identifiers in the command

The command contains the following data elements:

— **UTC-Timestamp-At-Configuration [OID ref: 1 0 24753 7 3 1]** (see 10.4.1 for processing)

— **Sample-Interval [OID ref: 1 0 24753 7 3 2 0]** for seconds, or **[OID ref: 1 0 24753 7 3 2 1]** for minutes. (see 10.4.2 for processing)

— **Monitor-Delay [OID ref: 1 0 24753 7 3 3 0]** for seconds, or **[OID ref: 1 0 24753 7 3 3 1]** for minutes (see 10.4.3 for processing)

— **Alarm-Values-Set [OID ref: 1 0 24753 7 3 4]** (see 10.4.4 for processing)

— **Memory-Rollover-Capability [OID ref: 1 0 24753 7 3 5]** (see 10.4.5 for processing)

— **Upper-Alarm-Threshold [OID ref: 1 0 24753 7 3 12]** (see 10.4.6 for processing)

— **Lower-Alarm-Threshold [OID ref: 1 0 24753 7 3 13]** (see 10.4.7 for processing)

Fields 6 to 11 are all associated with security aspects that, if implemented, are addressed by the interrogator or some other mechanism outside the scope of this International Standard. To ensure that the command is a valid output from the sensor data processor, these fields shall be zero filled with a string that is 16 bits long. This bit string is to precede the upper and lower alarm thresholds (fields 12 and 13), if present.

9.2.6.3 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 7 127 3 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 1 2]:
See 9.2.3 for details

9.2.6.4 Process requirements

The processes require the application interface values for each field to be converted to a bit string and for the resulting bit string to be concatenated to that already converted. The two threshold values have a length, in bits, determined by the data resolution.

9.2.7 Read-Sample-And-Configuration

This command is used to read the Sample and Configuration as a bit string and re-constructing the various component real values using the relevant conversion rules.

Sensor interface for command and response: 5-bit code = 00100

App-interface: Command = **[OID ref: 1 0 24753 7 126 4]**
Response = **[OID ref: 1 0 24753 7 127 4]**

9.2.7.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 4 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 4 2]
See 9.2.2 for details

9.2.7.2 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 7 127 4 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 4 2]:
See 9.2.3 for details

9.2.7.3 Memory-based object identifiers in the response

The response contains the fields defined in 9.2.6.2 used to configure the sensor.

The 16-bit string for fields 6 to 11 cannot be interpreted by rules defined in the current version of ISO/IEC 18000-6, so are likely to be passed to the implementation of this International Standard. As these six fields cannot be processed, they have to be discarded. However, as they precede the optional upper and lower alarm threshold values (which will be non-zero if present) care needs to be taken in interpreting the response to this command.

9.2.7.4 Process requirements

The processes require the bit string as read from the air interface to be parsed to each field and converted to the application interface values, each of which is the data associated with an object identifier in the response. The two threshold values have a length, in bits, determined by the data transmission format.

9.2.8 Read-Alarm-Status

This command is used to read information about alarms from different memory records on the sensor to enable more specific diagnostics to be invoked using other commands.

Sensor interface for command and response: 5-bit code = 00101

App-interface: Command = **[OID ref: 1 0 24753 7 126 5]**
Response = **[OID ref: 1 0 24753 7 127 5]**

9.2.8.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 5 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 5 2]
See 9.2.2 for details

9.2.8.2 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 7 127 5 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 5 2]:
See 9.2.3 for details

9.2.8.3 Memory-based object identifiers in the response

The response contains the following data elements:

— **Alarm-Values-Set [OID ref: 1 0 24753 7 3 4]** (see 10.4.4 for processing)

— **Alarm-Triggered [OID ref: 1 0 24753 7 4 6]** (see 10.5.7 for processing)

— **Sensor-Map [OID ref: 1 0 24753 7 2 4]** (see 10.3.4 for processing)

9.2.8.4 Process requirements

The processes require the bit string as read from the air interface to be parsed to each field and converted to the application interface values, each of which is the data associated with an object identifier in the response.

9.2.9 Read-Single-Memory-Record

This command returns data associated with single entry records from the Event record. The responses have different structures depending on the type of single entry record.

Sensor interface for command and response: 5-bit code = 00110

App-interface: Command = **[OID ref: 1 0 24753 7 126 6]**
Response = **[OID ref: 1 0 24753 7 127 6]**

9.2.9.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 6 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 6 2]
See 9.2.2 for details

Measurement-Type [OID ref: 1 0 24753 126 6 3]
Sensor interface: BIT STRING (4)
The valid codes are 0000 to 1001; codes 1010 to 1101 do not apply to this command; codes 1110 and 1111 are reserved and not defined in IEEE 1451.7

App-interface: INTEGER (0..9)

9.2.9.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Response-Code [OID ref: 1 0 24753 7 127 6 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 6 2]:
See 9.2.3 for details

9.2.9.3 Memory-based object identifiers in the response

The response contents differ, dependent on the measurement type code.

For measurement type code 0, the response contains the following data element:

Present-Value [OID ref: 1 0 24753 7 5 0] (see 10.6.4 for processing)

For measurement type code 1, the response contains the following data element:

Maximum-Value [OID ref: 1 0 24753 7 5 1] (see 10.6.5 for processing)

For measurement type code 2, the response contains the following data element:

Minimum-Value [OID ref: 1 0 24753 7 5 2] (see 10.6.6 for processing)

For measurement type code 3, the response contains the following data element:

Average-Value [OID ref: 1 0 24753 7 5 3] (see 10.6.7 for processing)

For measurement type code 4, the response contains the following data element:

Variance [OID ref: 1 0 24753 7 5 4] (see 10.6.8 for processing)

For measurement type code 5, the response contains the following data element:

Standard-Deviation [OID ref: 1 0 24753 7 5 5] (see 10.6.9 for processing)

For measurement type code 6, the response contains the following data elements:

— **Sample-Count-Predetermined-Time [OID ref: 1 0 24753 7 4 7]** (see 10.5.8 for processing)

— **Observed-Value-Predetermined-Time [OID ref: 1 0 24753 7 5 6]** (see 10.6.10 for processing)

For measurement type code 7, the response contains the following data elements:

— **Sample-Count-Critical-Event [OID ref: 1 0 24753 7 4 8]** (see 10.5.9 for processing)

— **Observed-Value-Critical-Event [OID ref: 1 0 24753 7 5 7]** (see 10.6.11 for processing)

For measurement type code 8, the response contains the following data element:

Count-Over-Upper-Threshold [OID ref: 1 0 24753 7 5 8] (see 10.6.12 for processing)

For measurement type code 9, the response contains the following data element:

Count-Under-Lower-Threshold [OID ref: 1 0 24753 7 5 9] (see 10.6.13 for processing)

9.2.9.4 Process requirements

If an invalid or reserved measurement code is returned, the processing of other data from the sensor shall be abort and the application advised of the error. The process requires the binary data from the sensor to be converted to the rules defined for each object identifier.

9.2.10 Read-Event-Administration-Record

This command returns the content of the Event Administration record, but only if the field is present; and for any of fields 7 to 10 present in memory, only if the field encodes a non-zero value.

Sensor interface for command and response: 5-bit code = 00111

App-interface: Command = **[OID ref: 1 0 24753 7 126 7]**
Response = **[OID ref: 1 0 24753 7 127 7]**

9.2.10.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 7 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 7 2]
See 9.2.2 for details

9.2.10.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Response-Code [OID ref: 1 0 24753 7 127 7 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 7 2]:
See 9.2.3 for details

9.2.10.3 Memory-based object identifiers in the response

The response contains the following data elements, depending on the presence of the field within the memory:

— **Code10-Sample-Capacity [OID ref: 1 0 24753 7 4 1]** [CONDITIONAL] (see 10.5.2 for determining the presence of this OID and for its processing)

- **Code11-Sample-Capacity [OID ref: 1 0 24753 7 4 2]** [CONDITIONAL] (see 10.5.3 for determining the presence of this OID and for its processing)
- **Code12-Sample-Capacity [OID ref: 1 0 24753 7 4 3]** [CONDITIONAL] (see 10.5.4 for determining the presence of this OID and for its processing)
- **Code13-Sample-Capacity [OID ref: 1 0 24753 7 4 4]** [CONDITIONAL] (see 10.5.5 for determining the presence of this OID and for its processing)
- **Sample-Count [OID ref: 1 0 24753 7 4 5]** (see 10.5.6 for processing)
- **Alarm-Triggered [OID ref: 1 0 24753 7 4 6]** (see 10.5.7 for processing)
- **Sample-Count-Predetermined-Time [OID ref: 1 0 24753 7 4 7]** [CONDITIONAL] (see 10.5.8 for determining the presence of this OID and for its processing)
- **Sample-Count-Critical-Event [OID ref: 1 0 24753 7 4 8]** [CONDITIONAL] (see 10.5.9 for determining the presence of this OID and for its processing)
- **Sample-Count-Events-Outside-Either-Threshold [OID ref: 1 0 24753 7 4 9]** [CONDITIONAL] (see 10.5.10 for determining the presence of this OID and for its processing)
- **Sample-Count-First-Threshold-Event [OID ref: 1 0 24753 7 4 10]** [CONDITIONAL] (see 10.5.11 for determining the presence of this OID and for its processing)

9.2.10.4 Process requirements

The process requires the binary data from the sensor to be converted to the rules defined for each object identifier.

9.2.11 Read-Event-Record-Segments

This command reads multiple sensor words from the event record in terms of a number of segments, each having the memory capacity for containing 32 sensor words. The sensor word comprises of the data, and for two record types, a sample count value.

Depending on the size of the data transmission format and the inclusion of the sample count, a response to this command can consist of a significant number of bits, even for a single segment. Although beyond the scope of this International Standard, each implementation needs to take into account the probability of a successful read transaction across the air interface and adjust the number of segments called by the command. If even one segment proves to create transfer problems, then the Read-Partial-Event-Record-Segment command (9.2.12) should be used.

IEEE 1451.7 supports the capability for asynchronous responses for this command, which a sensor might support. If the total number of segments requested in the command is less than the total number of encoded segments, the sensor can create additional responses of the same number of segments. As asynchronous responses are not supported by the ISO/IEC 18000-6 Type C air interface protocol, this processing is not supported by this International Standard.

Sensor interface for command and response: 5-bit code = 01000

App-interface: Command = **[OID ref: 1 0 24753 7 126 8]**
Response = **[OID ref: 1 0 24753 7 127 8]**

9.2.11.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 8 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 8 2]
See 9.2.2 for details

Measurement-Type [OID ref: 1 0 24753 126 8 3]
Sensor interface: BIT STRING (4)
The valid codes are 1010 to 1101; the reserved codes are 1110 and 1111; the others do not apply to this command

App-interface: INTEGER (10..13)

First-Segment-Number [OID ref: 1 0 24753 126 8 4]
Sensor interface: BIT STRING (3 or 11)
The segment number for measurement type 1011 is 3-bits long, and for the other three measurement types is 11-bits long

App-interface: INTEGER

Number-Of-Segments [OID ref: 1 0 24753 126 8 5]
Sensor interface: BIT STRING (6)

App-interface: INTEGER

Last-Segment-Number [OID ref: 1 0 24753 126 8 6]
Sensor interface: BIT STRING (3 or 11)
The segment number for measurement type 1011 is 3-bits long, and for the other three measurement types is 11-bits long

App-interface: INTEGER

9.2.11.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Response-Code [OID ref: 1 0 24753 7 127 8 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 8 2]:
See 9.2.3 for details

CRC16-Validation-Error [OID ref: 1 0 24753 7 127 8 3]
Sensor interface and App-interface:
0 = no error
1 = error, then this is followed by the next argument, presented as a list of segments that failed the CRC-16 validation

CRC16-Failed-Segment [OID ref: 1 0 24753 7 127 8 4]
Sensor interface: BIT STRING (3 or 11)
The segment number for measurement type 1011 is 3-bits long, and for the other three measurement types is 11-bits long

App-interface: INTEGER (0 to 255 for type 11; 0 to 2047 for the other three measurement types)

The **Error-Code** is as delivered from the sensor response across the air interface. The **CRC16-Validation-Error** is as the result of a process (see 10.6.3) specified in this International Standard and applied in a conformant implementation.

Because the object identifiers, as defined in the next sub-clause, identify the segment and sensor word the data from CRC-16 validated segments can be processed and passed to the application.

9.2.11.3 Memory-based object identifiers in the response

The response contains one of the following data elements, depending on the presence of the event record within the memory, and the request in the command.

For measurement type code 10, the response contains the following data element:

Observed-Value-Data-Log-All-Samples [OID ref: 1 0 24753 7 5 10 {segment} {sensor-word}] (see 10.6.14 for processing)

For measurement type code 11, the response contains the following data elements:

— **8bitSampleCount-Data-Log-Outside-Threshold [OID ref: 1 0 24753 7 5 11 1 {segment} {sensor-word}]** (see 10.6.15 for processing)

— **Observed-Value-Data-Log-Outside-Threshold [OID ref: 1 0 24753 7 5 11 2 {segment} {sensor-word}]** (see 10.6.15 for processing)

For measurement type code 12, the response contains the following data elements:

— **16bitSampleCount-Data-Log-Outside-Threshold [OID ref: 1 0 24753 7 5 12 1 {segment} {sensor-word}]** (see 10.6.16 for processing)

— **Observed-Value-Data-Log-Outside-Threshold [OID ref: 1 0 24753 7 5 12 2 {segment} {sensor-word}]** (see 10.6.16 for processing)

For measurement type code 13, the response contains the following data elements:

Observed-Value-Data-Log-After-Initial-Alarm [OID ref: 1 0 24753 7 5 13 {segment} {sensor-word}] (see 10.6.17 for processing)

9.2.11.4 Process requirements

In constructing the command, two conditions apply:

— The value of the last segment shall not be greater than the number of segments supported by the sensor

— The number of segments shall be no greater than (Last Segment Number) – (First Segment Number) +1. The number may be smaller.

The process requires the binary data from the sensor to be converted to the rules defined for each object identifier.

9.2.12 Read-Partial-Event-Record-Segment

This command reads part of an event record segment in terms of one or more sensor words.

Even though the response to this command transfers less data than the Read-Event-Record-Segments command (9.2.11), it can consist of a significant number of bits. Although beyond the scope of this

International Standard, each implementation needs to take into account the probability of a successful read transaction across the air interface and adjust the number of sensor words called by the command.

Because the size of the sensor word might not align with the physical boundary of any physical memory, some processing capability on the sensor itself is expected to organise the transfer of data so that only complete sensor words are transferred. This is a fundamental requirement for subsequent processing compliant with this International standard.

Sensor interface for command and response: 5-bit code = 01001

App-interface: Command = **[OID ref: 1 0 24753 7 126 9]**
Response = **[OID ref: 1 0 24753 7 127 9]**

9.2.12.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 9 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 9 2]
See 9.2.2 for details

Measurement-Type [OID ref: 1 0 24753 126 9 3]
Sensor interface: BIT STRING (4)
The valid codes are 1010 to 1101; the reserved codes are 1110 and 1111; the others do not apply to this command

App-interface: INTEGER

Segment-Number [OID ref: 1 0 24753 126 9 4]
Sensor interface: BIT STRING (3 or 11)
The segment number for measurement type 1011 is 3-bits long, and for the other three measurement types is 11-bits long

App-interface: INTEGER

First-Sample-Number [OID ref: 1 0 24753 126 9 5]
Sensor interface: BIT STRING (5) (00000..11111)
App-interface: INTEGER (1..32)

Number-Of-Samples [OID ref: 1 0 24753 126 9 6]
Sensor interface: BIT STRING (5) (00000..11111)
App-interface: INTEGER (1..32)

9.2.12.2 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 7 127 9 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 9 2]:
See 9.2.3 for details

9.2.12.3 Memory-based object identifiers in the response

The response for measurement type codes 10, 11, 12 and 13 are as defined in 9.2.11.3 for reading single or multiple segments.

9.2.12.4 Process requirements

The sum of (First Sample Number) + (Number of Samples) shall not be greater than the number of sample values (32) in a segment.

The process requires the binary data from the sensor to be converted to the rules defined for each object identifier.

9.2.13 Write-Event-Admin-Field7

This command is used to write the 16-bit value that equates to the sample count at a pre-determined event. This pre-determined time is calculated by the application independently of the International Standard, but the method defined in 10.5.8 can be used.

Sensor interface for command and response: 5-bit code = 01010

App-interface: Command = **[OID ref: 1 0 24753 7 126 10]**
Response = **[OID ref: 1 0 24753 7 127 10]**

9.2.13.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 10 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 10 2]
See 9.2.2 for details

9.2.13.2 Memory-based object identifier in the command

The command contains the following data element:

Sample-Count-Predetermined-Time [OID ref: 1 0 24753 7 4 7] (see 10.5.8 for processing)

9.2.13.3 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 7 127 10 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 10 2]:
See 9.2.3 for details

9.2.13.4 Process requirements

There are no process requirements because the value of the Sample-Count-Predetermined-Time is calculated by the application.

9.2.14 Read-Any-Field

This command is used to read any single entry field from any record of the sensor. Explicitly excluded are the data log records in the event record.

Sensor interface for command and response: 5-bit code = 01011

App-interface: Command = **[OID ref: 1 0 24753 7 126 11]**
Response = **[OID ref: 1 0 24753 7 127 11]**

9.2.14.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 11 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 11 2]
See 9.2.2 for details

Record-Id [OID ref: 1 0 24753 7 126 11 3]
Sensor interface:
00 = Primary Sensor Characteristics TEDS record
01 = Sample and Configuration record
10 = Event record
11 = Event Administration record

App-interface: INTEGER (0..3)

Field-Number [OID ref: 1 0 24753 7 126 11 4]
Sensor interface: BIT STRING (5), but constrained by the highest field number in the record.

App-interface: INTEGER (0..22), but constrained by the highest field number in the record as follows:

- Primary Sensor Characteristics TEDS record (1..22), this excludes the RFU type code 13 (until this is defined)
- Sample and Configuration record (1..13)
- Event record type (0..9), this excludes the multiple event record types 10 to 13, and the RFU type codes 13 and 14 (until these are defined)
- Event Administration record (1..12)

9.2.14.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Response-Code [OID ref: 1 0 24753 7 127 11 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 11 2]:
See 9.2.3 for details

9.2.14.3 Memory-based object identifiers in the response

Only the fields that can be processed by rules in this International Standard are defined below.

If the command requested a field from the Primary Sensor Characteristics TEDS record, the response contains, depending on the request field parameter in the command, one of the following data elements:

— **TEDS-Type [OID ref: 1 0 24753 7 2 1]** (see 10.3.1 for processing)

— **Sensor-Type [OID ref: 1 0 24753 7 2 2]** (see 10.3.2 for processing)

— **Units-Extension [OID ref: 1 0 24753 7 2 3]** (see 10.3.3 for processing)

— **Sensor-Map [OID ref: 1 0 24753 7 2 4]** (see 10.3.4 for processing)

— **Data-Resolution [OID ref: 1 0 24753 7 2 5]** (see 10.3.5 for processing)

— **Scale-Factor-Significand [OID ref: 1 0 24753 7 2 6]** (see 10.3.6 for processing)

— **Scale-Factor-Exponent [OID ref: 1 0 24753 7 2 7]** (see 10.3.7 for processing)

— **Scale-Offset-Significand [OID ref: 1 0 24753 7 2 8]** (see 10.3.8 for processing)

— **Scale-Offset-Exponent [OID ref: 1 0 24753 7 2 9]** (see 10.3.9 for processing)

— **Data-Uncertainty [OID ref: 1 0 24753 7 2 10]** (see 10.3.10 for processing)

— **Sensor-Reconfiguration-Capability [OID ref: 1 0 24753 7 2 11]** (see 10.3.11 for processing)

— **Memory-Rollover-Capability [OID ref: 1 0 24753 7 2 12]** (see 10.3.12 for processing)

If the command requested a field from the Sample and Configuration record, the response contains, depending on the request field parameter in the command, one of the seven data elements defined in 9.2.6.2.

If the command requested a single entry field from the Event record, the response contains, depending on the request field parameter in the command, one of the ten data elements defined in 9.2.9.3.

If the command requested a field from the Event Administration record, the response contains, depending on the request field parameter in the command, one of the ten data elements defined in 9.2.10.3.

9.2.14.4 Process requirements

The process requires the binary data from the sensor to be converted to the rules defined for each object identifier.

9.2.15 Erase-Event-Administration-Record

This command sets to 0_2 fields 5, 6, 7, 8, 9, and 10 of the Event Administration Record, given that fields 1, 2, 3, and 4 are locked by the sensor manufacturer.

Sensor interface for command and response: 5-bit code = 01100

App-interface: Command = **[OID ref: 1 0 24753 7 126 12]**
Response = **[OID ref: 1 0 24753 7 127 12]**

9.2.15.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 12 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 12 2]
See 9.2.2 for details

9.2.15.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Response-Code [OID ref: 1 0 24753 7 127 12 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 12 2]:
See 9.2.3 for details

9.2.15.3 Memory-based object identifiers

There are no memory-based object identifiers, because there are no processes carried out by this International Standard.

9.2.15.4 Process requirements

There are no process requirements because the command instructs to sensor to erase memory and responds with a simple response code.

9.2.16 Erase-Event-Records

This command erases the complete set of event records. The process to achieve this is beyond the scope of this International Standard and the associated air interface protocol standard.

Sensor interface for command and response: 5-bit code = 01101

App-interface: Command = **[OID ref: 1 0 24753 7 126 13]**
Response = **[OID ref: 1 0 24753 7 127 13]**

9.2.16.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 13 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 13 2]
See 9.2.2 for details

9.2.16.2 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 7 127 13 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 13 2]:
See 9.2.3 for details

9.2.16.3 Memory-based object identifiers

There are no memory-based object identifiers, because there are no processes carried out by this International Standard.

9.2.16.4 Process requirements

There are no process requirements because the command instructs to sensor to erase memory and responds with a simple response code.

9.2.17 Erase-Sample-And-Configuration-Record

This command is used to erase the sample and configuration record.

Sensor interface for command and response: 5-bit code = 01110

App-interface: Command = **[OID ref: 1 0 24753 7 126 14]**
Response = **[OID ref: 1 0 24753 7 127 14]**

9.2.17.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 14 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 14 2]
See 9.2.2 for details

9.2.17.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Response-Code [OID ref: 1 0 24753 7 127 14 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 14 2]:
See 9.2.3 for details

9.2.17.3 Memory-based object identifiers

There are no memory-based object identifiers, because there are no processes carried out by this International Standard.

9.2.17.4 Process requirements

There are no process requirements because the command instructs to sensor to erase memory and responds with a simple response code.

9.2.18 Begin-End-Mission

This command allows the sensor to be configured at one time and begin its mission at another time. This saves both memory and battery life. The command is also used to end a mission and for the sensor to cease monitoring its environment.

Sensor interface for command and response: 5-bit code = 01111

App-interface: Command = **[OID ref: 1 0 24753 7 126 15]**
Response = **[OID ref: 1 0 24753 7 127 15]**

9.2.18.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Sensor-Address-Type [OID ref: 1 0 24753 7 126 15 1]
See 9.2.2 for details

Sensor-Comms-Id [OID ref: 1 0 24753 7 126 15 2]
See 9.2.2 for details

Begin-End [OID ref: 1 0 24753 7 126 15 3]
Sensor and **App-interface**
0 = Begin
1 = End

NOTE IEEE 1451.7 may apply additional security features that are currently not supported by ISO/IEC 18000-6 and this International standard.

9.2.18.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Response-Code [OID ref: 1 0 24753 7 127 15 1]
See 9.2.3 for details

Battery-Status-Code [OID ref: 1 0 24753 7 127 15 2]:
See 9.2.3 for details

9.2.18.3 Memory-based object identifiers

There are no memory-based object identifiers, because there are no processes carried out by this International Standard.

9.2.18.4 Process requirements

There are no process requirements because the command instructs to sensor to erase memory and responds with a simple response code.

9.2.19 Other IEEE 1451.7 commands

IEEE 1451.7 lists the following commands, preceded by their command codes:

— 10000	Challenge
— 10001	Reader-Authenticate
— 10010	ReadWriteLock-Keys
— 10011	Request-RN
— 10100	Encryption-On-Off
— 10101	Close-Secure-Session

These commands are currently not supported by ISO/IEC 18000-6 and therefore also not supported by this International Standard. As and when support is provided by a referenced air interface, the commands will be supported by future revision of this International Standard.

9.3 Processing simple sensors functional application commands and responses

9.3.1 Read-Simple-Sensor-Data-Block

This command is used to read the Simple Sensor Data Block in air interface protocols where this is not returned automatically as an appended component to reading a unique item identifier.

Sensor interface for command and response: 5-bit code = 00001

App-interface: Command = **[OID ref: 1 0 24753 0 126 1]**
Response = **[OID ref: 1 0 24753 0 127 1]**

9.3.1.1 Command-based object identifiers

This command has no object identifiers other than the Module-OID for the command itself.

9.3.1.2 Response-based object identifiers

The following response argument and object identifier apply:

Error-Code [OID ref: 1 0 24753 0 127 1 1]
Sensor interface and **App-interface**:
0 = no error
1 = error, then this is the only element in the response

9.3.1.3 Memory-based object identifiers in the response

The response contains the following data elements, some of which are null fields for some types of simple sensor:

— **Sensor-Type [OID 1 0 24753 0 1 1]** (see 11.2.2 for processing)

— **Measurement-Span [OID 1 0 24753 0 1 2]** (see 11.2.3 for processing)

— **Accuracy [OID 1 0 24753 0 1 3]** (see 11.2.4 for processing)

— **Sampling-Regime [OID 1 0 24753 0 1 4]** (see 11.2.5 for processing)

— **High-In-Range-Limit [OID 1 0 24753 0 1 5]** (see 11.2.6 for processing)

— **Low-In-Range-Limit [OID 1 0 24753 0 1 6]** (see 11.2.7 for processing)

— **Monitor-delay [OID 1 0 24753 0 1 7]** (see 11.2.8 for processing)

— **High-Out-Of-Range-Alarm-Delay [OID 1 0 24753 0 1 8]** (see 11.2.9 for processing)

— **Low-Out-Of-Range-Alarm-Delay [OID 1 0 24753 0 1 9]** (see 11.2.10 for processing)

— **Alarms [OID 1 0 24753 0 1 10]** (see 11.2.11 for processing)

9.3.2 Read-Manufacturer-Record

This command provides accesses information from the sensor manufacturer, including indicating the presence and structure of other records associated with the simple sensor.

Sensor interface for command and response: 5-bit code = 00010

App-interface: Command = **[OID ref: 1 0 24753 0 126 2]**
Response = **[OID ref: 1 0 24753 0 127 2]**

9.3.2.1 Command-based object identifier

The following command argument and object identifier apply:

Response-Type [OID ref: 1 0 24753 0 126 2 1]
Sensor interface
00 = Respond with the first 32 bits
01 = Respond with the complete block in one transmission
10 = RFU
11 = RFU

App-interface: INTEGER (0..3)

9.3.2.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Error-Code [OID ref: 1 0 24753 0 127 2 1]
Sensor interface and **App-interface:**
0 = no error
1 = error, then this is the only element in the response

Packet-Number [OID ref: 1 0 24753 0 127 2 2]
This is currently RFU in the command structure; therefore the response data is zero

9.3.2.3 Memory-based object identifiers in the response

The response contains the following data elements:

— **Data-Transmission-Size [OID 1 0 24753 0 2 1]** (see 11.3.2 for processing)

— **Password-Indicator [OID 1 0 24753 0 2 2]** (see 11.3.3 for processing)

— **Calibration-Record-Indicator [OID 1 0 24753 0 2 3]** (see 11.3.4 for processing)

— **Transfer-Packet-Size [OID 1 0 24753 0 2 4]** (see 11.3.5 for processing)

— **Diagnostic-Memory-Capacity [OID 1 0 24753 0 2 5]** (see 11.3.6 for processing)

— **Configuration-History-indicator [OID 1 0 24753 0 2 6]** (see 11.3.7 for processing)

— **Time-Synchronisation-Type [OID 1 0 24753 0 2 7]** (see 11.3.8 for processing)

— **Number-Time-Synch-Records [OID 1 0 24753 0 2 8]** (see 11.3.9 for processing)

— **Additional-Manufacturer-Data [OID 1 0 24753 0 2 9]** (see 11.3.10 for processing)

9.3.3 Write-Password

This command is used to transfer a password selected by the owner of the sensor to be encoded on the sensor, to enable only authorised future access to parts of the simple sensor memory.

Sensor interface for command and response: 5-bit code = 00011

App-interface: Command = **[OID ref: 1 0 24753 0 126 3]**
Response = **[OID ref: 1 0 24753 0 127 3]**

9.3.3.1 Command-based object identifier

The following command argument and object identifier apply:

Password-Size [OID ref: 1 0 24753 0 126 3 1]
Sensor interface and **app-interface**: BIT STRING (2)
00 = No password
01 = 32 bit
10 = 64 bit
11 = 128 bit

Password [OID ref: 1 0 24753 0 126 3 2]
Sensor interface and **App-interface**: BYTE STRING (4, or 8, or 16)
Bits 5 and 6 of the manufacturer record declare the size of the password.

9.3.3.2 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 0 127 3 1]
Sensor Interface
00 = success
01 = sensor not identified
10 = password size not supported
11 = other failure

App-interface: INTEGER (0..3)

9.3.4 Read-Calibration-Record

This command provides access to the manufacturer's calibration record, which currently applies to temperature and humidity sensors, but not to tilt and impact sensors.

Sensor interface for command and response: 5-bit code = 00100

App-interface: Command = **[OID ref: 1 0 24753 0 126 4]**
Response = **[OID ref: 1 0 24753 0 127 4]**

9.3.4.1 Command-based object identifier

The following command argument and object identifier apply:

Response-Type [OID ref: 1 0 24753 0 126 4 1]
Sensor and **App-interface**
0 = Respond with the complete block in one transmission
1 = RFU

9.3.4.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Error-Code [OID ref: 1 0 24753 0 127 4 1]
Sensor interface and App-interface:
0 = no error
1 = error, then this is the only element in the response

Packet-Number [OID ref: 1 0 24753 0 127 4 2]
This is currently RFU in the command structure; therefore the response data is zero

Calibration-Data [OID ref: 1 0 24753 0 127 4 3 {cal-value}] (see11.4 for processing).
Where {cal-value} is one of the values fields in the Calibration record, that are presented in ordinal sequence. Therefore this element, but with successive OID references, is present up to the number of times that a calibrated value is provided. Temperature sensors have three calibrated values, humidity sensors have 17 values.

9.3.5 Write-Sample-And-Configuration-Record

This command is used to set monitoring parameters within the constraints of the simple sensor. From the user perspective these are equivalent to fields 4 to 9 of the Simple Sensor Data Block for sensor types 0000 to 0100 (see 8.3). The actual encoding is stored in the sample and configuration record to enable comparisons with the observations to be made when the sensor is active.

Some implementations of simple sensors have the capability to encode a series of configuration records. The data transferred in this command shall be written as a new record, and the mechanism to achieve this is beyond the scope of this International Standard.

Sensor interface for command and response: 5-bit code = 00101

App-interface: Command = **[OID ref: 1 0 24753 0 126 5]**
Response = **[OID ref: 1 0 24753 0 127 5]**

NOTE The command does not include the UTC timestamp, which is encoded to the rules of a different command (see 9.3.6). These independent commands are to enable the configuration data to be input using offline processes, and the timestamp to be encoded in real time, possibly during a production process.

9.3.5.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Password [OID ref: 1 0 24753 0 126 5 1]
Sensor interface and **App-interface**: BYTE STRING (4, or 8, or 16)
Bits 5 and 6 of the manufacturer record declare the size of the password.

9.3.5.2 Memory-based object identifiers in the command

The command contains the following data elements, some of which are null fields for some types of simple sensor:

— **Sampling-Regime [OID 1 0 24753 0 5 3]** (see 11.2.5 for processing, but using this OID)

— **High-In-Range-Limit [OID 1 0 24753 0 5 4]** (see 11.2.6 for processing, but using this OID)

— **Low-In-Range-Limit [OID 1 0 24753 0 5 5]** (see 11.2.7 for processing, but using this OID)

— **Monitor-delay [OID 1 0 24753 0 5 6]** (see 11.2.8 for processing, but using this OID)

— **High-Out-Of-Range-Alarm-Delay [OID 1 0 24753 0 5 7]** (see 11.2.9 for processing, but using this OID)

— **Low-Out-Of-Range-Alarm-Delay [OID 1 0 24753 0 5 8]** (see 11.2.10 for processing, but using this OID)

9.3.5.3 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 0 127 5 1]
Sensor interface
00 = success
01 = sensor not identified (e.g. password mismatch)
10 = erase of previous event and/or time synchronisation not applied
11 = parameter(s) are not logical

App-interface: INTEGER (0..3) or TEXT STRING

9.3.6 Initialise-Sensor-Monitoring

This command delivers the UTC timestamp and initialises the sensor to start monitoring its environment.

Sensor interface for command and response: 5-bit code = 00110

App-interface: Command = **[OID ref: 1 0 24753 0 126 6]**
Response = **[OID ref: 1 0 24753 0 127 6]**

9.3.6.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Password [OID ref: 1 0 24753 0 126 6 1]
Sensor interface and **App-interface**: BYTE STRING (4, or 8, or 16)
Bits 5 and 6 of the manufacturer record declare the size of the password.

9.3.6.2 Memory-based object identifier in the command

The command contains the following data element:

UTC-Timestamp-At-Configuration [OID ref: 1 0 24753 0 5 2] (see 6.4 for processing)
The requirement is to incorporate the 32-bit UTC timestamp in the command, either by converting from the human understandable date and time structure, or by direct transfer of a network UTC 32-bit timestamp. Care needs to be taken to ensure that the epoch is 1970-01-01.

9.3.6.3 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 0 127 6 1]
Sensor interface
00 = success
01 = sensor not identified (e.g. password mismatch)
10 = erase of previous event and/or time synchronisation not applied
11 = RFU

App-interface: INTEGER (0..3) or TEXT STRING

9.3.7 Read-Sample-And-Configuration-Record

This command returns the content of the sample and configuration record, which consists of:

a) the UTC timestamp at point of manufacture, which is permanently encoded

b) the UTC timestamp and configuration data at point of configuration

c) optionally, additional UTC timestamp and configuration data, with the earliest encode first

Bit 24 of the manufacturer record identifies whether option c is supported. If it is, then the set of configuration timestamp and data is locked so that on the next round of configuration on the re-use of the sensor the history is maintained. The different structures have implications for interpreting the data.

Sensor interface for command and response: 5-bit code = 00111

App-interface: Command = **[OID ref: 1 0 24753 0 126 7]**
Response = **[OID ref: 1 0 24753 0 127 7]**

9.3.7.1 Command-based object identifiers

The following command argument and object identifier apply:

Response-Type [OID ref: 1 0 24753 0 126 7 2]
Sensor and **App-interface**
0 = Respond with the complete block in one transmission
1 = RFU

9.3.7.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Error-Code [OID ref: 1 0 24753 0 127 7 1]
Sensor interface and App-interface:
0 = no error
1 = error, then this is the only element in the response

Packet-Number [OID ref: 1 0 24753 0 127 7 2]
This is currently RFU in the command structure; therefore the response data is zero

9.3.7.3 Memory-based object identifiers in the response

The response contains the following data elements:

— **UTC-Timestamp-At-Manufacture [OID ref: 1 0 24753 0 5 1]** (see 6.4 for processing)

— **Number-Of-Configurations-Recorded [OID ref: 1 0 24753 0 5 2]**

— **UTC-Timestamp-At-Configuration [OID ref: 1 0 24753 0 5 3 {seq}]** (see 11.5 for processing), where the {seq} arc identifies the sequence (1 to 4) of the encoded timestamp, with 1 for the earliest or only set.

— **Sampling-Regime [OID 1 0 24753 0 5 4 {seq}]** (see 11.2.5 for processing, but using this OID)

— **High-In-Range-Limit [OID 1 0 24753 0 5 5 {seq}]** (see 11.2.6 for processing, but using this OID)

— **Low-In-Range-Limit [OID 1 0 24753 0 5 6 {seq}]** (see 11.2.7 for processing, but using this OID)

— **Monitor-delay [OID 1 0 24753 0 5 7 {seq}]** (see 11.2.8 for processing, but using this OID)

— **High-Out-Of-Range-Alarm-Delay [OID 1 0 24753 0 5 8 {seq}]** (see 11.2.9 for processing, but using this OID)

— **Low-Out-Of-Range-Alarm-Delay [OID 1 0 24753 0 5 9 {seq}]** (see 11.2.10 for processing, but using this OID)

For fields 3 to 9, the {seq} arc identifies the sequence (1 to 4) of the encoded configuration data, with 1 for the earliest or only set. If a series of configuration records is encoded on the record then fields 3 to 9 for the first sequence shall be presented before these fields for the second sequence and so forth. This results in the current configuration being in the last sequence.

9.3.8 Read-Event-Record

This command returns the content of the event record, consisting of:

a) A counter value of the highest sample count

b) A counter value of the highest encode packet

c) Pairs of sample count and observed data as recorded based on the criteria for recording samples.

Sensor interface for command and response: 5-bit code = 01000

App-interface: Command = **[OID ref: 1 0 24753 0 126 8]**
Response = **[OID ref: 1 0 24753 0 127 8]**

The size of the data is based on the size of the data transmission. Because this might not be on an 8-bit length, or even an odd number length, the sequence of packets in the response need to be concatenated in the correct sequence to enable the data to be decoded.

If an alarm has been triggered, the memory is locked at that event. This means that the sample counter has a value that can be converted to the time that this event took place.

9.3.8.1 Command-based object identifiers

The following command argument and object identifier apply:

Response-Type [OID ref: 1 0 24753 0 126 8 1]
Sensor interface
00 = Respond with the first 24 bits
01 = Respond with the complete block in one transmission
10 = RFU
11 = RFU

App-interface: INTEGER (0..3)

9.3.8.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Error-Code [OID ref: 1 0 24753 0 127 8 1]
Sensor interface and **App-interface:**
0 = no error
1 = error, then this is the only element in the response

Packet-Number [OID ref: 1 0 24753 0 127 8 2]
This is currently RFU in the command structure; therefore the response data is zero

9.3.8.3 Memory-based object identifiers in the response

The response contains the following data elements:

— **Current-Sample-Counter [OID ref: 1 0 24753 0 6 1]** (see 11.9 for processing)

— **Packet-Counter [OID ref: 1 0 24753 0 6 2]** (see 11.6.1 for processing)

— **Sample-Count-At-Observation [OID ref: 1 0 24753 0 6 3 {seq}]** (see11.6.2 and 11.9for processing)

The {seq} arc identifies the sequence of the encoded sample count number at each recorded observation, with 0 for the earliest set.

— **Observed-Sample-Data [OID ref: 1 0 24753 0 127 8 6 {seq}]** (see11.6.2 and 11.9for processing)

The {seq} arc identifies the sequence of each recorded observation, with 0 for the earliest set. The length of the data is determined by the size of the transmitted data

Impact and tilt sensors are designed to deliver a single BOOLEAN state of within limit or exceeding the limit once, in which case the alarm is triggered. There is no observed data, but the application can consider that as the limit had been exceeded that the event was caused by a value greater than the limit. The time of the event can be calculated from the sensor counter value.

9.3.9 Write-UTC-Timestamp

This command delivers a UTC timestamp that is written to the time synchronisation record.

Sensor interface for command and response: 5-bit code = 01001

App-interface: Command = **[OID ref: 1 0 24753 0 126 9]**
Response = **[OID ref: 1 0 24753 0 127 9]**

9.3.9.1 Command-based object identifiers

The following command argument and object identifier apply:

Current-UTC-Timestamp [OID ref: 1 0 24753 0 126 9 1] (see 6.4 for processing)
Sensor and **application interface:** BIT STRING (32)
The requirement is to incorporate the 32-bit UTC timestamp in the command, either by converting from the human understandable date and time structure, or by direct transfer of a network UTC 32-bit timestamp. Care needs to be taken to ensure that the epoch is 1970-01-01.

9.3.9.2 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 0 127 9 1]
Sensor interface
00 = success
01 = sensor not identified
10 = time synchronisation record full
11 = RFU

App-interface: INTEGER (0..3) or TEXT STRING

9.3.10 Read-Time-Synchronisation-Record

This command is used to read the time determined by the sensor's real time clock compared with the accurate UTC timestamp derived from a network and delivered across the air interface. The number of records will depend on the opportunities for the **Write-UTC-Timestamp** command to be applied to the sensor during its monitoring cycle.

Sensor interface for command and response: 5-bit code = 01010

App-interface: Command = **[OID ref: 1 0 24753 0 126 10]**
Response = **[OID ref: 1 0 24753 0 127 10]**

9.3.10.1 Command-based object identifiers

The following command argument and object identifier apply:

Response-Type [OID ref: 1 0 24753 0 126 10 1]
Sensor and **application interface**
0 = Respond with the complete block in one transmission
1 = RFU

9.3.10.2 Response-based object identifiers

The following response arguments and object identifiers apply:

Error-Code [OID ref: 1 0 24753 0 127 10 1]
Sensor interface and App-interface:
0 = no error
1 = error, then this is the only element in the response

Packet-Number [OID ref: 1 0 24753 0 127 10 2]
This is currently RFU in the command structure; therefore the response data is zero

9.3.10.3 Memory-based object identifiers in the response

The response contains the following data elements:

— **RTC-Time [OID ref: 1 0 24753 0 7 1 1 {seq}]** [CONDITIONAL] (see 11.7 for processing)
This data element is conditional on bit 25 of the manufacturer record = 0
The {seq} arc identifies the sequence of the encoded RTC-time at each synchronisation point of time, with 0 for the earliest set.

— **RTC-Sample-Count [OID ref: 1 0 24753 0 7 1 2 {seq}]** [CONDITIONAL] (see 11.7 for processing)
This data element is conditional on bit 25 of the manufacturer record = 1
The {seq} arc identifies the sequence of the encoded RTC sample count number at each synchronisation point of time, with 0 for the earliest set.

— **UTC-Network-Time [OID ref: 1 0 24753 0 7 2 {seq}]** [CONDITIONAL] (see 11.7 for processing)

The {seq} arc identifies the sequence of the encoded RTC-time at each synchronisation point of time, with 0 for the earliest set.

9.3.11 Erase-Monitored-Data

This command is used to invoke an erasing process of the event record and or the time synchronisation record.

Sensor interface for command and response: 5-bit code = 01011

App-interface: Command = **[OID ref: 1 0 24753 0 126 11]**
Response = **[OID ref: 1 0 24753 0 127 11]**

9.3.11.1 Command-based object identifiers

The following command arguments and object identifiers apply:

Password [OID ref: 1 0 24753 0 126 11 1]
Sensor interface and **app-interface**: BYTE STRING (4, or 8, or 16)
Bits 5 and 6 of the manufacturer record declare the size of the password.

Selected-Block [OID ref: 1 0 24753 0 126 11 2]
Sensor interface
00 = RFU
01 = Event record block
10 = Time synchronisation block
11 = Both blocks

App-interface: INTEGER (0..3)

9.3.11.2 Response-based object identifiers

The following response argument and object identifier apply:

Response-Code [OID ref: 1 0 24753 0 127 11 1]
Sensor interface
00 = success
01 = sensor not identified (e.g. password mismatch)
10 = erase incomplete
11 = RFU

App-interface: INTEGER (0..3) or TEXT STRING

9.3.12 Activate-Simple-Sensor

This command enables the Simple Sensor to be changed from a disabled state (without sampling the sensor output) to invoke sensor monitoring when required. The Activate-Simple-Sensor command starts the Simple Sensor “Monitor Delay” timer and starts sampling the sensor output after the monitor delay time has elapsed.

Sensor interface for command and response: 5-bit code = 01100

App-interface: Command = **[OID ref: 1 0 24753 0 126 12]**
Response = **[OID ref: 1 0 24753 0 127 12]**

9.3.12.1 Command-based object identifier

The following command argument and object identifier apply:

Password [OID ref: 1 0 24753 0 126 12 1]
Sensor interface and **app-interface**: BYTE STRING (4, or 8, or 16)
Bits 5 and 6 of the manufacturer record declare the size of the password.

9.3.12.2 Response-based object identifier

The following response argument and object identifier apply:

> **Response-Code [OID ref: 1 0 24753 0 127 12 1]**
> > **Sensor interface** and **App-interface**:
> > 0 = success
> > 1 = error

9.3.13 Deactivate-Simple-Sensor

This command enables the Simple Sensor to stop the sensor output sampling operation. This command is essential to avoid alarm condition during the process of re-commissioning a Simple Sensor tag. If no alarm condition occurred during the Simple Sensor monitoring cycle, the deactivated Simple Sensor is identical to unused sensor before activation except for the remaining battery capacity.

> **Sensor interface** for command and response: 5-bit code = 01101

> **App-interface:** Command = **[OID ref: 1 0 24753 0 126 13]**
> Response = **[OID ref: 1 0 24753 0 127 13]**

9.3.13.1 Command-based object identifier

The following command argument and object identifier apply:

> **Password [OID ref: 1 0 24753 0 126 13 1]**
> > **Sensor interface** and **app-interface**: BYTE STRING (4, or 8, or 16)
> > Bits 5 and 6 of the manufacturer record declare the size of the password.

9.3.13.2 Response-based object identifier

The following response argument and object identifier apply:

> **Response-Code [OID ref: 1 0 24753 0 127 13 1]**
> > **Sensor interface** and **App-interface**:
> > 0 = success
> > 1 = error

10 Processing rules for full function sensors based on IEEE 1451.7 type 001

10.1 General

This clause defines the processing rules to interpret the bit-based code structure on IEEE 1451.7 Type 001 sensors into a format that is more meaningful for a business application. It also addresses the process of business requirement into the bit-based format for configuration purposes.

To enable communication with the business application through application commands and the software system infrastructure, each record and each field is identified with an Object Identifier. Each field is represented by an Object Identifier plus the associated data. The rule for converting each piece of data is defined in this clause, using the notation ***app-interface*** for data as presented to and from the application, and ***sensor interface*** for the bit based coding on the sensor.

The following sub-clauses are set out in a sequence based on the way records, and the fields within the record, are presented on the IEEE 1451.7 Type 001 sensor.

10.2 1451.7 sensor ID – 64-bit unique sensor identifier

This read only field has this OID: **1 0 24753 7 1**

Sensor interface:	64-bit code
App-interface:	64-bit code

The ISO/IEC 24753 processor retains this code value support additional communications with the same sensor.

10.3 Primary sensor characteristics TEDS (Type 1)

10.3.1 TEDS type

This read only field has this OID: **1 0 24753 7 2 1**

Sensor interface:	always 001_2
App-interface:	INTEGER, always 1, but generally retained within the ISO/IEC 24753 Processor

NOTE: The other possible values for the TEDS type are reserved for future assignments in IEEE 1451.7.

Processing the sensor data is aborted if the TEDS Type has a value other than 1.

10.3.2 Sensor Type

This read only field has this OID: **1 0 24753 7 2 2**

Sensor interface:	7-bit code
App-interface:	INTEGER (0 127)

The sensor types are defined in an annex of IEEE 1451.7. An implementation of this International Standard retains the value of the sensor type throughout the processing of the sensor, and should provide the descriptions as a basic extension to the code values in any user interface.

10.3.3 Units extension

This read only field has this OID: **1 0 24753 7 2 3**

This field is used to qualify the sensor type code. At the time of publication, the only unites extensions apply to chemical substances and are used to qualify sensor types 4 and 5 that measure the concentration of a substance.

Sensor interface:	5-bit code
App-interface:	INTEGER (0..31)

The extension codes that identify the chemical substances are defined in an annex of IEEE 1451.7. An implementation of this International Standard should provide the descriptions as a basic extension to the code values in any user interface.

Until the IEE 1451.7 units extension codes are modified or extended to have a different meanings than as a description of a chemical substance, the value 00000_2 shall be assigned to sensor types with values other than 4 and 5.

10.3.4 Sensor map

This read only field has this OID: **1 0 24753 7 2 4**

The sensor map defines by bit position in a 16 bit code the measurement type codes supported by the sensor. The list is defined in the IEEE 1451.7 table for the Supported Measurement Type Codes.

Sensor interface:	16-bit code
App-interface:	16-bit code, but generally retained within the ISO/IEC 24753 Processor

The bit positions in the table are to cover the codes 0 to 15, and if the bit is set =1, then the measurement type code is supported by the sensor. An implementation of this International Standard retains the value of the sensor map throughout the processing of the sensor.

10.3.5 Data resolution

This read only field has this OID: **1 0 24753 7 2 5**

Sensor interface:	5-bit code
App-interface:	INTEGER (1..31), but generally retained within the ISO/IEC 24753 Processor

The code identifies the length that is used to convey sensor data for measurement codes. The sensor bit code 00000_2 identifies a data transmission format of 1 bit long and 11111_2 identifies a length of 32 bits.

10.3.6 Scale Factor Significand

This read only field has this OID: **1 0 24753 7 2 6**

The scale factor declares the span of the measurement capabilities of the sensor. This span is expressed as a significand in this field and quantified by the exponent, as defined in the next sub-clause.

Sensor interface:	11-bit value, represented as a two's-complement numeral value.
App-interface:	A value in the range –1.024 to +1.023. This is calculated by converting the binary value to a decimal value and dividing the result by 1000. This is generally retained within the ISO/IEC 24753 Processor.

Converting from the sensor interface can be calculated by following the processes defined in IEEE 1451.7 for the processes defined for the scale factors for transmitted data and the scale factor significand (field 6 of the Primary Sensor Characteristics TEDS (Type 1) record).

10.3.7 Scale Factor Exponent

This read only field has this OID: **1 0 24753 7 2 7**

Sensor interface:	6-bit value, represented as a two's-complement numeral value.
App-interface:	A value in the range –32 to +31. This is generally retained within the ISO/IEC 24753 Processor.

Converting from the sensor interface can be calculated by following the processes defined in IEEE 1451.7 for the processes defined for the scale factors for transmitted data and the scale factor exponent (field 7 of the Primary Sensor Characteristics TEDS (Type 1) record).

10.3.8 Scale Offset Significand

This read only field has this OID: **1 0 24753 7 2 8**

The scale offset declares the lowest measurement capability of the sensor. This offset is expressed as a significand in this field and quantified by the exponent, as defined in the next sub-clause. The highest measurement capability of the sensor can be determined by adding the scale factor value to the offset value.

Sensor interface:	11-bit value, represented as a two's-complement numeral value.

App-interface:	A value in the range –1.024 to +1.023. This is calculated by converting the binary value to a decimal value and dividing the result by 1000. This is generally retained within the ISO/IEC 24753 Processor.

Converting from the sensor interface can be calculated by following the processes defined in IEEE 1451.7 for the processes defined for the scale factors for transmitted data and the scale offset significand (field 8 of the Primary Sensor Characteristics TEDS (Type 1) record).

10.3.9 Scale Offset Exponent

This read only field has this OID: **1 0 24753 7 2 9**

Sensor interface:	6-bit value, represented as a two's-complement numeral value.
App-interface:	A value in the range –32 to +31. This is generally retained within the ISO/IEC 24753 Processor.

Converting from the sensor interface can be calculated by following the processes defined in IEEE 1451.7 for the processes defined for the scale factors for transmitted data and the scale offset exponent (field 9 of the Primary Sensor Characteristics TEDS (Type 1) record).

10.3.10 Data uncertainty

This read only field has this OID: **1 0 24753 7 2 10**

Sensor interface:	3-bit code
App-interface:	Text output, for information, to provide the percentage uncertainty of the transmitted data

Conversion to the App-interface output is achieved by using the mapping table defined in IEEE 1451.7 for data uncertainty (field 10 of the Primary Sensor Characteristics TEDS (Type 1) record).

10.3.11 Sensor Reconfiguration Capability

This read only field has this OID: **1 0 24753 7 2 11**

Sensor interface:	1-bit value	
App-interface:	BOOLEAN	0 = Not supported
		1 = Supported

This information is generally retained within the ISO/IEC 24753 Processor.

10.3.12 Memory Rollover Capability

This read only field has this OID: **1 0 24753 7 2 12**

Sensor interface:	1-bit value	
App-interface:	BOOLEAN	0 = Not supported
		1 = Supported

This information is generally retained within the ISO/IEC 24753 Processor.

10.4 Sampling and Configuration Record

10.4.1 UTC timestamp at configuration

This read-write field has this OID: **1 0 24753 7 3 1**

Sensor interface:	32-bit string
App-interface:	UTC format (e.g. 2008-08-08 08:08:08)

The conversion process is described in 6.4.

10.4.2 Sample interval

This read-write field declares the sample interval in seconds or minutes, so two object identifiers apply on the application interface:

1 0 24753 7 3 2 0 for seconds

1 0 24753 7 3 2 1 for minutes

Sensor interface:	16-bit string, comprising the first bit (0 = seconds, 1 = minutes) and 15 bits indicating the interval in the defined unit of measure.
App-interface:	INTEGER, (0..32767)

The final arc of the Object Identifier identifies the unit of measure (seconds or minutes) and this equates to the value of the first bit as encoded on the sensor. A 16-bit code of all zeros is valid, indicating that sampling is continuous. Therefore the associated Object Identifier is the one indicating seconds and qualified by a 0 value integer.

10.4.3 Monitor delay

This read-write field expresses time in the same way as defined for the sample interval (above). The OIDs are:

1 0 24753 7 3 3 0 for seconds

1 0 24753 7 3 3 1 for minutes

Sensor interface:	16-bit string, comprising the first bit (0 = seconds, 1 = minutes) and 15 bits indicating the interval in the defined unit of measure.
App-interface:	INTEGER, (0..32767)

The final arc of the Object Identifier identifies the unit of measure (seconds or minutes) and this equates to the value of the first bit as encoded on the sensor. A 16-bit code of all zeros is valid, indicating that sampling is continuous. Therefore the associated Object Identifier is the one indicating seconds and qualified by a 0 value integer.

10.4.4 Alarm values set

This read-write field has this OID: **1 0 24753 7 3 4**

Sensor interface:	2-bit string
App-interface:	INTEGER (0..3) The code values are: 0 = none 1 = lower only 2 = upper only 3 = both

As it is a user choice of which alarms to set, an implementation of this International Standard should provide the descriptions of these options as a basic extension to the code values in any user interface.

10.4.5 Memory rollover capability

This read-write field has this OID: **1 0 24753 7 3 5**

Sensor interface:	1-bit string	
App-interface:	BOOLEAN	0 = switched off 1 = switched on

As it is a user choice of which alarms to set, an implementation of this International Standard should provide the descriptions of these options as a basic extension to the code values in any user interface.

10.4.6 Upper alarm threshold

This read-write field has this OID: **1 0 24753 7 3 6**

Sensor interface:	n-bit string, where n is the number of bits for the data resolution (see 10.3.5)
App-interface:	INTEGER

There are n-1 measurement steps in the sensor. Converting from the sensor interface to the application interface can be calculated by following the processes defined in IEEE 1451.7 for the upper alarm threshold value (field 6 of the Sample and Configuration record) and the IEEE 1451.7 clause that defines Scale Factors for Transmitted Data.

10.4.7 Lower alarm threshold

This read-write field has this OID: **1 0 24753 7 3 7**

Sensor interface:	n-bit string, where n is the number of bits for the data resolution (see 10.3.5)
App-interface:	INTEGER

The conversion process is as defined in 10.4.6.

10.5 Event Administration Record

10.5.1 General

The event administration record contains information that allows processes defined in this International Standard to calculate exactly how many sensor words are stored for measurement codes 10, 11, 12, and 13. Additionally, it contains the alarm triggered field and the sample counts needed for measurement codes 6 and 7. The processing for each field is defined in the following sub clauses.

The IEEE 1451.7 standard defines a segment as a group of 32 sensor words. The length of each word for measurement codes 10 and 13 is specified by field 5 of the Primary Characteristics TEDS record. The length of each word for measurement codes 11 and 12 is specified by the sum of the time tick length and the data transmission format given in field 5 of the Primary Characteristics TEDS record. In this way, memory size is given in multiples of one segment. The segment information is important to enable the interpretation of event records where the transmission across the air interface is in whole segments, multiple segments, or partial segments.

10.5.2 Code 10 Sample capacity

This read-only field has the OID: **1 0 24753 7 4 1** and is only present if measurement type 10 is declared as present in field 4 of the Primary Characteristics TEDS record. It declares the size of the encoding capacity in segments.

Sensor interface:	11-bit string
App-interface:	INTEGER

EXAMPLE

Assume an 11-bit data transmission

Assume the Code 10 sample capacity of 12 segments

A single segment comprises 32 X 11 = 352 bits, or 44 bytes

The encoding capacity comprises 352 X 12 = 4224 bits, or 528 bytes

The information of the segment size and number of segments is used in conjunction with information about the reading environment to achieve a balance between the minimum number of air interface transmissions and a robust system.

10.5.3 Code 11 Sample capacity

This read-only field has the OID: **1 0 24753 7 4 2** and is only present if measurement type 11 is declared as present in field 4 of the Primary Characteristics TEDS record. It declares the size of the encoding capacity in segments.

Sensor interface:	3-bit string
App-interface:	INTEGER

EXAMPLE

Assume an 11-bit data transmission

Each event record consists of an 8-bit time tick plus the 11-bit data

Assume the Code 10 sample capacity of 12 segments

A single segment comprises 32 X (8 +11) = 608 bits, or 76 bytes

The encoding capacity comprises 608 X 12 = 7296 bits, or 912 bytes

The information of the segment size and number of segments is used in conjunction with information about the reading environment to achieve a balance between the minimum number of air interface transmissions and a robust system.

10.5.4 Code 12 Sample capacity

This read-only field has the OID: **1 0 24753 7 4 3** and is only present if measurement type 11 is declared as present in field 4 of the Primary Characteristics TEDS record. It declares the size of the encoding capacity in segments.

Sensor interface:	11-bit string
App-interface:	INTEGER

EXAMPLE

Assume an 11-bit data transmission

Each event record consists of a 16-bit time tick plus the 11-bit data

Assume the Code 10 sample capacity of 12 segments

A single segment comprises 32 X (16 +11) = 864 bits, or 108 bytes

The encoding capacity comprises 864 X 12 = 10368 bits, or 1296 bytes

The information of the segment size and number of segments is used in conjunction with information about the reading environment to achieve a balance between the minimum number of air interface transmissions and a robust system.

10.5.5 Code 13 Sample capacity

This read-only field has the OID: **1 0 24753 7 4 4** and is only present if measurement type 10 is declared as present in field 4 of the Primary Characteristics TEDS record. It declares the size of the encoding capacity in segments. As only the sample data is encoded the size of the segment and memory capacity is calculated on the same basis as the example in 10.5.2.

Sensor interface:	11-bit string
App-interface:	INTEGER

10.5.6 Sample count

This read-only field has the OID: **1 0 24753 7 4 5** and is always present in the Event Administration record. It declares the number of the last sampled observation.

Sensor interface:	16-bit string
App-interface:	INTEGER

If the value is 65535, the sample count has reached its capacity and the sampling process could have ceased at some earlier time.

10.5.7 Alarm triggered

This read-only field has the OID: **1 0 24753 7 4 6** and is always present in the Event Administration record. It declares the different alarm values.

Sensor interface:	4-bit string
App-Interface:	TEXT STRING

The text string, which may vary by language and by application, generally follows these constructs. If the bit string is 0000, then the text is "NO-ALARMS". For other bit strings, the text is determined by the bit position having a "1" value, with the text being concatenated for the component parts, as follows:

1st bit = 1, then the text is "UPPER-ALARM"

2nd bit = 1, then the text is "LOWER-ALARM"

3rd bit = 1, then the text is "MEMORY-FULL"

4th bit = 1, then the text is "LOW-BATTERY"

10.5.8 Sample count at predetermined time

This field has the OID **1 0 24753 7 4 7** and is only present for measurement type code 6. The field can be written to at the time of configuration using the **Write-Event-Admin-Field7** command. It may also be part of a read response for reading the Event-Administration record.

Sensor interface:	16-bit string
App-interface:	INTEGER

This field simply determines the point in time, expressed as a sample count, when a single sample is taken and recorded.

For encoding, the sample count value can be determined by deciding the data and time of the event, subtracting the time at configuration, and dividing the result by the sample interval.

For decoding, the time of this event can be calculated by multiplying the sample count by the sample interval, and adding the date and time of configuration.

10.5.9 Sample count at critical event

This read-only field has the OID: **1 0 24753 7 4 8** and is only present for measurement type code 7. This field simply determines the point in time, expressed as a sample count, when a threshold limit had been cross for the first time. If an alarm has net been triggered the value is zero.

Sensor interface:	16-bit string
App-interface:	INTEGER

For decoding, the time of this event can be calculated by multiplying the sample count by the sample interval, and adding the date and time of configuration.

10.5.10 Sample count of events outside either threshold

This read-only field has the OID: **1 0 24753 7 4 9** and is only present for measurement type codes 11 and 12. If a threshold has not been crossed the value is zero. The count is of the total number of observations outside either or both limits depending on how the limits are set.

Sensor interface:	16-bit string
App-interface:	INTEGER

10.5.11 Sample count at the first threshold event

This read-only field has the OID: **1 0 24753 7 4 10** and is only present for measurement type code 13. This field simply determines the point in time, expressed as a sample count, when a threshold limit had been cross for the first time, enabling all subsequent observations to be recorded without the additional requirement of a timestamp. If a threshold has not been crossed the value is zero.

Sensor interface:	16-bit string
App-interface:	INTEGER

For decoding, the time of this event can be calculated by multiplying the sample count by the sample interval, and adding the date and time of configuration. Each subsequent recorded observation is indexed by one sample interval over the previous observation.

10.6 Event records

10.6.1 General

The existence of memory to encode a particular event record is determined by sensor's support for the associated measurement type code. Some of these measurement type codes only support the recording of one instance of a record, which can be based on a single event, or the record can be continually updated as sampling takes place. Either way there is only one value set in the record. The other main class of event records requires new data to be added into memory based on the sample and configuration criteria that had been set, resulting in multiple instances. This process continues until the sampling process ends, or when memory becomes full, or if memory becomes full and rollover is invoked until sampling is completed but with only the most recent records retained in memory.

10.6.2 Converting the observed data from binary to real values

The conversion process is as defined in 10.4.6.

10.6.3 CRC-16 validation for event records transmitted in packets of more than one segment

If the air interface response for multiple event records comprises more than one segment then each segment has an appended CRC-16 calculated by the sensor, using the using the polynomial $x_{16} + x_{12} + x_5 + 1$. The CRC-16 shall be validated to prove the data validity of the segment content. If proven, the CRC-16 shall be stripped off and the data processed as defined in 10.6.14 to 10.6.17. If the CRC-16 validation fails, a transmission error has probably occurred and the damaged segment(s) need to be retrieved with a new air interface command. The response to the application contains the list of segment that failed the CRC-16 validation. It is beyond the scope of this International Standard to define any recovery procedure, other than to re-invoke the necessary sensor commands to retrieve the segments that were rejected.

If a single segment is transferred across the air interface, the sensor does not generate a CRC-16, because the error detection is achieved by the air interface response CRC-16, which is stripped off by the interrogator.

10.6.4 Present-Value

This read-only field has the OID: **1 0 24753 7 5 0** and is only present for measurement type code 0.

Sensor interface:	n-bit string, where n is the number of bits in the data transmission format
App-interface:	INTEGER

Conversion from binary to decimal is as defined in 10.4.6.

10.6.5 Maximum-Value

This read-only field has the OID: **1 0 24753 7 5 1** and is only present for measurement type code 1.

Sensor interface:	n-bit string, where n is the number of bits in the data transmission format
App-interface:	INTEGER

Conversion from binary to decimal is as defined in 10.4.6.

10.6.6 Minimum-Value

This read-only field has the OID: **1 0 24753 7 5 2** and is only present for measurement type code 2.

Sensor interface:	n-bit string, where n is the number of bits in the data transmission format
App-interface:	INTEGER

Conversion from binary to decimal is as defined in 10.4.6.

10.6.7 Average-Value

This read-only field has the OID: **1 0 24753 7 5 3** and is only present for measurement type code 3.

Sensor interface:	n-bit string, where n is the number of bits in the data transmission format
App-interface:	INTEGER

Conversion from binary to decimal is as defined in 10.4.6.

10.6.8 Variance

This read-only field has the OID: **1 0 24753 7 5 4** and is only present for measurement type code 4.

Sensor interface:	n-bit string, where n is the number of bits in the data transmission format
App-interface:	INTEGER

Conversion from binary to decimal is as defined in 10.4.6.

10.6.9 Standard-Deviation

This read-only field has the OID: **1 0 24753 7 5 5** and is only present for measurement type code 5.

Sensor interface:	n-bit string, where n is the number of bits in the data transmission format
App-interface:	INTEGER

Conversion from binary to decimal is as defined in 10.4.6.

10.6.10 Observed-Value-Predetermined-Time

This read-only field has the OID: **1 0 24753 7 5 6** and is only present for measurement type code 6.

Sensor interface:	n-bit string, where n is the number of bits in the data transmission format
App-interface:	INTEGER

Conversion from binary to decimal is as defined in 10.4.6.

10.6.11 Observed-Value-Critical-Event

This read-only field has the OID: **1 0 24753 7 5 7** and is only present for measurement type code 7.

Sensor interface:	n-bit string, where n is the number of bits in the data transmission format
App-interface:	INTEGER

Conversion from binary to decimal is as defined in 10.4.6.

10.6.12 Count-Over-Upper-Threshold

This read-only field has the OID: **1 0 24753 7 5 8** and is only present for measurement type code 8, and is a simple count of the number of occurrences of samples above the upper threshold.

Sensor interface:	8-bit string
App-interface:	INTEGER

10.6.13 Count-Below-Lower-Threshold

This read-only field has the OID: **1 0 24753 7 5 9** and is only present for measurement type code 9, and is a simple count of the number of occurrences of samples below the lower threshold.

Sensor interface:	8-bit string
App-interface:	INTEGER

10.6.14 Data-Log-All-Samples

This read-only field is only present for measurement type code 10, and consists of a set of observations at every sample interval. Each observation is encoded in a sensor word of a length equivalent to the data transmission format.

The words are encoded in segments of 32 sensor words. The sensor word has the value 0 to 31. The segment has the ordinal value 0 to 2047, but with the maximum number of segments for the particular sensor declared by the Code10 Sample Capacity in the Event Administration record (see 10.5.2).

If the air interface response comprises more than one segment, then each segment has a CRC-16 appended by the sensor processes to enable error detection. The processing of this CRC-16 is defined in 10.6.3.

The OID structure for an individual sample observation is:

1 0 24753 7 5 10 {segment} {sensor-word}

The data associated with each OID is as follows:

Sensor interface:	n-bit string, where n is the number of bits in the data transmission format
App-interface:	INTEGER

Conversion from binary to decimal is as defined in 10.6.2.

To determine how many potential segments have been encoded, calculate:

[(sample count) – 1] / 32

where sample count is given by field 5 of the Event Administration record.

If the number of potential segments is greater than the encoding capacity (field 1 of the Event Administration record), then the memory is full and all segments are encoded.

If all segments are fully encoded and memory rollover capability (field 5 of the Sample and Configuration record) is set = 1, then rollover has been applied by the sensor processes, and only the most recent sample observations have been retained in memory. If this is not the case, then the history is truncated at the sample interval where the memory capacity was reached.

The re-construction of the history is beyond the scope of this International Standard, but with this guidance:

— If the memory capacity is not exceeded, then the set of sensor words (in word and segment sequence) represent the observations from sample 0 to sample number (sample count – 1)

— If the memory capacity is full and memory roll over capability has not been applied, then the set of sensor words (in word and segment sequence) represent the observations from sample 0 to sample number ((encoding capacity X 32) – 1).

— If the memory capacity is full and memory rollover capability has been applied, then the set of sensor words (in word and segment sequence) represents the most recent observations. The indexed sequence of one segment will probably contain the most recent observation immediately followed by the earliest retained observation. The indices can be calculated using the equation in the IEEE 1451.7 clause for Memory Rollover Support for Measurement Code 10.

10.6.15 Data-Log-Outside-Threshold-8bitSampleCount

This read-only field is only present for measurement type code 11, and consists of a set of observations outside of either limit, prepended by an 8-bit sample count value. Each observation is encoded in a sensor word of a length equivalent to the data transmission format plus 8 bits.

The words are encoded in segments of 32 sensor words. The sensor word has the value 0 to 31. The segment has the ordinal value 0 to 7, but with the maximum number of segments for the particular sensor declared by the Code11 Sample Capacity in the Event Administration record (see 10.5.3).

If the air interface response comprises more than one segment, then each segment has a CRC-16 appended by the sensor processes to enable error detection. The processing of this CRC-16 is defined in 10.6.3.

The OID structure for an individual sample count is:

1 0 24753 7 5 11 1 {segment} {sensor-word}

For decoding, the time of this event can be calculated by multiplying the sample count by the sample interval, and adding the date and time of configuration.

The data representation of each sample count is as follows:

Sensor interface:	8-bit string
App-interface:	INTEGER

The OID structure for an individual sample observation is:

1 0 24753 7 5 11 2 {segment} {sensor-word}

The data associated with each sample observation OID is as follows:

Sensor interface:	n-bit string, where n is the number of bits in the data transmission format
App-interface:	INTEGER

Conversion from binary to decimal is as defined in 10.6.2.

To determine how many potential segments have been encoded, calculate:

[(sample count of events outside either limit) – 1] / 32

where sample count of events outside either limit is given by field 9 of the Event Administration record

If the number of potential segments is greater than the encoding capacity (field 2 of the Event Administration record), then the memory is full and all segments are encoded.

If all segments are fully encoded and memory rollover capability (field 5 of the Sample and Configuration record) is set = 1, then rollover has been applied by the sensor processes, and only the most recent sample observations have been retained in memory. If this is not the case, then the history is truncated at the sample interval where the memory capacity was reached.

The re-construction of the history is beyond the scope of this International Standard, but with this guidance:

— This event record is only capable of recording observations until sample count number 255. If the sample count value (field 5 of the event administration record) is greater than 255, then records ceased earlier in the sampling process.

— If the memory capacity is not exceeded, then the set of sensor words (in word and segment sequence) represent the observations from the first out-of-limit observation to the last out-of-limit observation (with a sample count no greater than 255).

— If the memory capacity is full and memory roll over capability has not been applied, then the set of sensor words (in word and segment sequence) represent the observations from the first out-of-limit observation to the final out-of-limit observation capable of being recorded.

— If the memory capacity is full and memory rollover capability has been applied, then the set of sensor words (in word and segment sequence) represents the most recent observations. The indexed sequence of one segment will probably contain the most recent observation immediately followed by the earliest retained observation. The indices can be calculated using the equation in the IEEE 1451.7 clause for Memory Rollover Support for Measurement Code 11.

10.6.16 Data-Log-Outside-Threshold-16bitSampleCount

This read-only field is only present for measurement type code 12, and consists of a set of observations outside of either limit, prepended by a 16-bit sample count value. Each observation is encoded in a sensor word of a length equivalent to the data transmission format plus 16 bits.