INTERNATIONAL STANDARD

ISO/IEC 29173-1

First edition
2012-12-01

Information technology — Mobile item identification and management —

Part 1: Mobile RFID interrogator device protocol for ISO/IEC 18000-63 Type C

Technologies de l'information — Gestion et identification d'élément mobile —

Partie 1: Protocole de dispositif interrogateur RFID mobile pour l'ISO/CEI 18000-63 Type C

Reference number
ISO/IEC 29173-1:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

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 29173-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

ISO/IEC 29173 consists of the following parts under the general title *Information technology — Mobile item identification and management*:

— *Part 1: Mobile RFID interrogator device protocol for ISO/IEC 18000-63 Type C*

Introduction

This International Standard provides a mobile RFID interrogator device protocol defining the interface between a mobile AIDC application platform and a mobile RFID interrogator.

Without this standard protocol, manufacturers of mobile RFID interrogators would likely develop multiple mobile RFID interrogator device drivers for different mobile AIDC application platforms. Moreover, they also would likely develop a proprietary interface protocol between the mobile AIDC application platform and the mobile RFID interrogator. On the other hand, if both the mobile AIDC application platform and the Mobile RFID interrogator developers follow this standardized mobile RFID interrogator device protocol, various independent products will be mutually interoperable without extra costs.

Previously, the ISO and EPC standards defined an interrogator protocol that supports connection of an RFID interrogator to a host through a network. Those standards are applied mainly in cases in which an RFID interrogator is connected through a network.

In mobile RFID systems, an RFID interrogator is mounted inside a mobile phone or attached to a mobile phone in a dongle configuration. Such RFID systems require a protocol that enables a main control unit of the mobile phone to control the RFID interrogator. Therefore, the Mobile RFID system requires a new interface protocol between the mobile AIDC application platform and the mobile RFID interrogator to meet the new requirements.

There are alternative techniques to meet the use case addressed by this International Standard, and one is the use of EPC and ONS. Those interested in this technique are encouraged to contact GS1 for further information.

Information technology — Mobile item identification and management —

Part 1: Mobile RFID interrogator device protocol for ISO/IEC 18000-63 Type C

1 Scope

This part of ISO/IEC 29173 defines an interface protocol between a device driver of a mobile AIDC application platform and a mobile RFID interrogator within a mobile AIDC terminal. It is intended that this part of ISO/IEC 29173 be implemented in both the device driver of a mobile AIDC application platform and the mobile RFID interrogator.

In accordance with the ISO/IEC 18000-63 type C RFID air interface, this part of ISO/IEC 29173 includes:

— types of command/response/notification protocol messages and their usages,

— protocol message format, and

— protocol message exchange procedures.

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 18000-63, *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 19762 (all parts), *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary*

ISO/IEC 29143, *Information technology — Mobile item identification and management — Air interface specification for Mobile RFID interrogators*

ISO/IEC TR 29172, *Information technology — Mobile item identification and management — Reference architecture for Mobile AIDC services*

3 Terms, definitions, symbols and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762, ISO/IEC 18000-63, ISO/IEC TR 29172 and the following apply.

3.1.1
mobile AIDC application platform
set of computer instructions as the place to launch a mobile AIDC application in a mobile AIDC terminal

[ISO/IEC TR 29172]

3.1.2
mobile AIDC terminal
electronic device equipped with one or more mobile AIDC device(s) to support the functions of Mobile Item Identification and Management (MIIM) technologies

[ISO/IEC TR 29172]

3.1.3
mobile RFID
AIDC technique supporting the Mobile Item Identification and Management (MIIM) technologies of radio frequency identification (RFID)

[ISO/IEC TR 29172]

3.1.4
mobile RFID interrogator
electronic equipment that retrieves information from radio frequency tags by transmitting and receiving radio frequency signals to and from the tags

3.1.5
mobile RFID MIIM content name
description of the object associated with MIIM services, such as brand name, company name, food name, movie title, building name, etc.

3.2 Abbreviated terms

AFID	Application Family ID
A/I	Air Interface
AM	tag memory Access Mode
AP	Access Password
Arg	Argument
BM	Byte Mask
CRC	Cyclic Redundancy Check
DR	TRcal Divide Ratio
EAC	Embedded Application Code
EnMAC	Enable MAC
H/W	Hardware
KP	Kill Password
LD	Lock Data
MB	Tag Memory Bank
MB01EC	UII Memory Bank (01) data except the CRC

MB01ECL	MB01EC Length
MD	Memory Data of TID or content name
MDL	Memory Data Lengh
MI	Modulation Index
MII	Mobile Item Identifier
MSA	Memory Start Address
MT	Message Type
MTIME	Maximum elapsed Time to tagging
NUMT	Number of Maximum Tag to read
PL	Payload Length
PNTR	Pointer
RC	Repeat Cycle for inventory
RFU	Reserved for Future Use
RSV	Reserved
SDF	Storage Data Format
SeKey	Security Key
SelFM	Select Frequency Mode
SelSeM	Select Security Mode
S/N	Serial Number
TID	Tag Identifier
TN	Tag Number
TR	TRcal (Tag-to-interrogator Calibration symbol)
UID	Unique Identifier
UII	Unique Item Identifier
URI	Uniform Resource Identifier
WD	Write Data
WDL	Write Data Length
b	bit

3.3 Notation

This document uses the following notational conventions:

0bNNNN	binary notation
0xNNNN	hexadecimal notation

4 Overview

This part of ISO/IEC 29173 is organized as follows: Clause 5 describes the message format of a mobile RFID interrogator device protocol. Clause 6 summarizes and lists all commands, responses and notifications defined in this part of ISO/IEC 29173. Clause 7 describes the details of each command, response, and notification. Clause 8 describes the test mode protocol used for testing a mobile RFID interrogator. The role of this part of ISO/IEC 29173 in mobile RFID is described in ISO/IEC TR 29172.

5 Message format of mobile RFID interrogator device protocol

5.1 Overview

A mobile RFID interrogator device protocol includes a preamble, a header, a payload, and an end mark. Fig. 1 illustrates a message format of a mobile RFID interrogator device protocol.

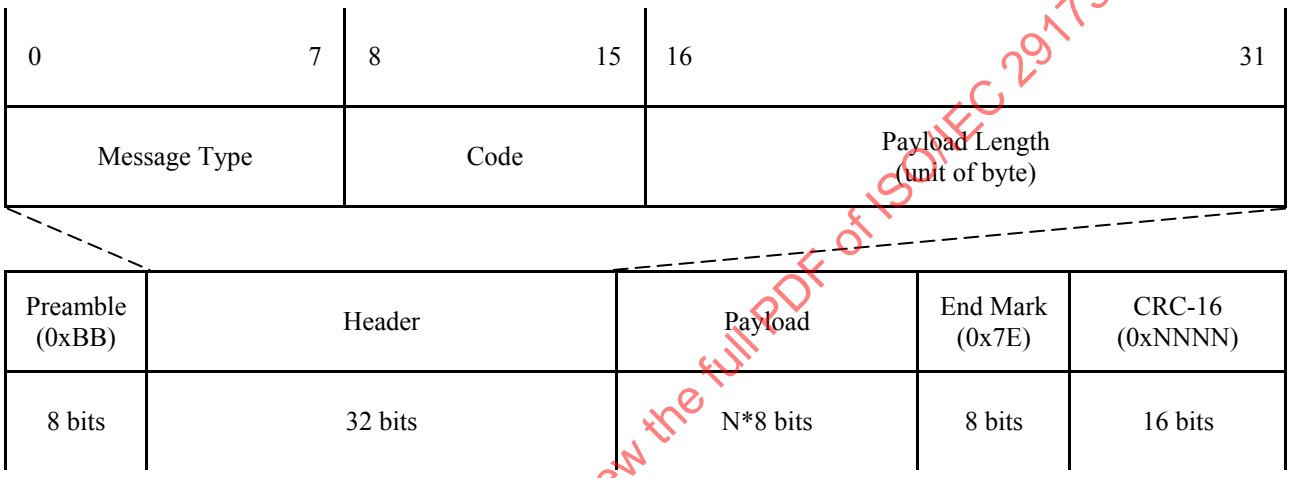


Figure 1 — Message format

5.2 Preamble field and end mark field

A preamble includes information for indicating the start of a protocol message, which is used to discern protocol messages. For example, the preamble may be configured in 8 bits and may have a value of 0xBB. A header includes information for indicating the message type, a corresponding code, and a payload length. That is, the payload length information is stored in the header and the information received from the RFID tag is stored in the payload. An end mark includes information for indicating the end of a protocol message, which is used to discern the protocol messages together with the preamble. For example, the end mark may be configured in 8 bits and may have a value of 0x7E.

5.3 Header field

5.3.1 General

The header includes three fields describing: 1) an RFID tag type, 2) command/response/notification type and code, and 3) the payload length. The message type field is used to discern a command that is transmitted from the processor to the interrogator and a response and notification transmitted from the interrogator to the processor. The code field is used to discern a variety of types of commands, responses or notifications. The code field includes information about the success or failure of commands in response and notification. The payload length field includes information indicating the length, in bytes, of the payload.

5.3.2 Message type field

The message type field associates commands, responses and notifications with the corresponding code value represented in 8 bits. The message type (e.g., command, response and notification) can be discerned using the values shown in Table 1 below.

Table 1 — Command/Response/Notification type and corresponding code value

Command/Response/Notification type	Corresponding code value (Hexadecimal)
Command	0x00
Response	0x01
Notification	0x02
Test mode	0x03
Reserved	0x04-0xFF

As shown in Table 1, a code value indicating a command is 0x00, a code value indicating a response is 0x01, a code value indicating a notification is 0x02, a code value indicating a test mode is 0x03, and code values indicating "Reserved" are 0x04 ~ 0xFF. The command, the response, the notification, and the test mode shown in Table 1 will be described later in detail.

5.3.3 Code field

A code field is used to discern the types of command, response, and notification. There may be various commands to be processed by a mobile RFID interrogator, various responses to the commands and various notifications to be sent by the interrogator. Therefore, when a code is assigned to a command, response or notification, the interrogator can accurately distinguish them by referring to the message type field and the code field. For example, when a value 0x00 and a value 0x01 are respectively assigned to a message type field and a code field for a power control command, the interrogator can recognize the received command as the power control command by the assigned values.

5.3.4 Payload length field

A payload length field indicates the length of the payload field, which occurs immediately after a header field. For example, the payload length field may be composed of 16 bits. Here, the unit of length is byte. When a payload length is represented in byte using 16 bits, the maximum length that can be represented becomes 65,536 bytes. This means that the maximum length of a payload cannot exceed 65,536 bytes.

5.4 Payload field

A payload field stores various types of data. The payload field may include arguments related to a command transmitted from the processor to the RFID interrogator, as well as various data contained in a response transmitted from the RFID interrogator to the processor. There may be various types of payloads suitable for respective commands and responses, such as the payloads illustrated in Figs. 2 and 3 that illustrate payload type A through payload type Q.

Each of the payloads illustrated in Figs. 2 and 3 includes a specific field. The use of the specific field and the method thereof will be described later in detail. The generation and configuration of each payload type will now be described in detail. The number of bits and the order mentioned in the following payload configurations are presented as examples and do not limit this part of ISO/IEC 29173.

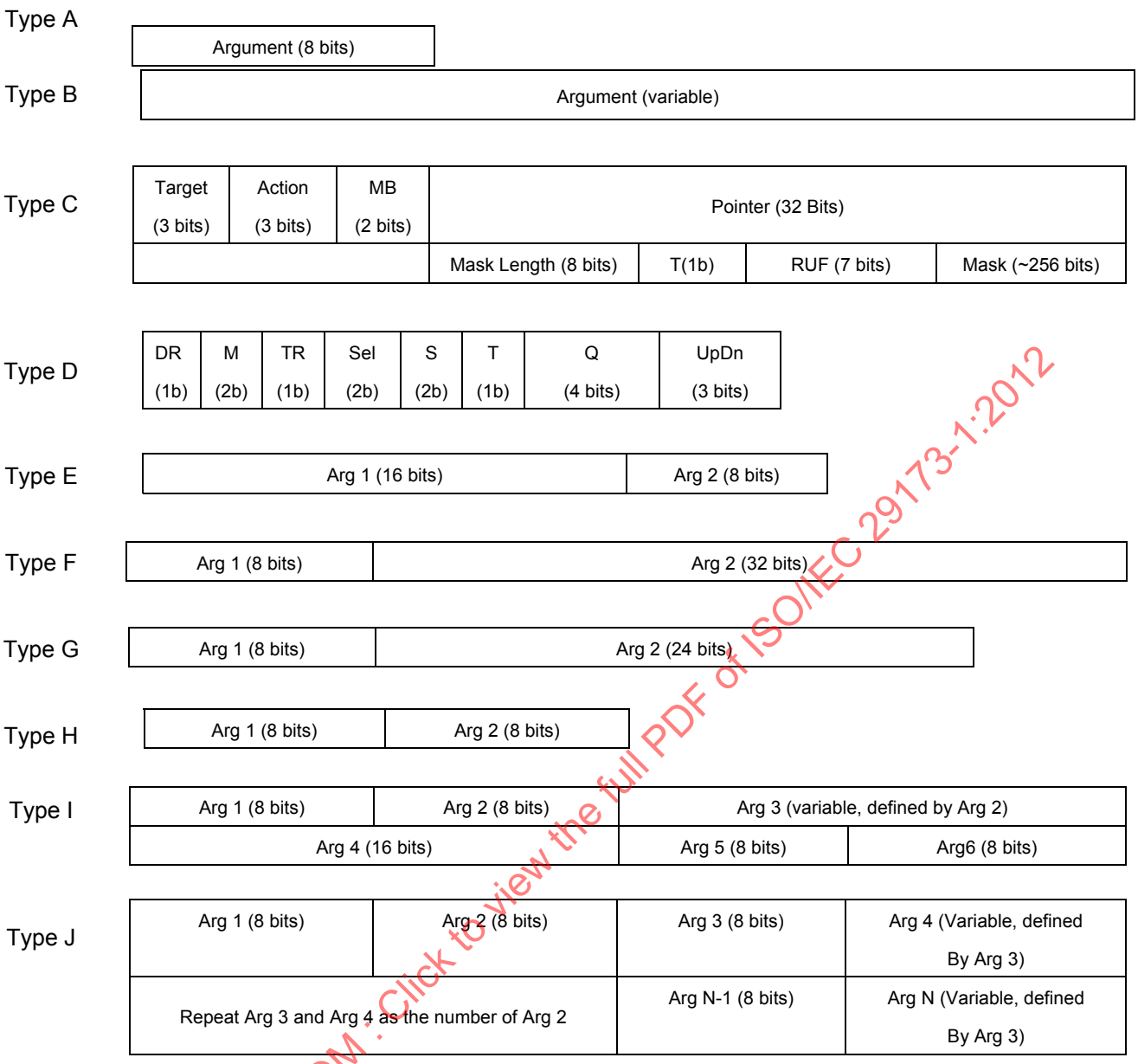


Figure 2 — Payload type A to payload type J

Type K

Arg 1 (8 bits)	Arg 2 (8 bits)	Arg 3 (8 bits)	RSV (5 bits)	DT (2 bits)	AM (1 bit)
Arg 7 (32 bits)					
Arg 8 (16 bits)		Arg 9 (8 bits)			

Type L

Arg 1 (8 bits)	Arg 2 (8 bits)	Arg 3 (8 bits)	Arg 4 (Variable, defined By Arg N-3)	Arg 5 (8 bits)	Arg 6 (Variable, defined By Arg 5)
Repeat from Arg 4 to Arg 6 as the number of Arg 2					
Arg N-3 (8 bits)	Arg N-2 (Variable, defined By Arg N-3)		Arg N-1 (8 bits)	Arg N (Variable, defined By Arg N-1)	

Type M

Arg 1 (32 bits)			
Arg 2 (8 bits)	Arg 3 (8 bits)	Arg 4 (Variable, defined by Arg 3)	
Arg 5 (16 bits)		Arg 6 (8 bits)	Arg 7 (Variable, defined By Arg 6)

Type P

Arg 1 (32 bits)		
Arg 2 (32 bits)		
Arg 3 (8 bits)	Arg 4 (Variable, defined by Arg 3)	Arg 5 (8 bits)

Type N

Arg 1 (32 bits)	
Arg 2 (8 bits)	Arg 3 (Variable, defined by Arg 2)
Arg 4 (8 bits)	Arg 5 (20 bits)

Type Q

Arg 1 (32 bits)		
Arg 2 (8 bits)	Arg 3 (Variable, defined by Arg 2)	
Arg 4 (8 bits)	Arg 5 (16 bits)	Arg 6 (8 bits)

Type O

Arg 1 (32 bits)

Figure 3 — Payload type K to payload type Q

The description of the preceding payload types, the use of each field, and the use method thereof will be described later in detail.

5.5 Endian format and transmission order format

All of the fields used in constructing the mobile RFID interrogator device protocol format follow the big-Endian format. According to the big-Endian format, the most significant byte value is written first followed by the least significant byte value. A preamble field, a header field, a payload field, and an end mark field are transmitted in the order named. In the header field, a message type field, a code field, and a payload length field are transmitted in the order named. In the payload field, a target field, an argument type field, a payload data length field, and a payload data and pending field are transmitted in the order named. In each field, the most significant byte is transmitted first.

5.6 Method of describing small-size data in fixed-size field

When a small-size data needs to be inserted into a protocol field larger than the small-sized data, the less significant bytes are filled first, then the remaining more significant bytes are filled with 0x00. In this case, the big-Endian format is also applied. For example, when a value of 12 needs to be inserted into a 16-bit length field, the less significant bytes are filed with 0x0C and the more significant bytes are filled with 0x00.

5.7 Cyclic Redundancy Check (CRC) field

5.7.1 CRC general

The commands and responses use the same CRC-16 to verify the accuracy of the message bits.

On receiving a command from a mobile AIDC application platform, the RFID interrogator shall verify that the checksum or CRC value is valid. If it is invalid, it shall discard the message and shall not respond and shall not take any other action. Otherwise it shall request the command again by sending the response with code field and result code of 0xFF. And on receiving a response from the RFID interrogator, the mobile AIDC application platform shall verify that the checksum or CRC value is valid. If it is invalid, response shall be discarded and the command shall be sent again within a limited retransmit count.

The 16-bit CRC shall be calculated on all the message bits from the message type field to the end mark field.

The polynomial used to calculate the CRC is $X^{16}+ X^{12}+ X^{5}+1$. The 16-bit register for the CRC calculation in a mobile AIDC application platform or an RFID interrogator shall be preloaded with 0xFFFF. The resulting CRC value shall be inverted, attached to the end of the packet (after End Mark field) and transmitted.

The most significant byte shall be transmitted first. The most significant bit of each byte shall be transmitted first.

NOTE A schematic of a possible implementation is provided in Annex A.

The CRC in receiving part shall be implemented in one of two ways:

5.7.2 Inversion of incoming CRC bits by the receiving part

At the receiving part, the incoming CRC bits are inverted and then clocked into the register. After the LSB CRC bit is clocked into the register, the 16-bit CRC register should contain all zero's.

5.7.3 Non-inversion of incoming CRC bits by the receiving part

If the received CRC bits are not inverted before clocking, after the LSB CRC bit is clocked into the register, the CRC register will have the value 0x1D0F.

6 Summary and list of command, response, and notification

6.1 Overview

The protocol between the processor of a mobile terminal and an RFID interrogator can be classified into a command, a response, and a notification according to ISO/IEC 18000-63.

In this part of ISO/IEC 29173, the command and the response always exist and operate in pairs. Only after a response to a command has been received is the next command executed. Each command has a specific code, which is described in the code field of the header and represented in 8 bits. When a response is successful, the code of a corresponding command is described in the code field and a response-related content is described in the payload field. On the other hand, when a response is unsuccessful, a value of 0xFF is described in the code field and the result code is described in the payload field. The payload varies according to the commands and responses. Detailed types of payloads will be described later in detail.

The commands and responses in a mobile RFID interrogator device protocol are categorized into interrogator control/management, tag read, tag write, tag kill/lock/erase, and additional functions. Table 2 below shows an example of a command list according to this part of ISO/IEC 29173. In Table 2, the commands are classified into mandatory and optional. All of the commands have corresponding responses. Commands corresponding to a tag write category and a tag kill/lock/erase category shall be implemented carefully because these commands may change the contents of a tag. When these commands are used incorrectly, security problems may occur.

Table 2 — Overview of mandatory and optional commands

Type	Mandatory command	Optional command
Interrogator control / management category	Get type C A/I select parameters Set type C A/I select parameters Get type C A/I query-related parameters Set type C A/I query-related parameters Set security key Set frequency mode Set MAC control	Get region Set region Interrogator power control Interrogator connection control Get interrogator information Get signal strength Set signal strength Reset interrogator Get automatic read parameters Set automatic read parameters
Tag read category	Read tag once Read multiple tags Stop multiple tags	
Tag write category		Write tag
Tag kill/lock/erase category	Kill tag Lock tag	Erase tag
Additional function category		Get last result Start test mode Stop test mode Star receive test Stop receive test

6.2 Interrogator control/management category

The RFID interrogator control/management command category includes the commands shown in Table 3 below. The main commands relate to interrogator power control, interrogator connection control, get interrogator information, interrogator RF signal strength control, and interrogator filter function control. The

interrogator power control command and the reset interrogator command, which are the most basic interrogator control commands, may be directly controlled by a hardware interface. In this case, the previous two commands may not be implemented separately.

Table 3 — Commands in interrogator control/management category

Command name	Code value (Hexadecimal)	Description of command
Interrogator power control	0x01	Power on/off interrogator (Active/Sleep)
Interrogator connection control	0x02	Initialize or end communication with interrogator
Get interrogator information	0x03	Get interrogator-related information (Interrogator at least have information about model, manufacturer, S/N, usable frequency, and supportable tag type)
Get signal strength	0x04	Get RF signal strength of interrogator
Set signal strength	0x05	Set RF signal strength of interrogator
Get region	0x06	Get nation/region information of interrogator
Set region	0x07	Set nation/region information of interrogator
Reset interrogator	0x08	Immediately end all operations of interrogator and initialize interrogator. Thereafter, interrogator control requires using interrogator connection control command
Get type C A/I select parameters	0x09	Get select parameters of air interface related to ISO/IEC 18000-63
Set type C A/I select parameters	0x0A	Set select parameters of air interface related to ISO/IEC 18000-63
Get type C A/I query related parameters	0x0B	Get query-related parameters of air interface related to ISO/IEC 18000-63
Set type C A/I query related parameters	0x0C	Set query-related parameters of air interface related to ISO/IEC 18000-63
Get automatic read parameters	0x0D	Get parameters related to automatic mode read
Set automatic read parameters	0x0E	Set parameters related to automatic mode read
Set security key	0x0F	Set the security key for data protection or access password
Set frequency mode	0x10	Set the frequency mode of interrogator
Set MAC control	0x11	Set the MAC parameter for ISO/IEC 29143
Vendor specific	0x12-0x1B	Vendor specific commands
Reserved	0x1C-0x1F	Reserved

Referring to Table 3, the interrogator control/management category includes a get automatic read parameters command and a set automatic read parameters command for getting and setting commands related to an automatic read command. These commands operate according to preset parameters. These preset parameters include a read cycle indicating the number of times of performance of a read operation, and also include a read delay time indicating a delay time between read operations when the interrogator performs a read operation more than two times.

6.3 Tag read category

A tag read category includes the commands shown in Table 4 below. These commands are mainly used to read the specific ID and the user data of tag.

Table 4 — Commands in tag read category

Command name	Code value (Hexadecimal)	Description of command
Read tag once	0x21	Read UII bank or/and TID bank or/and user memory bank of ISO/IEC 18000-63 tag once
Read multiple tags	0x22	Read UII bank or/and TID bank or/and user memory bank of ISO/IEC 18000-63 tag multiple times
Stop read multiple tags	0x23	Stop to read multiple tags operation
Vendor specific	0x24-0x29	Vendor specific commands
Reserved	0x2A-0x2F	Reserved

In Table 4 above, the tag read category further includes commands for automatically reading a plurality of successive tags. The commands related to the read multiple tags operation include a read multiple tags command and a stop read multiple tags command.

The read multiple tags command is configured to be repeated and a repeat cycle, which indicates the number of times a read operation of a read cycle, designated in the set automatic read parameters command, needs to be repeated. That is, the total number of times for a read operation of the RFID interrogator equals Read cycle × Repeat cycle. When a repeat cycle is an invalid value, a response indicating a wrong value is generated in a result code. When a read operation has been performed by the repeat cycle or there is no tag to be read, the RFID interrogator automatically stops the read operation.

The stop read multiple tags command is used to stop the read operation that is being performed by the start automatic read command.

Most commands cannot be executed during the read multiple tags operation. If such a command is to be executed, the command is regarded as failed and the result code receives 0x23 of Read in Operation. Examples of commands executable during the read multiple tags operation are reset interrogator, get signal strength, set signal strength, and stop read multiple tags.

Data obtained from an RFID tag by a tag read command are transmitted to the processor of the terminal through a notification or a response to a command. In this part of ISO/IEC 29173, a buffer function for storing data obtained from a tag by a interrogator chip is set to be optional.

6.4 Tag write category

The tag write category includes the commands shown in Table 5 below. These commands are used to write an ID code, a user memory bank area, and additional information in the tag.

Table 5 — Commands in tag write category

Command name	Code value (Hexadecimal)	Description of command
Write tag	0x31	Write UII bank and/or user memory bank of ISO/IEC 18000-63
Vendor specific	0x32-0x39	Vendor specific commands
Reserved	0x3A-0x3F	Reserved

6.5 Tag kill/lock/erase category

This category includes the commands shown in Table 6 below. The kill tag command is used to kill (erase) the contents of a tag. The lock tag command is used to lock the contents of a tag and the erase tag command is used to erase the contents of a tag.

Table 6 — Commands in tag kill/lock/erase category

Command name	Code value (Hexadecimal)	Description of command
Kill tag	0x41	Kill tag
Lock tag	0x42	Control lock of ISO/IEC 18000-63
Erase tag	0x43	Erase tag
Vendor specific	0x44-0x49	Vendor specific commands
Reserved	0x4A-0x4F	Reserved

6.6 Additional function category

An additional function category includes commands for additional functions, as shown in Table 7 below.

Table 7 — Commands in additional function category

Command name	Code value (Hexadecimal)	Description of command
Get last result	0x51	Interrogator gets the last event
Stat test mode	0x52	Convert mode of interrogator into test mode
Stop test mode	0x53	Stop test mode of interrogator
Stat receive test	0x54	Start count of successfully-received air interface packet to measure receive sensitivity
Stop receive test	0x55	Stop count of successfully-received air interface packet for measurement of receive sensitivity
Vendor specific	0x56-0x59	Vendor specific commands
Reserved	0x5A-0x5F	Reserved

The additional function category includes convenience-providing functions other than the basic contents for processing a tag by the interrogator. These include a filter function of the interrogator and commands for getting and setting the tag access status of the interrogator. The additional function category further includes commands for starting or ending a test mode. A start receive test command and a stop receive test command for measurement of receive sensitivity can be used only in the test mode. The test mode will be described later in detail.

6.7 Result code

A result code is used for the response to a command. The result code indicates the results of commands in the RFID interrogator. If the command is successful, the code value of the corresponding command is inserted into the code field of the header of the response message. If the command failed, a value of 0xFF is inserted into the code field of the header of the response message. An 8-bit result code is also inserted into the payload data section to provide an indicator for commands that are incorrectly executed. A result code of 0x00 indicates success and no separate result value exists. Table 8 below illustrates the types of results and the corresponding codes. If the CRC error occurs on a command received at the interrogator, the interrogator could send a response by the 0xFF of code field and 0xFF of payload field.

Table 8 — Result codes

Result type	Result code	Description
Success	0x00	Indicates success of command
Power control failure	0x01	Power control on/off operation is failed
Connection control failure	0x02	Connection control operation is failed
Cannot get interrogator info	0x03	Interrogator ID cannot be set or got
Cannot get signal strength	0x04	Signal strength cannot be got
Set signal strength failure	0x05	Signal strength cannot be set
Cannot get region	0x06	Region information of interrogator cannot be got
Region control failure	0x07	Region of interrogator cannot be set
Reset interrogator failure	0x08	Interrogator cannot be reset
Cannot control type C A/I parameters	0x09	Air interface parameters related to ISO/IEC 18000-63 cannot be set or got
Cannot get automatic read parameters	0x0D	Read multiple tags command parameters cannot be got
Set automatic read parameters failure	0x0E	Read multiple tags command parameters cannot be set
Security key setting failure	0x0F	Protection mode setting is failed
Frequency mode setting failure	0x10	Frequency mode setting is failed
MAC control setting failure	0x11	MAC control setting is failed
Reserved	0x12~0x1F	
Read failure	0x21	Read operation is failed
Read multiple tags failure	0x22	Read multiple tags operation is failed
Read in operation	0x23	Automatic read is already in operation
Read complete	0x24	Read multiple tags operation is completed
Cannot stop read multiple tags	0x25	Read operation cannot be stopped
Not in read operation	0x26	Read is not in operation
Reserved	0x27~0x2F	
Write tag failure	0x31	Write operation is failed
Reserved	0x32~0x3F	
Kill tag failure	0x41	Kill operation is failed
Lock control failure	0x42	Lock operation is failed
Erase tag failure	0x43	Erase operation is failed
Reserved	0x44~0x4F	
Cannot get last result	0x51	Last event cannot be got
Test mode control failure	0x52	Test mode control is failed
Interrogator is not in test mode	0x53	Test mode control is failed
Reserved	0x54~0x5F	
Invalid parameter	0x61	Parameter is invalid
Not supported command	0x62	Command is not supported
Undefined command	0x63	Command is not defined
Password not match	0x64	Password does not match
No tag detected	0x65	No tag is detected
No user memory bank	0x66	User memory bank does not exist
No more tags to read	0x67	No more tags remain to be read
Reserved	0x68~0x6F	
Vendor specific	0x71~0xFE	Vendor specific result codes
CRC error	0xFF	CRC error of command message

6.8 Vendor-specific command and response

In addition to the commands identified in this part of ISO/IEC 29173, vendor-specific commands of an RFID interrogator manufacturer may be included in all the categories described above. Preferably, these vendor-specific commands will use the code values in accordance with the categories proposed in this part of ISO/IEC 29173. For example, when a specific command corresponding to a tag read function needs to be added, it will preferably use the code value of 0x71~0xFE that is the vendor-specific area of the tag read category.

6.9 Notification

A notification is a protocol message that is transmitted from the RFID interrogator to the processor of the terminal. Unlike a response message, the notification protocol message is independent of a command. The notification is mainly used as a response for indicating the result of an operation repeated in an automatic mode, and is used for critical errors generated in the RFID interrogator.

In this part of ISO/IEC 29173, the notification protocol message may have the same format as the response protocol message. For example, a value of 0x02 may be used in a message type field to distinguish the notification protocol message from the response protocol message.

In addition, when a critical error is generated in the interrogator, the notification may be used to inform the processor of the error. In this case, a format may be identical to that of a command containing the error, which is identical to what is designated as a notification in a message type field. The critical errors are not defined in this part of ISO/IEC 29173, but may be defined by the vendor. An error that needs to be transmitted by a notification may be defined using a vendor-specific area of a result code.

7 Details of command, response, and notification

7.1 Interrogator control/management category

7.1.1 Interrogator power control

An interrogator power control command is used to control turning on/off power supplied to the hardware of the RFID interrogator. Power is supplied to the interrogator in the on state, while no power is supplied to the interrogator in the off state.

The interrogator power control command is constructed to include a message type, a code, a payload length, and an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x01 indicating interrogator power control. The payload type is represented by payload type A. The argument is 8-bit power state information, which is represented by 0xFF in an on state and by 0x00 in an off state.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x01	0x00	0x01	0xFF	0x7E	0xNNNN

Figure 4 — Interrogator power control command

Fig. 4 illustrates the structure of a protocol message in a power on state. Specifically, Fig. 4 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

A response to the interrogator power control command is constructed to include a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x01 for the case of success, and by 0xFF for the case of failure. The payload type is represented by payload type A. The argument is represented by a result code 0x00 indicating success and by a result code 0x01 indicating power control failure.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x01	0x00	0x01	0x00	0x7E	0xNNNN

Figure 5 — Interrogator power control response

Fig. 5 illustrates the structure of a protocol message for an interrogator power control response for the case of success. Specifically, Fig. 5 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

7.1.2 Interrogator connection control

An interrogator connection control command is used to connect/disconnect the processor to/from the interrogator. When the processor is connected to the interrogator, the interrogator can receive and process all commands. On the other hand, when the processor is disconnected from the interrogator, the interrogator can process only power/connection control commands. When the interrogator was supplied with power but cannot be connected, the minimum power is supplied.

The interrogator connection control command is constructed to include a message type, a code, a payload length, and an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x02 indicating interrogator connection control. The payload type is represented by payload type A. The argument is 8-bit interrogator connection state information, which is represented by 0xFF in case of connection and by 0x00 in case of disconnection.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x02	0x00	0x01	0xFF	0x7E	0xNNNN

Figure 6 — Interrogator connection control command

Fig. 6 illustrates the structure of a protocol message in a connection state. Specifically, Fig. 6 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

A response to the interrogator connection control command is constructed to include a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x02 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A. The argument is represented by a result code 0x00 indicating success and by a result code 0x02 indicating connection control failure.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x02	0x00	0x01	0x00	0x7E	0xNNNN

Figure 7 — Interrogator connection control response

Fig. 7 illustrates the structure of a protocol message for an interrogator connection control response for the case of success. Specifically, Fig. 7 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

7.1.3 Get interrogator information

A get interrogator Information command is used to get information from the interrogator. The information includes a model name, an S/N, a manufacturer, a use frequency, and the type of tag supported.

The get interrogator information control command is constructed to include a message type, a code, a payload length, and an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x03 indicating the get interrogator information command. The payload type is represented by payload type A. The argument is an 8-bit information type data indicating the type of information to be requested from the interrogator, which may include an interrogator model name (0x00), an interrogator S/N (0x01), an interrogator manufacturer (0x02), an interrogator use frequency (0x03).

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x03	0x00	0x01	0x02	0x7E	0xNNNN

Figure 8 — Get interrogator information command

Fig. 8 illustrates the structure of a protocol message when the interrogator manufacturer is requested. Specifically, Fig. 8 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

A response to the get interrogator information command is constructed to include a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x03 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type B in case of model name, S/N, manufacturer and frequency (MHz unit), and by payload type A in case of command failure. The argument is represented by a variable-length corresponding string in case of model name, S/N, manufacturer and frequency, by 'Bit OR' in case of supporting plurality, and by a result code 0x03 indicating cannot get interrogator info in case of command failure.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)			
0xBB	0x01	0x03	0x00	0x0E	0x4C(L)	0x47(G)	0x20( )
Argument							
0x45(E)	0x4C(L)	0x45(E)	0x43(C)	0x54(T)	0x52(R)	0x4F(O)	0x4E(N)
			End Mark	CRC-16			
0x49(I)	0x43(C)	0x53(S)	0x7E	0xNNNN			

Figure 9 — Get interrogator information response – String argument

Fig. 9 illustrates the structure of a protocol message for a get interrogator information response when the manufacture is 'LG ELECTRONICS'. Specifically, Fig. 9 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x03	0x00	0x01	0x03	0x7E	0xNNNN

Figure 10 — Get interrogator information response – Type argument

Fig. 10 illustrates a structure of response when the response is “Cannot get interrogator info”.

7.1.4 Get signal strength

A get signal strength command is used to get a currently-set RF signal strength of an RFID interrogator. The signal strength can be represented in percentage, and the maximum signal strength the interrogator can provide can be regarded as 100%.

The get signal strength command includes a message type, a code and a payload length, but does not include an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x04 indicating get signal strength.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x04	0x00	0x00	0x7E	0xNNNN

Figure 11 — Get signal strength command

Fig. 11 illustrates the structure of a protocol message for a get signal strength command. Specifically, Fig. 11 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an end mark field, and a CRC field.

A response to the get signal strength command is constructed to include a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is

represented by 0x04 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A. The argument is represented by 0 ~ 100 (0x00 ~ 0x64) indicating the signal strength in percentage, and by a result code 0x04 indicating cannot get signal strength.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x04	0x00	0x01	0x4B	0x7E	0xNNNN

Figure 12 — Get signal strength response

Fig. 12 illustrates the structure of a protocol message for a get signal strength response for the case of success when the signal strength is 75%. Specifically, Fig. 12 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

7.1.5 Set signal strength

A set signal strength command is used to set RF signal strength of the interrogator. The signal strength can be represented in percentage, and the maximum signal strength the interrogator can provide can be regarded as 100%.

The set signal strength command includes a message type, a code, a payload length, and an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x05 indicating set signal strength. The payload type is represented by payload type A. The argument is represented by 0x00 ~ 0x64 (0 ~ 100) indicating an 8-bit signal strength value.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x05	0x00	0x01	0x32	0x7E	0xNNNN

Figure 13 — Set signal strength command

Fig. 13 illustrates the structure of a protocol message for a set signal strength command when the signal strength is 50%. Specifically, Fig. 13 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

A response to the set signal strength command is constructed to include a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x05 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A. The argument is represented by a result code 0x00 indicating success, and by a result code 0x05 indicating signal strength control failure.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x05	0x00	0x01	0x00	0x7E	0xNNNN

Figure 14 — Set signal strength response

Fig. 14 illustrates the structure of a protocol message for a get signal strength response for the case of success. Specifically, Fig. 14 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

7.1.6 Get region

A get region command is used to get region/nation information set in the interrogator. That is, since the radio wave standard for the RFID interrogator can be different depending on the nations and regions, the get region command is used to get such region/nation information.

The get region command includes a message type, a code and a payload length, but does not include an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x06 indicating get region.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x06	0x00	0x00	0x7E	0xNNNN

Figure 15 — Get region command

Fig. 15 illustrates the structure of a protocol message for the get region command. Specifically, Fig. 15 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an end mark field, and a CRC field.

A response to the get region command is constructed to include a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x06 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A. The argument is represented by an 8-bit value indicating a region or a nation set in the interrogator, and by a result code 0x06 indicating cannot get region. For example, Korea, America, Europe, Japan, and China are represented by 0x01, 0x02, 0x04, 0x08, and 0x10, respectively.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x06	0x00	0x01	0x01	0x7E	0xNNNN

Figure 16 — Get region response

Fig. 16 illustrates the structure of a protocol message for a get region response when a region set in the interrogator is Korea. Specifically, Fig. 16 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

7.1.7 Set region

A set region command is used to set region/nation information in the interrogator. That is, since the radio wave standard the RFID interrogator can use can be different according to nations and regions, the set region command is used to set such region/nation information.

The set region command includes a message type, a code, a payload length, and an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x07 indicating Set Region. The payload type is represented by an 8-bit value indicating a region set in the interrogator, which is identical to that of get region.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x07	0x00	0x01	0x01	0x7E	0xNNNN

Figure 17 — Set region command

Fig. 17 illustrates the structure of a protocol message for the set region command when a nation set in the interrogator is Korea, which may include values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

A response to the set region command is constructed to include a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x07 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A. The argument is represented by a result code 0x00 indicating success and by a result code 0x07 indicating region control failure.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x07	0x00	0x01	0x00	0x7E	0xNNNN

Figure 18 — Set region response

Fig. 18 illustrates the structure of a protocol message for a set region response when a region set in the interrogator is Korea. Specifically, Fig. 18 illustrates values of a preamble field, a message type field, a code

field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

7.1.8 Reset interrogator

A reset interrogator command is used to promptly stop all operations of the interrogator and initialize the interrogator. Upon completion of the initialization, a response to the reset interrogator command is transmitted to the interrogator. Right after execution of the reset interrogator command, the aforementioned interrogator connection control command shall be used to connect the interrogator since the interrogator is initialized to a state where only power is supplied.

The reset interrogator command includes a message type, a code and a payload length, but does not include an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x08, indicating reset interrogator.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x08	0x00	0x00	0x7E	0xNNNN

Figure 19 — Reset interrogator command

Fig. 19 illustrates the structure of a protocol message for the reset interrogator command, which may include values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an end mark field, and a CRC field.

A response to the reset interrogator command is constructed to include a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x08 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A. The argument is represented by a result code 0x00 indicating success and by a result code 0x08 indicating cannot reset interrogator.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x08	0x00	0x01	0x00	0x7E	0xNNNN

Figure 20 — Reset interrogator response

Fig. 20 illustrates the structure of a protocol message for a response to the reset interrogator command in case of success. Specifically, Fig. 20 illustrates values of a preamble field, a message type field, a code field, a payload length field MSB, a payload length field LSB, an argument field, an end mark field, and a CRC field.

7.1.9 Get type C A/I select parameters

A get type C A/I select parameters command is used to get A/I select parameters related to ISO/IEC 18000-63. The get type C A/I select parameters command includes a message type, a code and a payload length, but does not include an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x09 indicating get type C A/I select parameters.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x09	0x00	0x01	0x7E	0xNNNN

Figure 21 — Get type C A/I select parameters command

Fig. 21 illustrates the structure of a protocol message for the get type C A/I select parameters command.

A response to the get type C A/I select parameters command includes a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x09 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type C, and by payload type A in case of failure.

In case of failure, the argument is represented by a result code 0x09. In case of success, the argument is represented by a 3-bit target value to which a parameter is applied [Inventoried S0 (0b000), Inventoried S1 (0b001), Inventoried S2 (0b010), Inventoried S3 (0b011), SL (0b100)], a 3-bit action value defined in type C, a binary value indicating a memory bank of a tag [RFU (0b00), UII (0b01), TID (0b10), User (0b11)], a 32-bit start address pointer of a tag memory to be compared, an 8-bit length value of the tag memory to be compared, a 1-bit truncated flag representing Enable (0b1) and Disable (0b0), a 7-bit RFU (use a reserved value of 0b0000000), and a bit mask (0~255 bits) defined in type C.

Preamble	Msg Type	Code	PL (MSB)		PL (LSB)	T	A	M	PNTR(MSB)	
0xBB	0x01	0x09	0x00		0x0B	000	000	11	0x00	0x00
	PNTR (LSB)	Length	T	RFU	Mask(MSB)					Mask(LSB)
0x00	0xFF	0x20	0	0000000	0xFF	0xFF			0x00	0x00
End Mark	CRC-16									
0x7E	0xNNNN									

Figure 22 — Get type C A/I select parameters response

Fig. 22 illustrates the structure of a response protocol message to the get type C A/I select parameters command in case that Target=S0, Action=assert SL or inventoried -> A, MB=User memory bank, Pointer=0x000000FF, Length=0x20, T=0, and Mask=0b11111111111111110000000000000000.

7.1.10 Set type C A/I select parameters

A set type C A/I select parameters command is used to set A/I select parameters related to ISO/IEC 18000-63. The set type C A/I select parameters command includes a message type, a code, a payload length, and an argument.

The message type is represented by 0x00 indicating a command. The code is represented by 0x0A indicating set type C A/I select parameters. The payload type is represented by payload type C.

The argument is represented by a 3-bit target value to which a parameter is applied [Inventoried S0 (0b000), Inventoried S1 (0b001), Inventoried S2 (0b010), Inventoried S3 (0b011), SL (0b100)], a 3-bit action value defined in Type C, a 2-bit value indicating a memory bank of a tag [RFU (0b00), UII (0b01), TID (0b10), User (0b11)], a 32-bit start address pointer of a tag memory to be compared, an 8-bit length value of the tag memory to be compared, a 1-bit truncated flag representing Enable (0b1) and Disable (0b0), a 7-bit RFU (use a reserved value of 0b0000000), and a bit mask (0~255 bit) defined in type C.

Preamble	Msg Type	Code	PL (MSB)		PL (LSB)	T	A	M	PNTR (MSB)	
0xBB	0x00	0x0A	0x00		0x0B	000	000	11	0x00	0x00
	PNTR (LSB)	Length	T	RFU	Mask(MSB)					Mask(LSB)
0x00	0xFF	0x20	0	0000000	0xFF	0xFF			0x00	0x00
End Mark	CRC-16									
0x7E	0xNNNN									

Figure 23 — Set type C A/I select parameters command

Fig. 23 illustrates the structure of a protocol message for the set type C A/I select parameters command in case that Target=S0, Action=assert SL or inventoried -> A, MB=User, Pointer=0x000000FF, Length=0x20, T=0, and Mask=0b11111111111111110000000000000000.

A response to the set type C A/I select parameters command includes a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x0A in case of Success, and by 0xFF in case of failure. The payload type is represented by payload type A.

The argument is represented by a result code 0x00 in case of success, and by a result code 0x09 in case of cannot control type C A/I parameters.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x0A	0x00	0x01	0x00	0x7E	0xNNNN

Figure 24 — Set type C A/I select parameters response

Fig. 24 illustrates the structure of a response protocol message to the set type C A/I select parameters command.

7.1.11 Get type C A/I query-related parameters

A get type C A/I query-related parameters command is used to get A/I query-related parameters related to ISO/IEC 18000-63.

The get type C A/I query-related parameters command includes a message type, a code and a payload length, but does not include an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x0B indicating get type C A/I query-related parameters.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x0B	0x00	0x00	0x7E	0xNNNN

Figure 25 — Get type C A/I query-related parameters command

Fig. 25 illustrates the structure of a protocol message for the get type C A/I query-related parameters command.

A response to the get type C A/I query-related parameters command includes a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x0B in case of success, and by 0xFF in case of failure. The payload type is represented by payload type D in case of success, and by payload type A in case of failure. In case of cannot control type C A/I parameters, the argument is represented by a result code 0x09.

In case of success, the argument is represented by a 1-bit value indicating DR (TRcal divide ratio) (if DR is “8” or “64/3", the 1-bit value is set to '0b0' or '0b1', respectively), a 2-bit value M indicating the number of cycles per symbol (if the number of cycles is 1, 2, 4, or 8, M is set to '0b00', '0b01', '0b10', or '0b11', respectively), a 1-bit Trex value (if Pilot Tone exists, the value is set to '0b1'; if not, the value is set to '0b0'), a 2-bit Sel value (A11:'0b00' or '0b01'; ~SL:'0b10'; and SL:'0b11'), a 2-bit session value (S0:'0b00'; S1:'0b01'; S2:'0b10'; and S3:'0b11'), a 1-bit target value (A:'0b0'; and B: '0b1'), a 4-bit value Q indicating the number of slots per round, and a 3-bit UpDn value (if Q is unchanged, it is set to '0b000'; if Q=Q+1, Q is set to '0b110'; and if Q=Q-1, Q is set to '0b011').

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	DR	M	TR	Sel	S	T	Q	UpDn
0xBB	0x01	0x0B	0x00	0x00	0	00	0	00	00	0	000	000
End mark	CRC-16											
0x7E	0xNNNN											

Figure 26 — Get type C A/I query-related parameters response

Fig. 26 illustrates the structure of a response protocol message to the get type C A/I query-related parameters command for the case where DR=8, M=1, Trex=no pilot tone, Sel=A11, Session=S0, Target=A, Q=8, and UpDn=not changed.

7.1.12 Set type C A/I query-related parameters

A set type C A/I query-related parameters command is used to set A/I query-related parameters related to ISO/IEC 18000-63.

The set type C A/I query-related parameters command includes a message type, a code, a payload length, and an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x0C indicating set type C A/I query-related parameters. The payload type is represented by payload type D. The argument is represented by a 1-bit value indicating DR (TRcal divide ratio) (if DR is 8 or 64/3, the 1-bit value is set to '0b0' or '0b1', respectively), a 2-bit value M indicating the number of cycles per symbol (if the number of cycles is 1, 2, 4, or 8, M is set to '0b00', '0b01', '0b10', or '0b11', respectively), a 1-bit Trex value (if Pilot Tone exists, the value is set to '0b1'; if not, the value is set to '0b0'), a 2-bit Sel value (A11:'0b00' or '0b01'; ~SL:'0b10'; and SL:'0b11'), a 2-bit session value (S0:'0b00'; S1:'0b01'; S2:'0b10'; and S3:'0b11'), a 1-bit target value (A:'0b0'; and B: '0b1'), a 4-bit value Q indicating the number of slots per round, and a 3-bit UpDn value (if Q is unchanged, it is set to '0b000'; if Q=Q+1, Q is set to '0b110'; and if Q=Q-1, Q is set to '0b011').

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	DR	M	TR	Sel	S	T	Q	UpDn
0xBB	0x00	0x0C	0x00	0x02	0	00	0	00	00	0	000	000
End mark	CRC-16											
0x7E	0xNNNN											

Figure 27 — Set type C A/I query-related parameters command

Fig. 27 illustrates the structure of a protocol message for the set type C A/I query-related parameters command for the case where DR=8, M=1, Trex=no pilot tone, Sel=A11, Session=S0, Target=A, Q=8, and UpDn=not changed.

A response to the set type C A/I query-related parameters command includes a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x0C in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A. The argument is represented by a result code 0x00 in case of success, and by a result code 0x09 in case of cannot control type C A/I parameters.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x0C	0x00	0x01	0x00	0x7E	0xNNNN

Figure 28 — Set type C A/I query-related parameters response

Fig. 28 illustrates the structure of a response protocol message to the set type C A/I query-related parameters command.

7.1.13 Get automatic read parameters

A get automatic read parameters command is used to get automatic tag read parameters.

The get automatic read parameters command includes a message type, a code and a payload length, but does not include an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x0D indicating get automatic read parameters.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x0D	0x00	0x00	0x7E	0xNNNN

Figure 29 — Get automatic read parameters command

Fig. 29 illustrates the structure of a protocol message for the get automatic read parameters command.

A response to the get automatic read parameters command includes a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x0D in case of success, and by 0xFF in case of failure. The payload type is represented by payload type E in case of success, and by payload type A in case of failure. In case of success, the argument may include a 16-bit read cycle value indicating the number of times of read operation performed by the interrogator, and an 8-bit read delay time value representing a delay (expressed in msec) between read

operations performed by the interrogator. In case of cannot get automatic parameters, the argument may include a result code 0x0D. In case of not supported command, the payload may include a result code 0x62.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg1 (MSB)	Arg1 (LSB)	Arg2
0xBB	0x01	0x0D	0x00	0x03	0x00	0x32	0x32
End Mark	CRC-16						
0x7E	0xNNNN						

Figure 30 — Get automatic read parameters response

Fig. 30 illustrates the structure of a response protocol message to the set automatic read parameters command for the case where Read Cycle=50, and Read Delay Time=50msec.

7.1.14 Set automatic read parameters

A set automatic read parameters command is used to set automatic read parameters.

The set automatic read parameters command includes a message type, a code, a payload length, an argument, End Mark, and CRC. The message type is represented by 0x00 indicating a command. The code is represented by 0x0E indicating set automatic read parameters. The payload type is represented by payload type E. The argument may include a 16-bit read cycle value indicating the number of times of read operation performed by the interrogator, and an 8-bit read delay time value representing a delay (msec) between read operations performed by the interrogator.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg1 (MSB)	Arg1 (LSB)	Arg2
0xBB	0x00	0x0E	0x00	0x03	0x00	0x32	0x32
End Mark	CRC-16						
0x7E	0xNNNN						

Figure 31 — Set automatic read parameters command

Fig. 31 illustrates the structure of a protocol message for the set automatic read parameters command for the case where Read Cycle=50, and Read Delay Time=50msec.

A response to the set automatic read parameters command includes a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x0E in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A. The argument is represented by a result code 0x00 in case of success, and by a result code 0x0E in case of set automatic parameter failure. When Read Cycle and Read Delay Time have invalid parameters, the argument is represented by a result code 0x61. In case of not supported command, the argument is represented by 0x62.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x0E	0x00	0x01	0x00	0x7E	0xNNNN

Figure 32 — Set automatic read parameters response

Fig. 32 illustrates the structure of a response protocol message for the case of success.

7.1.15 Set security key

A set security key command is used to set security key or access password related to ISO/IEC 18000-63. The set security key command includes a message type, a code, a payload length, and an argument.

The message type shall be represented by 0x00 indicating a command. The code shall be represented by 0x0F indicating set security key. The payload type shall be represented by payload type F.

The arguments are consistent with an 8-bit select security mode (SelSeM) and a 32-bit security key value (SeKey) or access password which is sent to an interrogator. The value of select security mode field could be

selected by 0x00 in case of data security mode disable (default), 0x01 in case of data security mode enable, 0x10 in case of access password for reserved memory bank of tag under security mode disable, 0x11 in case of access password of tag under security mode enable. The security key value (SeKey) could be ESN, Phone number, and user defined key.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	SelSeM	SeKey	
0xBB	0x00	0x0F	0x00	0x05	0x01	0x80	0x20
		End Mark	CRC-16				
0x21	0x80	0x7E	0xNNNN				

Figure 33 — Set security key command – ESN for security key

Fig. 33 illustrates the structure of a protocol message for the set security key command in case that SelSeM (Select Security Mode)=0x01, SeKey (Security Key)=0x80202180.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	SelSeM	SeKey	
0xBB	0x00	0x0F	0x00	0x05	0x10	0x12	0x34
		End Mark	CRC-16				
0x56	0x78	0x7E	0xNNNN				

Figure 34 — Set security key command – Access password

Fig. 34 illustrates the structure of a protocol message for the set security key command in case that SelSeM (Select Security Mode)=0x10, Access Password=0x12345678

A response to the set security key command includes a message type, a code, a payload length, and an argument. The message type shall be represented by 0x01 indicating a response. The code shall be represented by 0x0F in case of success, and by 0xFF in case of failure. The payload type shall be represented by payload type A.

The argument shall be represented by a result code 0x00 in case of success, and by a result code 0x0F in case of set security key command fail. In case of not supported command, the argument shall be represented by 0x62.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x0F	0x00	0x01	0x00	0x7E	0xNNNN

Figure 35 — Set type C A/I select parameters response

Fig. 35 illustrates the structure of a response protocol message to the set security key command.

7.1.16 Set frequency mode

This command is for testing purposes. A set frequency mode command is used to set the frequency mode or to set a special frequency. The set frequency mode command includes a message type, a code, a payload length, and an argument.

The message type shall be represented by 0x00 indicating a command. The code shall be represented by 0x10 indicating set frequency mode. The payload type shall be represented by payload type G.

The arguments are consisted of an 8-bit Arg1 and a 24-bit Arg2. Arg1 is used for the select frequency mode (SelFM), which shall be represented by a 0x00 in case of using a frequency-hopping mode (default), by a 0x01 in case of using a LbT mode, and by 0x02 in case of using a special one frequency. And Arg2 is used by a LbT threshold (dBm unit) if an Arg1 was the 0x01, or by a frequency (kHz unit) if an Arg1 was the 0x02.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	SelFM	Arg2	
0xBB	0x00	0x10	0x00	0x04	0x02	0x0E	0x0A
	End Mark	CRC-16					
0x3D	0x7E	0xNNNN					

Figure 36 — Set frequency mode command

Fig. 36 illustrates the structure of a protocol message for the set frequency mode command in case that SelFM (Select Frequency Mode)=0x02 for a special frequency, Arg2 (frequency)=0x0E0A3D (920.125 MHz).

A response to the set frequency mode command includes a message type, a code, a payload length, and an argument. The message type shall be represented by 0x01 indicating a response. The code shall be represented by 0x10 in case of success, and by 0xFF in case of failure. The payload type shall be represented by payload type A.

The argument shall be represented by a result code 0x00 in case of success, and by a result code 0x10 in case of set frequency mode command fail. In case of not supported command, the argument shall be represented by 0x62.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x10	0x00	0x01	0x00	0x7E	0xNNNN

Figure 37 — Set frequency mode response

Fig. 37 illustrates the structure of a response protocol message to the set frequency mode command.

7.1.17 Set MAC control

A set MAC control command is used to set MAC (Medium Access Control), especially related to ISO/IEC 29143. The set MAC control command includes a message type, a code, a payload length, and an argument.

The message type shall be represented by 0x00 indicating a command. The code shall be represented by 0x11 indicating set MAC control. The payload type shall be represented by payload type H.

The arguments are consisted of an 8-bit Arg1 and an 8-bit Arg2. Arg1 is used for enable/disable of MAC function (EnMAC), which shall be represented by a 0x00 in case of enable a MAC function (default), by a 0x01 in case of disable a MAC function. The MSB 4 bits of Arg2 are used for an amplitude threshold of interference interrogator, which range from 0x0 (0%) to 0xA (100%, default). The LSB 4 bits of Arg2 are used for an average threshold on the collision count by interference interrogator, which range from 0x0 to 0xF (default 0x7).

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	EnMAC	Arg2
0xBB	0x00	0x11	0x00	0x02	0x00	0xA7
End Mark	CRC-16					
0x7E	0xNNNN					

Figure 38 — Set MAC control command

Fig. 38 illustrates the structure of a protocol message for the set MAC control command in case that EnMAC=0x00 (Enable MAC), amplitude threshold of interference interrogator=0xA, average threshold of collision count=0x7.

A response to the set MAC control command includes a message type, a code, a payload length, and an argument. The message type shall be represented by 0x01 indicating a response. The code shall be represented by 0x11 in case of success, and by 0xFF in case of failure. The payload type shall be represented by payload type A.

The argument shall be represented by a result code 0x00 in case of success, and by a result code 0x11 in case of set MAC control command fail. In case of Not Supported command, the argument shall be represented by 0x62.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x11	0x00	0x01	0x00	0x7E	0xNNNN

Figure 39 — Set MAC control response

Fig. 39 illustrates the structure of a response protocol message to the Set MAC control command.

7.2 Tag read category

7.2.1 Read tag once

A read tag once command is used to read the memory of an ISO/IEC 18000-63, Type C tag. The ISO/IEC 18000-63, Type C tag has four memory banks for the Reserved, UII, TID and User.

The read tag once command includes a message type, a code, a payload length, a payload, the end mark, and CRC-16. The message type should be represented by 0x00 indicating a command. The code should be represented by 0x21. The payload type should be represented by payload type I. The payload should include a 8-bits MB (Memory Bank to read. 0x00=reserved bank, 0x01=UII bank, 0x02=TID bank, 0x03=user memory bank), 8-bits MB01ECL (MB01EC Length, word unit), MB01EC, 16-bits MSA (Memory Start Address), 8-bits MDL (Data Length to read, word unit) and 32-bits AP (Access Password).

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	MB	MB01ECL
0xBB	0x00	0x21	0x00	0x16	0x03	0x06
MB01EC(MSB)	...	MB01EC(LSB)	MSA(MSB)	MSA(LSB)	MDL	AP(MSB)
0x96	...	0x01	0x03	0x00	0x16	0x12
		AP(LSB)	End Mark	CRC-16		
0x34	0x56	0x78	0x7E	0xNNNN		

Figure 40 — Read tag once command for user memory bank

Fig. 40 illustrates the structure example of a protocol message for the read tag once command for reading a data in user memory bank. (MB: 0x03 = user memory bank, MB01ECL: 0x06 = 6 words = 96 bits, MB01EC: 96 bits, MSA: 0x0300 = start address, MDL: 0x16 = 22 words, AP : 0x12345678).

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	MB	MB01ECL
0xBB	0x00	0x21	0x00	0x0C	0x01	0x00
MSA(MSB)	MSA(LSB)	MDL	AP(MSB)			AP(LSB)
0x00	0x00	0x00	0x00	0x00	0x00	0x00
End Mark	CRC-16					
0x7E	0xNNNN					

Figure 41 — Read tag once command for UII memory bank

Fig. 41 illustrates the structure example of a protocol message for the read tag once command for reading a UII memory bank. (MB: 0x01 = UII memory bank, MB01ECL : 0x00, no MB01EC, MSA : 0x0000, MDL : 0x00, AP : 0x00000000). In case of reading UII memory bank, In this case, the values from the MB01ECL field to AP field mean nothing to interrogator.

The response message for the read tag once command includes a message type, a code, a payload length, a payload, the end mark, and CRC-16. The message type should be represented by 0x01 indicating a response. The code should be represented by 0x21 in case of success, and by 0xFF in case of failure. The payload type should be represented by payload type J in case of success, and by payload type A in case of failure or no tag detected.

In case of success, the payload consists of the MB (Memory Bank), TN (the number of Tag), MDL (Data Length, word unit), MD (Tag memory bank data). The MDL and MD shall be repeated as the number of TN.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	MB	TN
0xBB	0x01	0x21	0x00	0x27	0x01	0x02
MDL#1	MD#1(MSB)	…	MD#1(LSB)	MDL#2	MD#2(MSB)	…
0x08	0x28	…	0x01	0x08	0x28	…
MD#2(LSB)	End Mark	CRC-16				
0x02	0x7E	0xNNNN				

Figure 42 — Read tag once response for UII memory bank

Fig. 42 illustrates the structure example for a response protocol message in case of the command was the read tag once for reading the UII memory bank, The meaning of the arguments in the payload is that the MB (Memory bank) is 0x01 (UII memory bank), TN (Tag number) is 0x02 (two tags), MDL#1 (Data Length of tag 1) is 0x08 (8 words), MD#1 (UII memory bank data of tag1) is 0x28…01, MDL#2 (Data Length of tag2) is 0x08 (8 words), MD#2 (UII memory bank data of tag2) is 0x28…02.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	MB	TN
0xBB	0x01	0x21	0x00	0x32	0x03	0x01
MDL#1	MD#1(MSB)	…	MD#1(LSB)	End Mark	CRC-16	
0x16	0x22	…	0x01	0x7E	0xNNNN	

Figure 43 — Read tag once response for user memory bank

Fig. 43 illustrates the structure example of a response protocol message in case of the command was the read tag once for reading the user memory bank, The meaning of the arguments in the payload is that the MB (Memory bank) is 0x03 (user memory bank), TN (Tag number) is 0x01 (one tag), MDL#1 (Data Length of tag 1) is 0x16 (22 words), MD#1 (user memory bank data of tag1) is 0x22…01.

The Fig. 44 illustrates the structure of a response protocol message for the case of failure.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xFF	0x00	0x01	0x21	0x7E	0xNNNN

Figure 44 — Read tag once response - Failure

7.2.2 Read multiple tags

A read multiple tags command is used to read {UII}s or {UII and other data}s in TID or user memory bank. The response for this command can send the data repeatedly by units of 10 tags.

A protocol message constituting this command shall include a message type, a code, a payload length, a payload, an end mark, and a CRC-16. The message type should be represented by 0x00 indicating a command. The code shall be represented by 0x22 indicating read multiple tags. The payload type shall be represented by payload type K.

The payload consists of an 8-bits RC (Repeat Cycle for inventory), 8-bits NUMT (Number of Maximum tag to read), 8-bits MTIME (Maximum elapsed Time for tagging, second unit), 5-bit RSV (Reserved bits), 2-bits DT (Data type), 1 bit AM (tag memory Access Mode), 32-bits AP (Access Password), 16-bits MSA (Memory Start Address of TID or Content name), 8-bits MDL (Memory data Length to read, word unit). The RC field indicates the number of times for repetition of the read cycle defined in the set automatic read parameters command. (i.e., the number of inventory rounds for the interrogator = ReadCycle x RepeateCycle). The NUMT and MTIME could be represented by 0x00 for tagging regardless of the number of tags or elapsed time. But if the MTIME field is 0xFF, the tagging shall be stopped only by the stop read multiple tags command. The priority between RC, NUMT and MTIME is NUMT > MTIME > RC. MTIME has the highest priority. The DT field indicates that binary “0b01” is UII bank data only (default) and “0b10” is UII bank data and TID data defined by MSA and MDL, and “0b11” is UII bank data and content name in user memory bank, and “0b00” is not used. If the DT field is “0b01”, the other fields from the next DT are not valid to the interrogator and the response includes only a set of UII memory bank data. If the DT field is “0b10”, then the response includes the set of UII

memory bank data and the data in TID memory bank. If the DT field is “0b11”, then the response includes the set of UII memory bank data and the content name data in the user memory bank as described in ISO/IEC 29143. The AM field indicates that binary ‘0b0’ is for indirect mode (default) and ‘0b1’ is for direct access mode of tag’s user memory as described in ISO/IEC 29143.

Preamble	Msg Type			Code	PL (MSB)	PL (LSB)	RC	NUMT
0xBB	0x00			0x22	0x00	0x0D	0x05	0x07
MTIME	RSV	DT	AM	AP(MSB)			AP(LSB)	MSA(MSB)
0x05		01	0	0x12	0x34	0x56	0x78	0x03
MSA(LSB)	MDL			End Mark	CRC-16			
0x00	0x02			0x7E	0xNNNN			

Figure 45 — Read multiple tags command

Fig. 45 illustrates the structure example of a protocol message for the read multiple tags command for the case where RC (Repeat Cycle) = 5 times, NUMT (Number of Maximum tag to read) = 0x07 (seven tags), MTIME = 0x05 (5 seconds), DT (Data type of response) = “0b01” (only UII memory bank), AM (Access Mode) = indirect mode, AP (Access Password) = 0x12345678, MSA (Start Address) = 0x0300, MDL (Memory data Length to read) = 0x02 (2 word). In this case, because the DT is “0b01”, the fields from AM to MDL mean nothing to the interrogator.

A response protocol message for the read multiple tags command includes a message type, a code, a payload length, a payload, an end mark, and CRC-16. The message type shall be represented by 0x01 indicating a response. The code shall be represented by 0x22 in case of success, and by 0xFF in case of failure. The payload type can be represented by payload type L. The payload consists of 8-bits DT (Data Type), 8-bits TN (the number of tag), 8-bit MB01ECL (UII memory bank Data Length, word unit), MB01EC (UII memory bank Data, PC+MII + XPC_W1 + XPC_W2), 8-bits MDL (Memory Data Length, word unit), MD (Memory data of TID memory bank or content name of user memory bank). The MB01ECL, MB01EC, MDL, MD fields shall be repeated by the number of TN (Tag number) field. The usage of DT field is the same as the read multiple tags command. If the value of DT field is 0x01, the MB01ECL and MB01EC fields are related to the UII memory bank data, and MDL is zero and MD does not exist. If the DT is 0x02, the MDL and MD are related to TID memory bank data. And if the DT is 0x03, the MDL and MD are related to the content name in user memory according to ISO/IEC 29143. The maximum number of TN (Tag Number) field shall be less than decimal 11. The interrogator should send the response with TN less than 11 tags. For example, if an interrogator reads 35 tags, then the interrogator should repeat the response message 4 times. The MB01EC field includes the UII memory bank that is consisting of PC + MII + XPC_W1 + XPC_W2. The existence of XPC bits complies with ISO/IEC 18000-63.

When the interrogator tagging is successful, Fig. 46 shows an example of the response message in the case that the DT field is 0x01. In the payload, the first argument, the DT field indicates that the data type is 0x01 (only UII memory bank data) according to command. The TN (Tag number) is three. Therefore the three MB01ECL, MB01EC and MDL fields are followed after the TN field. The fields related to the first tag, MB01ECL#01 (UII memory bank Data Length) is 6 words, and MB01EC#01 (UII memory bank data) is 0x96…01, and MDL#01 is 0x00.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	DT	TN
0xBB	0x01	0x22	0x00	0x2F	0x01	0x03
MB01ECL01	MB01EC01(MSB)	…	MB01EC01(LSB)	MDL01	MB01ECL02	MB01EC02(MSB)
0x06	0x96	…	0x01	0x00	0x06	0x96
…	MB01EC02(LSB)	MDL02	MB01ECL03	MB01EC03(MSB)	…	MB01EC03(LSB)
…	0x02	0x00	0x06	0x96	…	0x03
MDL03	End Mark	CRC-16				
0x00	0x7E	0xNNNN				

Figure 46 — Read multiple tags response example 1 – Success

The Fig. 47 illustrates the response message example in case of that DT field of the read multiple tags command was “0b11”.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	DT	TN
0xBB	0x01	0x22	0x00	0x2D	0x03	0x02
MB01ECL01	MB01EC01(MSB)	…	MB01EC01(LSB)	MDL01	MD01(MSB)	…
0x06	0x96	…	0x01	0x02	0x32	…
MD01(LSB)	MB01ECL02	MB01EC02(MSB)	…	MB01EC02(LSB)	MDL02	MD02(MSB)
0x01	0x06	0x96	…	0x02	0x02	0x02
…	MB01EC02(LSB)	End Mark	CRC-16			
…	0x01	0x7E	0xNNNN			

Figure 47 — Read multiple tags response example 2 – Success

The DT is 0x03 and TN is 0x02. Therefore the UII memory bank and content name in the user memory bank are followed after the TN field. The UII memory bank data of the first tag is 0x96…01, and the content name is 0x32…01. The UII memory bank data of the second tag is 0x96…02, and the content name is 0x02…01.

In case that read multiple tags command is a failure, the code shall be 0xFF, the argument shall include a result code 0x61 for the case of invalid parameter, 0x65 for the case of no tag detected, 0x62 for the case of not support command, 0x22 for the case of command failure

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xFF	0x00	0x01	0x22	0x7E	0xNNNN

Figure 48 — Read multiple tags response – Failure

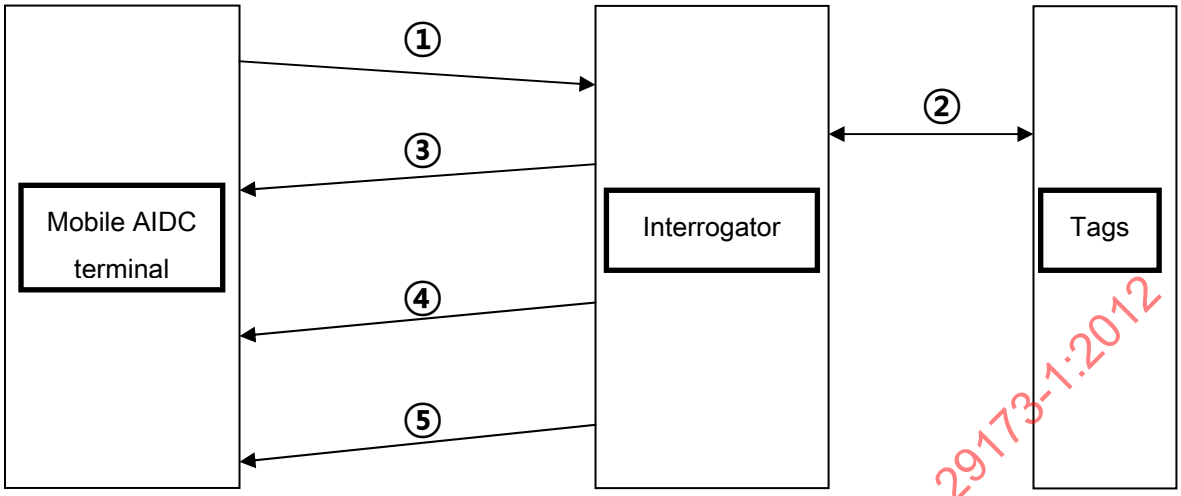
Fig. 48 illustrates the structure of a response protocol message for the case of failure.

A notification message is shown in Fig. 49, can be used for read multiple tags response. This notification message shall include a message type, a code, a payload length, and an argument. The message type shall be represented by 0x02 indicating notification. In the case where the read multiple tags is performed as the predetermined condition (command completed), interrogator could send notification message to AIDC terminal with an argument field included result code 0x24. When there are no more tags to read (No more Tags to Read), the argument could include a result code 0x67.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x02	0x22	0x00	0x01	0x24	0x7E	0xNNNN

Figure 49 — Read multiple tags notification – Read completed

Fig. 50 is shown to the flow scheme of read multiple tags command and response between the AIDC terminal and the interrogator in case of command success.



Key

① Read multiple tags command

② Inventory round or access and read as condition of command

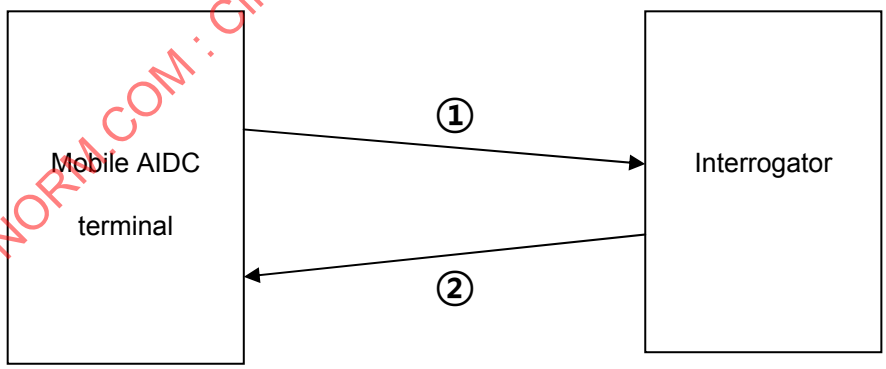
③ Response up to 10 tag information at a time

④ If there is a need to send the tag information will repeat to response

⑤ Notification with read complete

Figure 50 — The flow scheme of read multiple tags operation – Success

Fig.51 is shown to the flow scheme of read multiple tags in case of command failure



Key

① Read multiple tags command

② Response with result code “Fail”

Figure 51 — The flow scheme of read multiple tags operation – Failure

7.2.3 Stop read multiple tags

A stop read multiple tags command is used to stop a read multiple tags operation. The stop read multiple tags command includes a message type, a code and payload length, but does not include a payload. The message type shall be represented by 0x00 indicating a command. The code shall be represented by 0x23 indicating stop read multiple tags.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x23	0x00	0x00	0x7E	0xNNNN

Figure 52 — Stop read multiple tags command

Fig. 52 illustrates the structure of a protocol message for the stop read multiple tags command.

A response protocol message for the stop read multiple tags command includes a message type, a code, a payload length, and an argument. The message type shall be represented by 0x01 indicating a response. The code shall be represented by 0x23 in case of success, and by 0xFF in case of failure. The payload type can be represented by payload type A. In case of success, the argument shall include a result code 0x00. In case of cannot stop read multiple tags, the argument shall include a result code 0x25. In case where a read multiple tags operation is not being performed, the argument shall include a result code 0x26.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x23	0x00	0x01	0x00	0x7E	0xNNNN

Figure 53 — Stop read multiple tags response

Fig. 53 illustrates the structure of a protocol message for the stop read multiple tags response for the case of success

7.3 Tag write category

A write tag command is used to write the data in a tag. The write tag command includes a message type, a code, a payload length, a payload, an end mark, and CRC-16. The message type should be represented by 0x00 indicating a command. The code shall be represented by 0x31 indicating write tag. The payload type shall be represented by payload type M. The payload shall include a 32-bits AP (Access Password), 8-bits MB (Memory Bank), 8-bits MB01ECL (MB01EC Length, word unit), MB01EC, 16-bit MSA (Memory Start Address), 8-bits WDL (Write Data Length, word unit), WD (Write Data). The MB field indicates that 0x00 is reserved memory bank, 0x01 is UII memory bank, 0x03 is user memory bank. The MSA field indicates a start address of the memory bank defined by MB field. And the WDL indicates the size of data to be written, and WD field is a data written corresponding to a length designated by the WDL field.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP(MSB)	
0xBB	0x00	0x31	0x00	0x24	0x12	0x34
	AP(LSB)	MB	MB01ECL	MB01EC(MSB)	…	MB01EC(LSB)
0x56	0x78	0x01	0x06	0x96	…	0x01
MSA(MSB)	MSA(LSB)	WDL	WD(MSB)	…	WD(LSB)	End Mark
0x00	0x00	0x06	0x96	…	0x01	0x7E
CRC-16						
0xNNNN						

Figure 54 — Write tag command

Fig. 54 illustrates the structure example of a protocol message for the write tag command for the case where Access Password=0x12345678, Memory Bank=0x01 (UII memory bank), MB01EC=00x96…01 (6 words), Memory Start Address=0x00, and Data to be written=0x96…01 (6 words).

A response message for the write tag command includes a message type, a code, a payload length, a payload (one argument), an end mark, and CRC-16. The message type should be represented by 0x01 indicating a response. The code should be represented by 0x31 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A. The argument shall include a result code 0x00 for

the case of success, a result code 0x65 for the case of no tag detected, a result code 0x31 for the case of write failure, and a result code 0x62 for the case of not supported command.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x31	0x00	0x01	0x00	0x7E	0xNNNN

Figure 55 — Write tag response

Fig. 55 illustrates the structure of a response protocol message for the write tag command.

7.4 Tag kill/lock/erase category

7.4.1 Kill tag

A kill tag command is used to kill an ISO/IEC 18000-63 tag. An access password and a kill password are both required for the killing operation, which aims at security.

The kill tag command includes a message type, a code, a payload length, a payload, an end mark, and a CRC-16. The message type is represented by 0x00 indicating a command. The code is represented by 0x41 indicating kill tag. The payload type is represented by payload type N.

The payload may include a 32-bit access password required for accessing a tag, a 32-bit kill password required for killing a tag, a 8-bit MB01EC length indicating the length of MB01EC, an MB01EC (variable) indicating an ISO/IEC 18000-63 tag to be killed, and 8-bit RFU/REC for re-commissioning that just LSB 3 bits is valid.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP(MSB)		
0xBB	0x00	0x41	0x00	0x13	0x12	0x34	0x56
AP(LSB)	KP(MSB)			KP (LSB)	MB01ECL	MB01EC(MSB)	...
0x78	0x87	0x65	0x43	0x21	0x06	0x96	...
MB01EC(LSB)	RFU/REC	End Mark	CRC-16				
0x01	0x03	0x7E	0xNNNN				

Figure 56 — Kill tag command

Fig. 56 illustrates the structure of a protocol message for the kill tag command for the case where Access Password=0x12345678, Kill Password=0x87654321, and MII=0x96…01 (6 words).

A response message for the kill tag command includes a message type, a code, a payload length, an argument, end mark, and CRC-16. The message type is represented by 0x01 indicating a response. The code is represented by 0x41 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A.

The argument protocol may include a result code 0x00 for the case of success, a result code 0x65 for the case where there is no tag to be killed (No tag detected), and a result code 0x41 for the case of kill failure.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x41	0x00	0x01	0x00	0x7E	0xNNNN

Figure 57 — Kill tag response

Fig. 57 illustrates the structure of a response protocol message for the kill tag command for the case of success.

7.4.2 Lock tag

A lock tag command is used to control lock of a type C tag. The lock tag command includes a message type, a code, a payload length, a payload, an end mark, and a CRC-16. The message type is represented by 0x00 indicating a command. The code is represented by 0x42 indicating lock tag. The payload type is represented

by payload type O. The payload may include a 32-bit Access Password (AP) required for a locking operation, an 8-bit MB01ECL (MB01EC Length, word unit) indicating the length of MB01EC, an MB01EC (variable) indicating a type C tag to be locked, a 4-bit reserved field (RSV) for future, and a 24-bit Lock Data (LD) for controlling a locking operation (use a 20-bit flag for controlling a locking operation).

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP(MSB)			
0xBB	0x00	0x42	0x00	0x11	0x87		0x65	0x43
AP(LSB)	MB01ECL	MB01EC(MSB)	…	MB01EC(LSB)	RSV	LD		LD(LSB)
0x21	0x06	0x96	…	0x01	0000	0000	0xC0	0x30
End Mark	CRC-16							
0x7E	0xNNNN							

Figure 58 — Lock tag command

Fig. 58 illustrates the structure of a protocol message for the lock tag command for the case where MB01EC=0x96…01 (6 words), Access Password=0x87654321, and a MB01EC code is permanently locked.

A response protocol message for the lock tag command includes a message type, a code, a payload length, an argument, an end mark, and a CRC-16. The message type is represented by 0x01 indicating a response. The code is represented by 0x42 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A.

The argument may include a result code 0x00 for the case of success, a result code 0x65 for the case where there is no tag to be locked (No tag detected), a result code 0x42 for the case of lock control failure, and a result code 0x62 for the case of not supported command.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x42	0x00	0x01	0x00	0x7E	0xNNNN

Figure 59 — Lock tag response

Fig. 59 illustrates the structure of a response protocol message for the lock tag command for the case of success.

7.4.3 Erase tag

An erase tag command is used to erase a data of a type C tag. The erase tag command includes a message type, a code, a payload length, a payload, an end mark, and a CRC-16. The message type should be represented by 0x00 indicating a command. The code should be represented by 0x43 indicating erase tag. The payload type shall be represented by payload type P. The payload should include a 32-bit Access Password (AP) required for an erase operation, a 8-bit MB01ECL (MB01EC Length, word unit) indicating the length of MB01EC, an MB01EC (variable size) indicating a type C tag to be erased, a 6-bit reserved field (RSV) for future, a 2-bit MB (Memory Bank), a 16-bit MSA (Memory Start Address for erase), and a 8-bit MDL (Memory data Length to erase) for controlling a erase operation.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP(MSB)			
0xBB	0x00	0x43	0x00	0x12	0x87		0x65	0x43
AP(LSB)	MB01ECL	MB01EC (MSB)	…	MB01EC (LSB)	RSV	MB	MSA(MSB)	MSA(LSB)
0x21	0x06	0x96	…	0x01	000000	11	0x00	0x10
MDL	End Mark	CRC-16						
0x03	0x7E	0xNNNN						

Figure 60 — Erase tag command

Fig. 60 illustrates the structure of a protocol message for the erase tag command for the case where Access Password=0x87654321, MB01EC=0x96…01 (6 words), MB=0b11 (user memory bank), MSA=0x0010, and MDL=4 word.

A response protocol message for the erase tag command includes a message type, a code, a payload length, an argument, an end mark, and a CRC-16. The message type should be represented by 0x01 indicating a response. The code should be represented by 0x43 in case of success, and by 0xFF in case of failure. The payload type shall be represented by payload type A.

The argument shall include a result code 0x00 for the case of success, a result code 0x65 for the case where there is no tag to be locked (No tag detected), a result code 0x43 for the case of erase failure, and a result code 0x62 for the case of not supported command.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x43	0x00	0x01	0x00	0x7E	0xNNNN

Figure 61 — Erase tag response

Fig. 61 illustrates the structure of a response protocol message for the erase tag command for the case of success.

7.5 Additional function category

7.5.1 Get last result

A get last result command is used to get the last result code. The get last result command includes a message type, a code and a payload length, but does not include an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x51 indicating get last result.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x51	0x00	0x00	0x7E	0xNNNN

Figure 62 — Get last result command

Fig. 62 illustrates the structure of a protocol message for the get last result command.

A response protocol message for the get last result command includes a message type, a code, a payload length, and an argument. The message type is represented by 0x01 indicating a response. The code is represented by 0x51 in case of success, and by 0xFF in case of failure. The payload type is represented by payload type A. The argument may include the last result code for the case of success, a result code 0x51 for the case of cannot get last result, and a result code 0x62 for the case of not supported command.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x51	0x00	0x01	0x09	0x7E	0xNNNN

Figure 63 — Get last result response

Fig. 63 illustrates the structure of a response protocol message for the case where the last result is read failure.

7.5.2 Start test mode

A start test mode command is used to change the RFID interrogator into a test mode. The start test mode command includes a message type, a code and a payload length, but does not include an argument. The message type is represented by 0x00 indicating a command. The code is represented by 0x52 indicating start test mode.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x52	0x00	0x00	0x7E	0xNNNN

Figure 64 — Smart test mode command

Fig. 64 illustrates the structure of a protocol message for the start test mode command.