INTERNATIONAL STANDARD

ISO/IEC 15961-4

First edition
2016-08-01

Information technology — Radio frequency identification (RFID) for item management: Data protocol —

Part 4: Application interface commands for battery assist and sensor functionality

Technologies de l'information — Identification par radiofréquence (RFID) pour la gestion d'objets: Protocole de données —

Partie 4: Commandes de l'interface d'application pour l'assistance de la batterie et la fonctionnalité du capteur

Reference number
ISO/IEC 15961-4:2016(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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

A list of all parts in the ISO/IEC 15961 series can be found on the ISO website.

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. In special cases, when the sensor is attached to or integrated within an RFID tag, this 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 shall 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 this document 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 document 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 may, independently, evolve to support additional air interface protocols and different types of sensors.

The documents that comprise the data protocol are the following.

— ISO/IEC 15961-1 defines the transfer of data to and from the application, supported by appropriate application commands and responses.

— ISO/IEC 15961-2 defines the registration procedure of data constructs to ensure that as new applications adopt the data protocol, it becomes a relatively straightforward process to support that application. This can be achieved by the registration authority publishing regular updates of the RFID data constructs that have been assigned, and for a means of incorporating these updates into the processes of ISO/IEC 15961-1.

— ISO/IEC 15961-3 defines the data constructs and the rules that govern their use.

— ISO/IEC 15961-4 defines the transfer of sensor data to and from the application, supported by appropriate application commands and responses.

— ISO/IEC 15962 specifies the overall process and the methodologies developed to format the application data into a structure to store on the RFID tag.

— ISO/IEC 24753 specifies the overall process and methodologies developed to format and process sensory information in a standardised manner and provide an interface with the appropriate air interface protocol.

Information technology — Radio frequency identification (RFID) for item management: Data protocol —

Part 4: Application interface commands for battery assist and sensor functionality

1 Scope

This document provides a set of application commands and their associated responses for the following functions:

— to start and stop battery assistance;

— to select and de-select a particular sensory function supported by the RFID tag;

— to set sensor parameters both initially and ongoing;

— to start and stop the sensor monitoring the environment;

— to access sensor data;

— to establish the battery status.

ISO/IEC 24753 defines the encoding rules for identifying sensors, their functions, their delivered measurements, and the processing rules for sensor data. As such, it receives commands as defined in this document and provides the information that is required for the appropriate responses.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 18000-63, *Information technology — Radio frequency identification for item management – Part 63: Parameters for air interface communications at 860 MHz to 960 MHz Type C*

ISO/IEC 18000-64, *Information technology — Radio frequency identification for item management — Part 64: Parameters for air interface communications at 860 MHz to 960 MHz Type D*

ISO/IEC 24753:2011, *Information technology — Radio frequency identification (RFID) for item management — Application protocol: encoding and processing rules for sensors and batteries*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762, ISO/IEC IEEE 21451-7, ISO/IEC 24753, and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

3.1
sensor processor
implementation of the processes specified in ISO/IEC 24753 to convert between data and information relevant to the application layer and the bit based encoding on the sensor memory

4 Conformance

4.1 General

The commands and responses in this document are only expressed in an abstract syntax. Their structure is determined by the records and fields on the particular sensor. As such, conformance to this document for a particular sensor system is specifically indicated by the resultant proper encoding according to ISO/IEC 24753 and then passed through RFID air interface protocols to the sensor.

The arguments and fields contained in individual commands and responses identify what needs to be taken into account for correct input to the ISO/IEC 24753 Sensor Processor to achieve a valid encoding. Also, they identify what an application expects to have returned following access to a sensor on an RFID tag. Because of the way the protocol is structured, the commands and responses specified in this document are, to a large extent, independent of particular RFID tag types that support sensors. The effect of this is that ISO/IEC 24753 can specify conformance requirements for valid encoding, which this document cannot.

All the commands and arguments, and their associated processes, are specified in detail in ISO/IEC 24753. Object Identifiers are used throughout that document to uniquely identify arguments within commands and responses for each type of sensor. Object Identifiers are also used to identify fields with particular sensor records.

4.2 Conformance of the Sensor Processor

The Sensor Processor is, effectively, the implementation of ISO/IEC 24753. An implementation of ISO/IEC 24753 is required to 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.

4.3 Application conformance

An application is expected to support the commands and responses that are defined in ISO/IEC 24753 for full function sensors and/or simple sensors. Therefore, this document shall support either one or both options a) and b) in 4.2 as determined by the implementation of ISO/IEC 24753 with which it interfaces.

In addition, the application conformance requirements defined by the commands and responses in this document may be simplified to address a specific type of simple or full function sensor, even to the extent of only the records and commands required for that sensor. For the commands that are supported, all the arguments in the command and response shall be supported to achieve the interface with the sensor processor.

5 Logical interface model

5.1 General

The processes defined in this document are implemented between the application and the air interface protocol. This document performs similar functions for sensory data as ISO/IEC 15961-1 does for item-related data. The relationship and basic functions of the standards are illustrated in Figure 1.

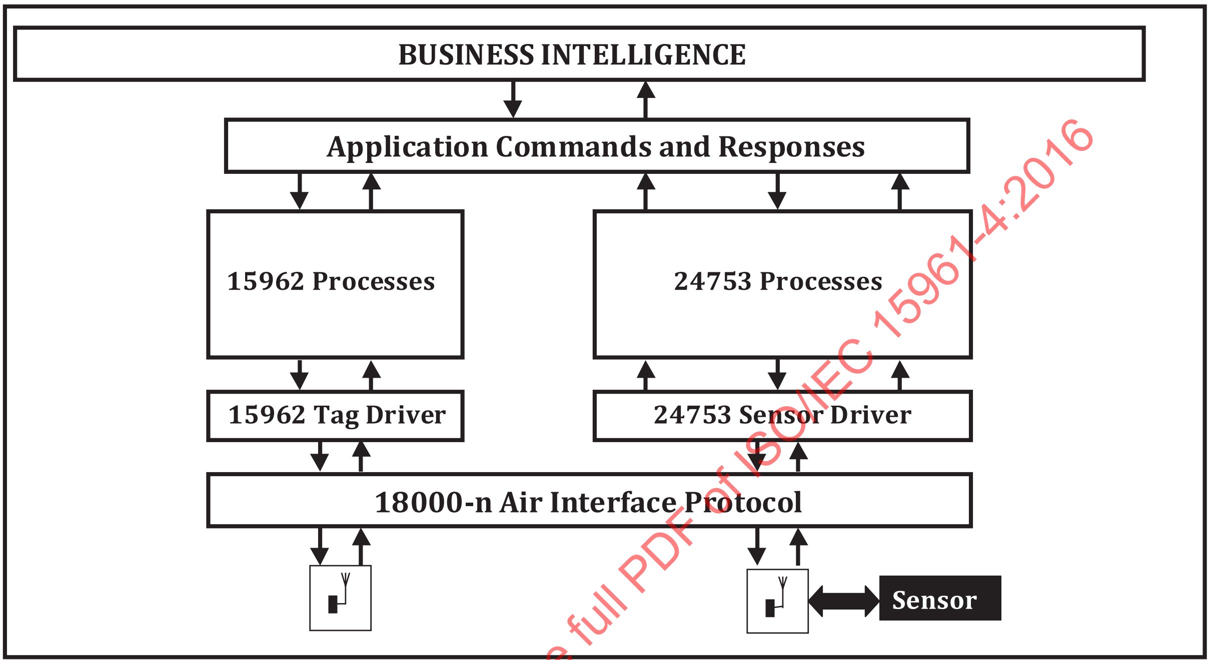

Figure 1 — Basic application interface model

ISO/IEC 24753 is an essential reading in implementing this document. Reference needs to be made to that standard for a full description of the component parts of the model relevant to sensors and batteries. An overview relevant to this document is provided below.

5.2 Application commands

A set of functional application commands is 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. These are defined in Clause 6 for simple sensors and Clause 7 for full function sensors.

The structure of the application commands and response can be determined by clauses in ISO/IEC 24753 that use the same name. The structure of these commands may be derived from the set of Object Identifiers applicable for each command and response as specified in ISO/IEC 24753. Because of this, only selected application interface commands are fully described in this document.

5.3 The sensor information model for full function sensors

The sensor information model for full function sensors (Figure 2) shows the relationship between component processes and structures described later in ISO/IEC 24753 for full function sensors specified in ISO/IEC IEEE 21451-7. 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.

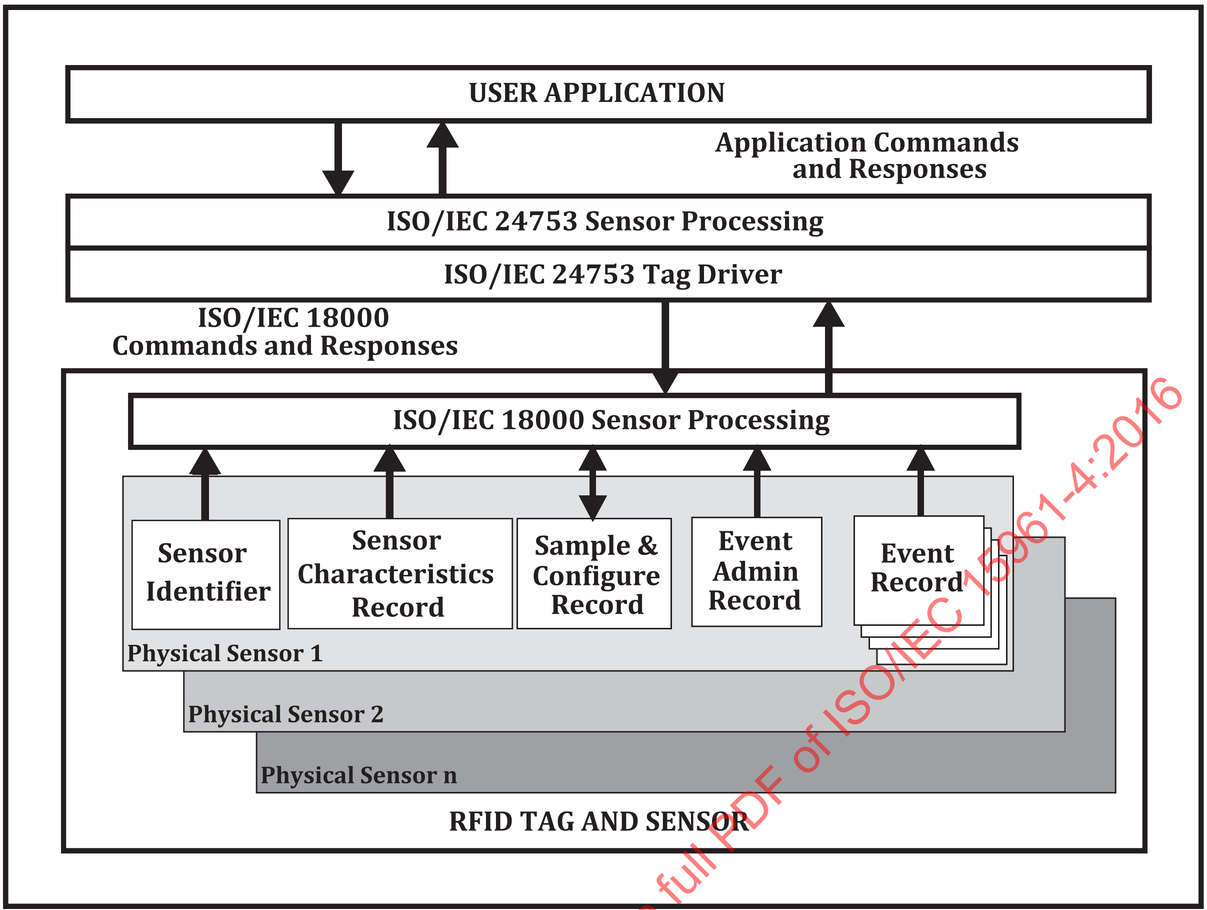

Figure 2 — Sensor information model for full function sensors

Figure 2 clearly illustrates that the commands and responses defined in this document need to be able to communicate with the Sensor Processor, which is the implementation of ISO/IEC 24753. In turn, the specific arguments within the commands and responses need to comply with the requirements of the five sensor records:

— Record 1: Sensor identifier;

— Record 2: Sensor characteristics record;

— Record 3: Sample and configuration record;

— Record 4: Event admin record;

— Record 5: Event record.

The commands are described in Clause 6.

5.4 The sensor information model for simple sensors

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 restrict parameter setting from a fully open systems application, but allow data to be captured using open system air interface commands and processes.

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. Currently, this is 32-bits long, but provision exists for a maximum length of 48-bits.

There are two formats of simple sensor. The memory-mapped simple sensor supports only the simple sensor data block, which is on the same integrated circuit platform as the data on the RFID tag. The ported simple sensor supports additional mandatory and optional records, as detailed in the list below. An annex of ISO/IEC 18000-63 defines the requirements for processing these records if present on the ported simple sensor.

NOTE ISO/IEC 18000-64 does not support ported simple sensors.

The sequence of records is as follows:

— Record 1: Simple sensor data block (mandatory for both implementations);

— Record 2: Manufacturer record (mandatory only for the ported simple sensor);

— Record 3: Authorization password record (optional for the ported simple sensor);

— Record 4: Calibration record (recommended for the ported simple sensor);

— Record 5: Sample and configuration record (mandatory only for the ported simple sensor);

— Record 6: Event record (recommended for the ported simple sensor);

— Record 7: Time synchronisation record (mandatory only for the ported simple sensor and only if the event record is present).

6 Simple sensor commands

6.1 Current air interface reference

The processing of commands (and responses) for simple sensors is specified in ISO/IEC 24753, which uses Object Identifiers to identify the specific arguments. As such, it is possible in this document to specify the structure of commands and responses in a manner that does not depend on the existence of a particular type of simple sensor. There can only be 16 different types of simple sensor, and the sensor manufacturer permanently encodes a 3-bit binary value into a predefined location in the sensor memory to identify the sensor type. In turn, the type code is included as a specific arc in the Object Identifier.

Simple sensors as specified in ISO/IEC 18000-63 and ISO/IEC 18000-64 are used throughout this document to describe arguments and processes. Later versions of these air interface protocols need to be checked for type codes not addressed here (see the current list in 6.3.1.1). If the basic design for simple sensors is maintained in the air interface protocol and in ISO/IEC 24753, then this document can persist. However, the introduction of a possible 48-bit simple sensor can only be supported with a revision to this document.

6.2 Memory mapped simple sensors

The encoding for configuring and reading memory-mapped simple sensors is specified in the ISO/IEC 18000 series of standards that support such sensors. The current air interface protocols that support the memory-mapped simple sensors to achieve this are as follows.

— For ISO/IEC 18000-63 (Type C), standard read and write commands are used in addressing the relevant memory bank to transfer the bit string representing the simple sensor data block. The simple sensor data block can also be transmitted as part of the reply to the *ACK* command, where it is appended to the unique item identifier encoded in memory bank 1.

— The simple sensor data block is transmitted as part of the data packet for an ISO/IEC 18000-64 (Type D) tag.

ISO/IEC 24753 defines two commands for memory mapped simple sensors:

— **Write-Simple-Sensor-Data-Block** command;

— **Read-Simple-Sensor-Data-Block** command.

However, no processes are specified to achieve the bit string that needs to be transferred via the air interface write command, nor any rules to interpret these bits when the simple sensor data block is read from the RFID tag. The application interface functions required are identical to two commands more fully defined for ported simple sensors in this document. Therefore, the equivalent command and response defined in this document (see 6.3.1 and 6.3.2) and the associated processes in ISO/IEC 24753 should be applied to the memory-mapped simple sensors. The memory mapped **Write-Simple-Sensor-Data-Block** command is directly equivalent to the ported simple sensor **Write-Sample-And-Configuration-Record** command (see 6.3.1). The memory-mapped **Read-Simple-Sensor-Data-Block** command is directly equivalent to the ported simple sensor command of the same name (see 6.3.1).

6.3 Ported simple sensors

6.3.1 Write-Sample-And-Configuration-Record

6.3.1.1 Write-Sample-And-Configuration-Record command

The **Write-Sample-And-Configuration-Record** command is used to write user-controlled parameters to a simple sensor, either for the initial mission or to reconfigure on a subsequent mission. The command cannot be invoked for reconfiguration if any of the alarm bits has been set. The command is only concerned with providing input that will result in the encoding of bits 22 to 4 (where bit 22 is MSB) of the simple sensor data block.

This command applies to both types of simple sensor: memory-mapped and ported simple sensor.

Before this command can be invoked, it is necessary to read the simple sensor data block on the tag. This can be achieved by invoking the **Read-Simple-Sensor-Data-Block** command (see 6.3.2) and ignoring all but these three fields, represented by the encoding in bits 31 to 23:

— sensor type, for which the following type codes apply:

  — Type 0 for temperature sensors with a span of 14 °C;

  — Type 1 for temperature sensors with a span of 28 °C;

  — Type 2 for relative humidity sensors;

  — Type 3 for impact sensors;

  — Type 4 for tilt sensors;

— measurement span;

— accuracy.

The **Password** argument is conditional because it only applies to some ported simple sensors. If the password is not known, then its size can be determined by invoking the **Read-Manufacturer's-Record** command where bits 5 and 6 declare the size of the password. The password is a write-once process and is not readable. Therefore, to process the configuration command when a password is set on the ported simple sensor, it is essential to match the password both in terms of length and value.

The **Sampling-Regime** argument applies only to temperature and humidity simple sensors. It defines one of 16 sampling intervals, ranging from 5 min to 8 h. The definition of the sample intervals and the mapping between the bit-based codes and the presentation in the application is given in tables in annexes of ISO/IEC 18000-63 and ISO/IEC 18000-64.

The **High-In-Range-Limit** and **Low-In-Range-Limit** arguments apply only to temperature and humidity simple sensors, and the mapping between the bit-based codes and the presentation to the application are given in tables in annexes of ISO/IEC 18000-63 and ISO/IEC 18000-64. The presentation should output the real value factored by the measurement span.

The **Monitor-Delay** argument applies only to temperature and humidity simple sensors. It is intended to defer the beginning of the monitoring process for logical process control reasons. For example, a temperature sensor might be applied and configured to a product prior to, or during, a manufacturing or packaging process where temperature is both controlled and different from the post-production environment. The monitor delay is used to ensure that the temperature is only monitored after this controlled period. The monitor delay is a multiplier of the sampling-regime, with a value 0 to 7.

The **High-Out-Of-Range-Alarm-Delay** and the **Low-Out-Of-Range-Alarm-Delay** argument used to exclude (if applicable to the item being monitored) random sample values that would otherwise trigger the alarm. For example, if one or two instances of an out-of-limit reading are not considered detrimental to the item being monitored and sufficient to trigger the alarm, these delay arguments should be used. They are applied independently and are set as an integer value with a value 0 to 7. In some circumstances (e.g. for changes in temperature that affect molecular structure), a low alarm delay of 0 might be applicable because one instance of crossing this threshold might adversely have an impact on the item being monitored, whereas a high alarm delay of 4 or more might be acceptable. This document provides no advice on the relevant values of the delays to be applied for both the high values and the low values.

Write-Sample-And-Configuration-Record command

Module-OID: OBJECT IDENTIFIER = 1 0 24753 0 126 5

Singulation-Id: BYTE STRING (0..255)

Password [OID ref: 1 0 24753 0 126 5 1]: [CONDITIONAL] BYTE STRING (4, or 8, or 16)

Sampling-Regime [OID 1 0 24753 0 5 2 {type} 1]: [CONDITIONAL] TEXT STRING

This argument does not apply to impact and tilt sensors, and the ISO/IEC 24753 sensor processing creates null values for this encoding.

High-In-Range-Limit [OID 1 0 24753 0 5 2 {type} 2] TEXT STRING

This argument does not apply to impact and tilt sensors, and the ISO/IEC 24753 sensor processing creates null values for this encoding.

Low-In-Range-Limit [OID 1 0 24753 0 5 2 {type} 3] TEXT STRING

This argument does not apply to impact and tilt sensors, and the ISO/IEC 24753 sensor processing creates null values for this encoding.

Monitor-Delay [OID 1 0 24753 0 5 2 {type} 4] INTEGER (0..7)

High-Out-Of-Range-Alarm-Delay [OID 1 0 24753 0 5 2 {type} 5] INTEGER

Low-Out-Of-Range-Alarm-Delay [OID 1 0 24753 0 5 2 {type} 6] INTEGER

The **Object-Identifiers** used in the command are the same as those used in ISO/IEC 24753 for processing the command. However, ISO/IEC 24753 also uses a common process **Object-Identifier**, based on the record structure for encoding the data. The final two arcs of a process **Object-Identifier** identify the record number and field number. Although not precise enough for command processing, some implementations of ISO/IEC 24753 might return the process **Object-Identifiers**. The relationship between the two OID structures is defined in Table 1.

Table 1 — Cross-reference between Command OID and Process OID

Function	Command OID	Process OID
Sampling-Regime	1 0 24753 0 5 2 {type} 1	1 0 24753 0 1 4
High-In-Range-Limit	1 0 24753 0 5 2 {type} 2	1 0 24753 0 1 5
Low-In-Range-Limit	1 0 24753 0 5 2 {type} 3	1 0 24753 0 1 6

Table 1 *(continued)*

Function	Command OID	Process OID
Monitor-Delay	1 0 24753 0 5 2 {type} 4	1 0 24753 0 1 7
High-Out-Of-Range-Alarm-Delay	1 0 24753 0 5 2 {type} 5	1 0 24753 0 1 8
Low-Out-Of-Range-Alarm-Delay	1 0 24753 0 5 2 {type} 6	1 0 24753 0 1 9

NOTE Similar mappings between the command OID and process OID apply to other commands

6.3.1.2 Write-Sample-And-Configuration-Record response

The **Write-Sample-And-Configuration-Record** response indicates whether the configuration process was successful or not.

Write-Sample-And-Configuration-Record response

Module-OID: OBJECT IDENTIFIER = 1 0 24753 0 127 5

Response-Code [OID ref: 1 0 24753 0 126 5 1]: INTEGER (0..3) or TEXT STRING

Possible Values:

Value	Definition
0	success
1	sensor not identified (e.g. password mismatch)
2	erase of previous event and/or time synchronisation not applied
3	parameter(s) are not logical

6.3.2 Read-Simple-Sensor-Data-Block

6.3.2.1 Read-Simple-Sensor-Data-Block command

This command is used to read and interpret the bit string from the simple sensor data block. It can also be used for the air interface protocols where the simple sensor data block is returned automatically as an appended component to reading a unique item identifier.

Read-Simple-Sensor-Data-Block command

Module-OID: OBJECT IDENTIFIER = 1 0 24753 0 126 1

Singulation-Id: [CONDITIONAL] BYTE STRING (0..255)

This is only required if the simple sensor data block cannot be appended to the unique item identifier.

6.3.2.2 Read-Simple-Sensor-Data-Block response

The **Sensor-Type** argument identifies the type of simple sensor, with the current supported set defined in 6.3.1.1.

The **Measurement-Span** argument defines between one to eight measurement spans, depending on the type of simple sensor. The **Measurement-Span** included in the response for a particular sensor identifies the range over which the sensor has been manufactured to provide results and the span over which it can be configured.

EXAMPLE 1 Assume a simple sensor of Type 0 for temperature sensors with a span of 14 °C. The only temperature span support near the freezing temperature of water is that which measures in the range or span of −1 °C ± 7 °C.

The **Accuracy** argument defines between one to four levels of accuracy, depending on the type of simple sensor. The **Accuracy** included in the response for a particular sensor identifies the precision to which the sensor has been manufactured. The accuracy level is determined by the sensor manufacturing process and cannot be changed by the user.

EXAMPLE 2 Assume a simple sensor of Type 0 for temperature sensors with a span of 14 °C with a measurement span of −1 °C ± 7 °C. The highest accuracy possible is ±0,5 °C, so a reading of +2 °C, can be for a real temperature between +1,5 °C and +2,5 °C. The lowest accuracy possible is ±2 °C, so a reading of +2 °C, can be for a real temperature between 0 °C and +4 °C.

The **Sampling-Regime, High-In-Range-Limit, Low-In-Range-Limit, Monitor-Delay, High-Out-Of-Range-Alarm-Delay and Low-Out-Of-Range-Alarm-Delay** arguments are all as defined in 6.3.1.1.

The **Alarms** argument is the text representation of four independent alarm states declared by a 1-bit set in a given position of 4-bit field as defined in a table for alarms in annexes of ISO/IEC 18000-63 and ISO/IEC 18000-64. The presentation to the application should show the text string(s) shown in the table. If more than one alarm has been triggered, the text strings should be separated by a semi-colon.

EXAMPLE 3 The bit code pattern 10101 equates to the text string: “Low battery; Delayed high out-of-range.”

Read-Simple-Sensor-Data-Block response

Module-OID: OBJECT IDENTIFIER = 1 0 24753 0 127 1

Sensor-Type [OID 1 0 24753 0 1 1] INTEGER (0..15) or TEXT STRING

The currently supported set of sensor type is defined in 6.3.1.1

Measurement-Span [OID 1 0 24753 0 1 2] TEXT STRING

The measurement span differs with each type of simple sensor, and the mapping between the bit-based codes and the presentation to the application is given in tables in an annex of ISO/IEC 18000-63 or of ISO/IEC 18000-64.

Accuracy [OID 1 0 24753 0 1 3]. TEXT STRING

The accuracy differs with each type of simple sensor, and the mapping between the bit-based codes and the presentation to the application is given in tables in an annex of ISO/IEC 18000-63 or of ISO/IEC 18000-64.

Sampling-Regime [OID 1 0 24753 0 1 4]: [CONDITIONAL] TEXT STRING

This argument does not apply to impact and tilt sensors, and the ISO/IEC 24753 sensor processing creates null values for this encoding.

High-In-Range-Limit [OID 1 0 24753 0 1 5] TEXT STRING

This argument does not apply to impact and tilt sensors, and the ISO/IEC 24753 sensor processing creates null values for this encoding.

Low-In-Range-Limit [OID 1 0 24753 0 1 6] TEXT STRING

This argument does not apply to impact and tilt sensors, and the ISO/IEC 24753 sensor processing creates null values for this encoding.

Monitor-Delay [OID 1 0 24753 0 1 7] INTEGER

High-Out-Of-Range-Alarm-Delay [OID 1 0 24753 0 1 8] INTEGER

Low-Out-Of-Range-Alarm-Delay [OID 1 0 24753 0 1 9] INTEGER

Alarms [OID 1 0 24753 0 1 10] TEXT STRING(S)

Response-Code [OID ref: 1 0 24753 0 127 1 1] INTEGER (0..1) or TEXT STRING
Possible Values:
Value Definition
0 no error
1 error, then this is the only element in the response

6.3.3 Other simple sensor commands

Table 2 defines the list of simple sensor commands. The detailed structure of the **Read-Simple-Sensor-Data-Block** and **Write-Sample- And-Configuration-Record** commands has already been defined in this document. Other ported simple sensor commands can be created using the same construction principles. The OBJECT IDENTFIERS for the commands use the penultimate arc {126}, while the responses use the penultimate arc {127}. The final arc value is the same for a command and its associated response.

Table 2 — Simple Sensor Commands and their OBJECT IDENTIFIERS

Command name	**OBJECT IDENTIFIER**
Read-Simple-Sensor-Data-Block	1 0 24753 0 126 1
Read-Manufacturer-Record	1 0 24753 0 126 2
Write-Password	1 0 24753 0 126 3
Read-Calibration-Record	1 0 24753 0 126 4
Write-Sample-And-Configuration-Record	1 0 24753 0 126 5
Initialise-Sensor-Monitoring	1 0 24753 0 126 6
Read-Sample-And-Configuration-Record	1 0 24753 0 126 7
Read-Event-Record	1 0 24753 0 126 8
Write-UTC-Timestamp	1 0 24753 0 126 9
Read-Time-Synchronisation-Record	1 0 24753 0 126 10
Erase-Monitored-Data	1 0 24753 0 126 11
Activate-Simple-Sensor	1 0 24753 0 126 12
Deactivate-Simple-Sensor	1 0 24753 0 126 13

7 Full function sensors

7.1 General

The encoding for configuring and reading full function sensors is specified in the ISO/IEC 18000 series of standards that support such sensors. The current air interface protocols that support full function sensors achieve this as follows:

— For ISO/IEC 18000-63 (Type C), a *HandleSensor* command is used to embed the ISO/IEC IEEE 21451-7 sensor command and address a physical sensor.

— ISO/IEC IEEE 21451-7 makes no distinction between a ported and a logical sensor, so the processes defined in this clause may be applied to other air interface protocols as they are developed to support sensors.

7.2 Write-Sample-And-Configuration

7.2.1 Write-Sample-And-Configuration command

The **Write-Sample-And-Configuration** command is used to write user-controlled parameters to a full function sensor, either for the initial mission, or to reconfigure on a subsequent mission. The command cannot be invoked for reconfiguration if any of the alarm bits has been set.

There are various addressing mechanisms specified in ISO/IEC IEEE 21451-7, all of which are supported by this command. Before this command can be invoked it is necessary to establish the **Sensor-Address-Type** and the associated **Sensor-Comms-Id**. This can be achieved by invoking the **Read-Sensor-Identifier** command.

Although code values are required for input, ISO/IEC 24753 recommends that if the **Sensor-Type** field is returned as part of the **Sensor-Comms-Id** argument, then additionally the description of the sensor is provided in the response to the **Read-Sensor-Identifier** command.

The values that are returned are then used in their associated arguments to address the specific sensor and ensure communication with the specific sensor. This is important if the RFID tag supports more than one full function sensor, and critically important if there is more than one sensor of the same type.

EXAMPLE 1 The **Read-Sensor-Identifier** response identifies that the **Sensor-Address-Type** has the value 2 encoded as {10} and therefore, a 15-bit string comprising {3-bit TEDS-Type, 7-bit Sensor-Type, and 5-bit Units-Extension} is used for the **Sensor-Comms-Id**. Currently, ISO/IEC IEEE 21451-7, and therefore ISO/IEC 24753, only supports TEDS-Type = {001}. For this example, assume that the 7-bit Sensor-Type = {0010111}, whose value is the binary equivalent derived from an annex in ISO/IEC IEEE 21451-7. The **Read-Sensor-Identifier** response from ISO/IEC 24753 and the response from this document would have indicated that this is a “Temperature, Celsius” sensor. As such it has no assigned extension value, so the 5-bit Units-Extension = {00000}.

NOTE 1 The manufacturer of the sensor determines the structure of the **Sensor-Address-Type** and **Sensor-Comms-Id.**

The **UTC-Timestamp-At-Configuration** argument is expected to deliver the UTC timestamp in a human readable manner (e.g. 2008-08-08 08:08:08). Entering this effectively as a text string is likely to result in the actual time of configuration being later than as declared by the timestamp. If creating the human-readable UTC timestamp, converting it to a 32-bit code and the sensor processor encoding this in a sensor command creates a latency problem, then an application may choose to apply a system solution closer to the sensor command being transmitted across the RFID air interface protocol. Although this document specifies no rules for this, the following three points are worth considering for such a system solution.

— The timestamp on the sensor is a 32-bit value, counting in increments of 1 s.

— The timestamp shall be based on the UTC time epoch beginning at 1970-01-01 00:00:00.

— ISO/IEC 18000-63 supports commands to deliver this 32-bit timestamp to the tag memory, so the same timestamp could be used in the relevant *HandleSensor* command.

NOTE 2 ISO/IEC 24753 also advises that different program languages use different epochs, and it essential that the epoch for this document begins on January 1, 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.

The **Sample-Interval** and **Monitor-Delay** arguments may be set independently of one another to measure in seconds or minutes. In each case, the upper limit is either 9 h 46 min and 7 s, or 22 d 18 h and 7 min. For continuous sampling, the **Sample-Interval** argument is set to the OID for seconds and the INTEGER value 00000 is used. The **Monitor-Delay** argument is used to defer the beginning of the monitoring process for logical process control reasons. For example, a temperature sensor might be applied and configured to a product prior to, or during, a manufacturing or packaging process where temperature is both controlled and different from the post-production environment. The monitor delay is used to ensure that the temperature is only monitored after this controlled period. It is also possible to have a zero value for the **Monitor-Delay** argument, in which case, the OID for seconds is used and the INTEGER value 00000 is used.

NOTE 3 Although a zero value in minutes is mathematically equivalent, the unit of measure is required to be in seconds to align with ISO/IEC IEEE 21451-7 rules.

The **Alarm-Values-Set** argument is used to set whether observed readings outside the lower and/or upper thresholds can trigger an alarm. It is also possible to set neither threshold, which will result in all observations being recorded.

The **Memory-Rollover-Enabled** argument is conditional and depends on whether the sensor supports this feature. This is declared in the Sensor Characteristics TEDS (Type 1) record via the **Memory-Rollover-Capability** field. If memory rollover is supported, the user has the choice of whether to invoke it using this argument.

With **Memory-Rollover-Enabled** switched off, samples are recorded until the associated memory becomes full, at which point a Memory Full alarm is triggered and no further data are recorded. In contrast, with memory rollover switched on, when the first instance of the memory becoming "full" is reached, the sensor will overwrite the earliest recorded sample. This process continues on a FIFO basis. When **Memory-Rollover-Enabled** is switched on, a memory-full alarm is triggered only when the sample time mechanism reaches its capacity.

NOTE 4 **Memory-Rollover-Enabled** refers to a user selectable argument, which can only be invoked if the sensor manufacturer provides a memory rollover facility and logic on the sensor. This is declared by the **Memory-Rollover-Capability** field in the Primary sensor characteristics TEDS (Type 1) record and returned as argument in certain responses. ISO/IEC 24753 is not always clear on the distinction between the two terms.

The **Upper-Alarm-Threshold** and **Lower-Alarm-Threshold** arguments determine the limits at which alarms and/or observed data are recorded depending in the capability of the sensor. These threshold values are stored on the sensor memory as binary values to enable simple comparators to be used with the observed binary values, usually obtained directly from the analog-to-digital converter (ADC). This document requires the threshold values to be input as decimal integers. However, this still requires some understanding of the procedures. ISO/IEC IEEE 21451-7 uses scale factors to define the span over which the sensor is capable of providing meaningful readings. This is so that the processes can be generic independent of the type of sensor. From an application perspective, the procedure might appear to be complex, so two methods are provided which produce similar results.

The first method uses the scale factors as defined in ISO/IEC IEEE 21451-7. Each of the following four fields is in the **Primary Sensor Characteristics TEDS (Type 1)** record:

— The **Scale-Offset-Significand** field is an 11-bit value, when combined with the **Scale-Offset-Exponent** field of 6-bits represents the lowest real value at which the sensor is designed to function. The "offset" is the difference between this real value and zero.

— The **Scale-Factor-Significand** field is an 11-bit value, when combined with the **Scale-Factor Exponent** field represents the range from low to high at which the sensor is designed to function.

EXAMPLE 2

Temperature sensor, range –10 °C to 75 °C, with 12-bit ADC

The real number –10 will be represented by decimal 0 and therefore by binary 0, and the real number 75 will be represented by decimal 4 095 (based on the capability of the 12-bit ADC) and therefore by binary 111111111111. The consequent constraints on the scale factor SF and scale offset SO are:

$-10 = 0 \times SF + SO$

$75 = 4\,095 \times SF + SO.$

Therefore, $SO = -10 = -1 \times 10^1$, and $SF = (75 + 10)/4\,095 = 0{,}0208 = 0{,}208 \times 10^{-1}$, and

$SOS_d = –1{,}000_{10}$, and $SOS_b = 1{,}0000011000_2$

$SOE_d = 1_{10}$, and $SOE_b = 000001_2$

$SFS_d = 0{,}208_{10}$, and $SFS_b = 0001101000_2$

$SFE_d = –1_{10}$, and $SFE_b = 111111_2$.

The sensor processor returns the decimal values via application commands. To set an upper threshold of 28 °C, the real value offset needs to be determined. As the lowest reading possible is –10 °C, the threshold value is calculated as follows:

T = N10 × 0,0208 – 10

28 = N10 × 0,0208 – 10

Simple algebra reveals that N10 = 1826,9 or 1827.

The second method uses details from the manufacturer's specification and the **Data-Resolution** field of the **Primary Sensor Characteristics TEDS (Type 1)** record which determines the number of measurement steps in the sensor. Data resolution is determined by the sensor manufacturer and can be set as a value between 1-bit and 32-bits. In most cases, this will be equivalent to the bit output of the ADC, but this value can also be truncated by the processes used by the sensor manufacturer. As an example, a sensor can have an 11-bit ADC, but the sensor manufacturer might choose to ignore the least significant bit resulting in a 10-bit data resolution that provides 1 023 ({2^10} - 1) measurement steps.

EXAMPLE 3

Temperature sensor, range –10 °C to 75 °C, with 12-bit data resolution and an upper threshold of 28 °C.

The span of this sensor is 85 °C.

The 12-bit data resolution results in 4 095 resolution steps ({2^12} - 1).

Therefore, each 1 °C step = 4 095 / 85.

The threshold of 28 °C has the offset of 38 above the lowest reading.

Therefore, the decimal integer value for the command = 38 × 4 095 / 85 = 1 831.

Write-Sample-And-Configuration command

Module-OID: OBJECT IDENTIFIER = **1 0 24753 7 126 3**

Singulation-Id: BYTE STRING (0..255)

Sensor-Address-Type [OID ref: 1 0 24753 7 126 3 1] INTEGER (0..3)

Possible Values:

Value	Definition
0	no sub-addressing
1	use 7-bit sub-addressing
2	use sensor type sub-addressing
3	use sensor ID sub-addressing

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

UTC-Timestamp-At-Configuration [OID ref: 1 0 24753 7 3 1]

This is delivered in the command in the formal UTC format YYYY-MM-DD HH:MM:SS.

Sample-Interval [OID ref: 1 0 24753 7 3 2 {n}] INTEGER, (0..32767)

The final arc {n} is required to identify the unit of time, with these values:

0 for seconds

1 for minutes

Monitor-Delay [OID ref: 1 0 24753 7 3 3 {n}] INTEGER, (0..32767)

The final arc {n} is required to identify the unit of time, with these values:

0 for seconds

1 for minutes

Alarm-Values-Set [OID ref: 1 0 24753 7 3 4] INTEGER (0..3)

Possible Values:

Value	Definition
0	none
1	lower only
2	upper only
3	both

Memory-Rollover-Enabled [OID ref: 1 0 24753 7 3 5] [CONDITIONAL] BOOLEAN

Possible Values:

Value	Definition
F (0)	switched off
T (1)	switched on

Upper-Alarm-Threshold [OID ref: 1 0 24753 7 3 12] INTEGER

The INTEGER value represents the decimal equivalent of the upper threshold value of an n-bit string, where n is the number of bits for the data resolution. The example (above this module specification) explains how this can be calculated from an application perspective.

Lower-Alarm-Threshold [OID ref: 1 0 24753 7 3 13] INTEGER

The INTEGER value represents the decimal equivalent of the upper threshold value of an n-bit string, where n is the number of bits for the data resolution. The example (above this module specification) explains how this can be calculated from an application perspective.

7.2.2 Write-Sample-And-Configuration response

The **Response-Code** argument identifies whether the command succeeded or failed, and in case, it provides an indication of reason for the failure. Some of the response codes specified in ISO/IEC IEEE 21451-7 are based on functions defined in that standard that are not supported by the ISO/IEC 18000 series of standards.

The **Battery-Status-Code** argument identifies whether the battery power is acceptable or low, against a criterion set by the sensor manufacturer.

Write-Sample-And-Configuration response	
Module-OID: OBJECT IDENTIFIER = 1 0 24753 7 127 3	
Response-Code [OID ref: 1 0 24753 7 127 3 1]: INTEGER (0..7)	
Possible Values:	
Value	Definition
0	Sensor not properly addressed, reply truncated following last bit of response code
1	Command not recognized, reply truncated following last bit of response code
2	Unspecified failure, reply truncated following last bit of battery code
6	Failure due to command details not being supported by the particular sensor, reply truncated following last bit of battery code
7	Success
Code values 3, 4, and 5 are not applicable as they either address sensor security errors or the sensor processing air interface errors. None of these functions are supported by the current version of ISO/IEC 24753 or the air interface protocols that support sensors.	
Battery-Status-Code [OID ref: 1 0 24753 7 127 3 2]: INTEGER (0..1)	
Possible Values:	
Value	Definition
0	Battery OK
1	Battery low

7.3 Read-Alarm-Status

7.3.1 Read-Alarm-Status command

The **Read-Alarm-Status** 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. The ISO/IEC IEEE 21451-7 sensor processes are responsible for collating the relevant information from the different parts of sensor memory.

Sensor addressing using the **Sensor-Address-Type** and **Sensor-Comms-Id** arguments is as defined in 7.2.1.

Read-Alarm-Status command	
Module-OID: OBJECT IDENTIFIER = **1 0 24753 7 126 5**	
Singulation-Id: BYTE STRING (0..255)	
Sensor-Address-Type [OID ref: 1 0 24753 7 126 5 1] INTEGER (0..3)	
Possible Values:	
Value	Definition
0	no sub-addressing
1	use 7-bit sub-addressing
2	use sensor type sub-addressing
3	use sensor ID sub-addressing

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

7.3.2 Read-Alarm-Status response

The **Alarm-Values-Set** argument returns the codes for the settings when the sensor was last configured.

The **Alarm-Triggered** argument returns whether no alarm has been triggered, and if this is not the case, then any of four alarms has been triggered. The values are based on the interpretation by the ISO/IEC 24753 sensor processor based on a 4-bit code present in the Event Administration record of the ISO/IEC IEEE 21451-7 sensor. Although particular text strings are provided in the response module, the text in the response is determined by the design of the sensor processor and may vary by language and by application.

The **Sensor-Map** argument is a direct abstract of a 16-bit field in the ISO/IEC IEEE 21451-7 **Primary Sensor Characteristics TEDS (Type 1)** record. The **Sensor-Map** identifies up to 16 types of measurement types that the particular sensor supports. Bit positions 0 to 9 denote measurement types for which only one measured value is returned in response to a request for data, (e.g. maximum, or peak, value). The other bit positions denote measurement types for which multiple data values can be returned by the sensor (e.g. data log of observed value with time tick (8-bit code) reporting outside either threshold), or which are currently reserved for future definition. A table in ISO/IEC IEEE 21451-7 specifies the full list of interpretations.

The **Sensor-Map,** when used in conjunction of the ISO/IEC IEEE 21451-7 table specifying all the measurement type codes, enables other commands to be constructed to read data from a particular record.

NOTE The ISO/IEC 24753 sensor processor retains the 16-bit code string to validate whether any subsequent command to read a particular record on the sensor is possible. This is to avoid unnecessary air interface and sensor transactions.

The **Response-Code** argument identifies whether the command succeeded or failed, and in case, it provides an indication of reason for the failure. Some of the response codes specified in ISO/IEC IEEE 21451-7 are based on functions defined in that standard that are not supported by the ISO/IEC 18000 series of standards.

The **Battery-Status-Code** argument identifies whether the battery power is acceptable or low, against a criterion set by the sensor manufacturer.

Read-Alarm-Status response

Module-OID: OBJECT IDENTIFIER = **1 0 24753 7 127 5**

Alarm-Values-Set [OID ref: 1 0 24753 7 3 4]: INTEGER (0..3)

Possible Values:

Value	Definition
0	none
1	lower only
2	upper only
3	both

Alarm-Triggered [OID ref: 1 0 24753 7 4 6]: TEXT STRING

If no alarms have been triggered then the text string is “NO-ALARMS”. However, there are up to four alarm messages based in the settings of a bit string on the sensor memory. These are:

“UPPER-ALARM”

“LOWER-ALARM”

“MEMORY-FULL”

“LOW-BATTERY”

Sensor-Map [OID ref: 1 0 24753 7 2 4]: 16-bit code

The interpretation of this code requires access to the table of measurement type codes in ISO/IEC IEEE 21451-7.

Response-Code [OID ref: 1 0 24753 7 127 5 1]: INTEGER (0..7)

Possible Values:

Value	Definition
0	Sensor not properly addressed, reply truncated following last bit of response code
1	Command not recognized, reply truncated following last bit of response code
2	Unspecified failure, reply truncated following last bit of battery code
6	Failure due to command details not being supported by the particular sensor, reply truncated following last bit of battery code
7	Success

Code values 3, 4, and 5 are not applicable as they either address sensor security errors or the sensor processing air interface errors. None of these functions are supported by the current version of ISO/IEC 24753 or the air interface protocols that support sensors.

Battery-Status-Code [OID ref: 1 0 24753 7 127 5 2]: INTEGER (0..1)

Possible Values:

Value	Definition
0	Battery OK
1	Battery low

7.4 Read-Event-Record-Segments

7.4.1 Read-Event-Record-Segments command

The **Read-Event-Record-Segments** command reads all the event-based records placed in the appropriate sensor memory segment(s) by the sensor.

To process the responses to this command, the **Read-Event-Administration-Record** command should have been invoked immediately beforehand. Some of the **Read-Event-Administration-Record** response argument values are essential to the correct reconstruction of the event history.

Sensor addressing using the **Sensor-Address-Type** and **Sensor-Comms-Id** arguments is as defined in 7.2.1.

A specific code value for the **Measurement-Type** argument needs to be set in the command. This and the data resolution value determine the size of the segment in bits.

EXAMPLE

Assume a sensor with 12-bit data resolution and using an 8-bit time tick. This sensor should be capable of storing 256 event records (known as “sensor words” in ISO/IEC IEEE 21451-7. Each sensor word is 20-bits long (= 8-bits for the time tick + 12-bits for the recorded data). There are 32 sensor words per segment, so reading one segment from this sensor requires 640 bits of data, plus other bits required in the responses and command. Reading multiple records increases the number of bits significantly, as can requiring to support a greater number of bits for data resolution.

The 16-bit time tick combined with the highest resolution produces a sensor word of 48 bits (= 16-bits for the time tick + 32-bits for the recorded data), resulting in a single sensor segment of the largest size: 960 bits.

If the reading environment cannot support the transfer of large bit strings in the responses, then the **Read-Partial-Event-Record-Segment** command should be used.

Although the command can support the reading of segments up to the maximum support by the ISO/IEC IEEE 21451-7 command, the first constraint is the maximum number of segments that the sensor manufacturer defines is possible from the memory allocated to the **Measurement-Type**. The actual maximum number of segments that the sensor can support is given by the Code n sample capacity field of the **Event Administration Record**.

There are three additional constraints on the construction of the arguments associated with the number of segments.

— Some implementations assume that the sensor has the capability of supporting the delivery of a repeated number of segments asynchronously when the value of the **Number-Of-Segments** is less than the difference between the first and last segment in the command.

— In turn, this asynchronous packet delivery of multiple responses to the one air interface command has to be supported by the air interface protocol.

— The size of the packet, even though supported by a CRC-16 on each segment, might still result in some radio interference across the air interface.

This document provides no rules on apply constraints to these factors. They, therefore, have to be applied by setting values for the arguments, particularly the **Number-Of-Segments** when addressing asynchronous packet delivery, based on the capability of the sensor and the air interface protocol.

The **First-Segment-Number** argument identifies the initial segment to be read and returned in the response. Its value shall not be greater than the number of segments supported by the sensor.

Number-Of-Segments argument determines the number of segments that the sensor is expected to deliver in a response to the command. The number of segments shall be no greater than (Last Segment Number) − (First Segment Number) + 1. The number may be smaller if the sensor and the air interface protocol support asynchronous packet delivery. In this case, multiple responses are returned for the one command.

ISO/IEC IEEE 21451-7 states that the value of the total number of segments may have one of two logical values as follows.

a) If the number covers the span from first to last segment in the command, then the sensor shall respond with the entire packet of all the required segments.

b) If the number calls for fewer segments, the intention is for the sensor to respond with that number of segments encapsulated in an RFID response, then to respond immediately with another similar sized packet and to continue until all the segments that were requested have been transmitted. This command and response asynchronous cycle shall only be possible under the following conditions.

   1) The sensor shall have the facility to support this type of response. If it does not, then it shall ignore the number of segments and deliver all the requested segments as a data packet.

   2) The value of the Number of Segments parameter in the command shall be an integer fraction of the total number of segments requested in the command. If this is not the case, then the sensor shall ignore the number of segments and deliver all the requested segments as a data packet.

NOTE ISO/IEC IEEE 21451-7 only addresses the conditions that apply to the sensor and take no account of limitations of this procedure based on the air interface protocol.

The **Last-Segment-Number** argument identifies the final segment to be read and returned in the response. Its value shall not be greater than the number of segments supported by the sensor.

Read-Event-Record-Segments command

Module-OID: OBJECT IDENTIFIER = **1 0 24753 7 126 8**

Singulation-Id: BYTE STRING (0..255)

Sensor-Address-Type [OID ref: 1 0 24753 7 126 8 1] INTEGER (0..3)

Possible Values:

Value	Definition
0	no sub-addressing
1	use 7-bit sub-addressing
2	use sensor type sub-addressing
3	use sensor ID sub-addressing

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

Measurement-Type [OID ref: 1 0 24753 126 8 3] INTEGER (10..13)

Possible Values:

Value	Definition
10	Data log of observed value recorded at each sample interval
11	Data log of observed value with time tick (8-bit code) reporting outside either threshold
12	Data log of observed value with time tick (16-bit code) reporting outside either threshold
13	Data log of all observed values after an initial alarm value has triggered

First-Segment-Number [OID ref: 1 0 24753 126 8 4] INTEGER (0..2047)

If the **Measurement-Type** code = 11 (for the 8-bit time tick record) the highest value is 255. The highest value for the other three types is 2047.

Number-Of-Segments [OID ref: 1 0 24753 126 8 5] INTEGER (0..63)

This value is equal to or less than (Last Segment Number) – (First Segment Number) + 1. If less than this value, then it implies that some form of packet control is expected to be supported by the sensor.

Last-Segment-Number [OID ref: 1 0 24753 126 8 6] INTEGER (0..2047)

If the **Measurement-Type** code = 11 (for the 8-bit time tick record) the highest value is 255. The highest value for the other three types is 2047.

7.4.2 Read-Event-Record-Segments response

The **Read-Event-Record-Segments** response contains a number of conditional arguments that depend on the value of the **Measurement-Type** set in the command. Each individual event record is identified by the OID for the type of record qualified by two trailing arcs: {segment} {sensor-word}. 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 2 047, but with the maximum number of segments for the particular sensor declared by the sample capacity for the in the Event Administration record.

The sensor processor, which is the implementation of ISO/IEC 24753, does not address the reconstruction of the event records along a timeline. However, ISO/IEC 24753 does provide some