INTERNATIONAL STANDARD

ISO/IEC 18000-4

Fourth edition
2018-08

Information technology — Radio frequency identification for item management —

Part 4: Parameters for air interface communications at 2,45 GHz

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

Partie 4: Paramètres de communications d'une interface d'air à 2,45 GHz

Reference number
ISO/IEC 18000-4:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: 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.

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

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

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

This fourth edition cancels and replaces the third edition (ISO/IEC 18000-4:2015), which has been technically revised. The main changes compared to the previous edition are as follows:

— Mode 4, described in Clause 9 as "MODE 4: Configurable data rate active RFID system", has been added.

A list of all parts in the ISO 18000 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document is one of a series of International Standards and Technical Reports developed by ISO/IEC JTC 1/SC 31 for the identification of items (item management) using radio frequency identification (RFID) technology.

This document defines four 2,45 GHz protocols. Each of the specific physical/data link configuration is defined in a separate subclause. The configuration descriptions include a physical layer and a data link layer.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning radio frequency identification technology given in all parts of this document.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights.

The holders of these patent rights have assured the ISO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC. Information may be obtained from the following companies.

	Contact details
Patent holder:	
Legal name	iControl Inc.
Contact for licence application:	
Name and department	George Cavage
Address	3235 Kifer Road, Suite 260, Santa Clara, CA 94109, USA
Tel.	+1 408 730 5364
Fax	
E-mail	gcavage@icontrol-inc.com
URL (optional)	www.icontrol-inc.com
Patent holder:	
Legal name	Impinj, Inc.
Contact for licence application:	
Name and department	Stacy Jones, Impinj, Inc.
Address	701 N 34th Street, Suite 300, Seattle, WA 98103, USA
Tel.	+1 206 834 1032
Fax	+1 206 517 5262
E-mail	stacy.jones@impinj.com
URL (optional)	www.impinj.com

Contact details

Patent holder:	
Legal name	Zebra Technologies Corporation
Contact for licence application:	
Name and department	James O'Hagan, Director of Patents & Technology
Address	475 Half Day Road, Suite 500, Lincolnshire, IL 60069, USA
Tel.	+1 (847) 793-6798
Fax	+1 (847) 955-4514
E-mail	johagan@zebra.com
URL (optional)	
Patent holder:	
Legal name	Chengdu West Valley Digital Technologies, Inc.
Contact for licence application:	
Name and department	Jason Y. Liao
Address	Room 305 Building #9, #88 Tian Chen West High-Tech Zone, Chengdu, China
Tel.	86-28-66767921
Fax	
E-mail	
URL (optional)	www.westv.cn
Patent holder:	
Legal name	Nationz Technologies Inc.
Contact for licence application:	
Name and department	Yang, Xianwei and Mobile Payment and RCC Department
Address	China Technology Exchange Building, 66 North 4th Ring Road West, Haidian District, Beijing
Tel.	86-10-82868162
Fax	
E-mail	
URL (optional)	www.nationz.com.cn
Patent holder:	
Legal name	Shenzhen ZTE Top sky Information Technology Co., Ltd.
Contact for licence application:	
Name and department	Min Zhu
Address	Room B, 10/F, No.3 China Academy of Science-Tech Development Building, Gaoxin 1st Road South, High-Tech Industrial Park, Nanshan District, Shenzhen, China
Tel.	86-0755-86079364
Fax	
E-mail	
URL (optional)	www.zte-v.com.cn

Contact details

Patent holder:	
Legal name	Xidian University
Contact for licence application:	
Name and department	Changxing Pei
Address	No.2, South Taibai Road, Xi'an, Shanxi, China
Tel.	86-29-88204486
Fax	
E-mail	
URL (optional)	www.xidian.edu.cn
Patent holder:	
Legal name	CEC Huada Electronic Design Co., Ltd.
Contact for licence application:	
Name and department	Lan Tian
Address	Floor 5, Building A, Wangjing Science and Technology Park, No.2, Lizezhonger Rd., Chaoyang District, 100102, Beijing, China
Tel.	86-10-64365577
Fax	
E-mail	
URL (optional)	www.hed.com.cn
Patent holder:	
Legal name	China Electronics Standardization Institute (Corresponding corporation) CEC Huada Electronic Design Co., Ltd.
Contact for licence application:	
Name and department	Wang Wenfeng
Address	No.1 Andingmen East Street, Beijing
Tel.	86-10-84042998
Fax	
E-mail	
URL (optional)	www.cesi.cn
Patent holder:	
Legal name	Shanghai Super Electronics Technology Co., Ltd.
Contact for licence application:	
Name and department	Qin Zhong
Address	6/F Building 52, 1089 North Qinzhou Rd., Shanghai, China
Tel.	86-21-61613336
Fax	86-21-6161339
E-mail	qinzhong@superrfid.net
URL (optional)	www.superrfid.net

Information technology — Radio frequency identification for item management —

Part 4:
Parameters for air interface communications at 2,45 GHz

1 Scope

This document defines the air interface for radio frequency identification (RFID) devices operating in the 2,45 GHz industrial, scientific, and medical (ISM) band used in item management applications. This document provides a common technical specification for RFID devices that can be used by ISO committees developing RFID application standards. This document is intended to allow for compatibility and to encourage inter-operability of products for the growing RFID market in the international marketplace. This document defines the forward and return link parameters for technical attributes including, but not limited to, operating frequency, operating channel accuracy, occupied channel bandwidth, maximum equivalent isotropically radiated power (EIRP), spurious emissions, modulation, duty cycle, data coding, bit rate, bit rate accuracy, bit transmission order, and where appropriate, operating channels, frequency hop rate, hop sequence, spreading sequence, and chip rate. This document further defines the communications protocol used in the air interface.

This document contains four modes. Mode 1 is an interrogator talks first with passive tag. Mode 2 is a tag talks first with battery-assisted passive tag. Mode 3 is a globally available, ubiquitous network supporting (but not limited to) the logistics and transportation industry; agnostic to any device, commercial or otherwise, requiring global availability. Mode 4 is a configurable data rate active RFID system. It provides the functions of long range objects identification and environmental sense, and it is intended to realize the low cost device and low power consumption, long range identification, fast and reliable tags access. The detailed technical differences between the modes are shown in the parameter tables.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 646, *Information technology — ISO 7-bit coded character set for information interchange*

ISO/IEC 7816-6, *Identification cards — Integrated circuit cards — Part 6: Interindustry data elements for interchange*

ISO/IEC 15963, *Information technology — Radio frequency identification for item management — Unique identification for RF tags*

ISO/IEC 19762, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary*

ISO/IEC/TR 18047-4, *Information technology — Radio frequency identification device conformance test methods — Part 4: Test methods for air interface communications at 2,45 GHz*

ISO/IEC/IEEE 8802-15-4:2018, *Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 15-4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)*

IEEE 802.15.4:2006, *Information technology — Local and metropolitan area networks — Specific requirements — Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)*

3 Terms and definitions

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

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

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

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

3.1
associated
successful negotiation of a bi-directional wireless link with a coordinator

Note 1 to entry: Associated networks require communication be maintained and monitored for a period of time.

3.2
association
service used to establish membership for a device communicating within the wireless network described in this document

3.3
authentication
ATH
process of determining whether an entity or data is/are who or what, respectively, it claims to be

Note 1 to entry: The types of entity authentication referred to in this document are Tag authentication, Interrogator authentication, and Tag-Interrogator mutual authentication. For data authentication, see *authenticated communications* (3.4).

3.4
authenticated communications
communications in which message integrity is protected

3.5
block cipher
cryptographic function that operates on strings of fixed size

3.6
chip
smallest unit of signal after spectrum spreading

Note 1 to entry: In the spreading process, an information bit is usually represented by a number of coded signals, wherein a coded signal is called a chip.

3.7
coordinator
full-function device (FFD) capable of relaying messages

Note 1 to entry: If a coordinator is the principal controller of a personal area network (PAN), it is called the PAN coordinator.

3.8
data server
device data termination point

3.9
device
any entity that meets the ISO/IEC/IEEE 8802-15-4 medium access control (MAC), physical interface to the wireless medium, and this document

Note 1 to entry: A device may be a reduced-function device (RFD) or a full-function device (FFD).

3.10
encryption
transformation of a message into a new representation so that privileged information is required to recover the original representation

3.11
full-function device
FFD
device capable of operating as a coordinator

3.12
group key
key that is known only to a set of devices

3.13
hailing channel
ISO/IEC/IEEE 8802-15-4 radio channel used to broadcast the NDB

3.14
key
privileged information that may be used, for example, to protect information from disclosure to, and/or undetectable modification by, parties that do not have access to this privileged information

3.15
keyID
numerical designator for a single key

3.16
link key
secret key that is shared between precisely two devices

3.17
mesh networking
type of network where an FFD serves as a relay for other devices

3.18
message authentication code
MAC1
code, computed over bits in a message, that an Interrogator or a Tag may use to verify the integrity of the message

3.19
message integrity code
MIC
data whereby an entity receiving a message corroborates evidence about the true source of the information in the message and, thereby, evidence that the message has not been modified in transit

3.20
network channel
primary ISO/IEC/IEEE 8802-15-4 radio channel between a coordinator and remote devices

3.21
packet
formatted, aggregated bits that are transmitted together in time across the physical medium

3.22
password
information that is used to verify identities in the *tag access* (3.29) process

3.23
payload data
contents of a data message that is being transmitted

3.24
reduced-function device
RFD
device that is not capable of operating as a coordinator

3.25
secure communications
communications in which message confidentiality is protected

3.26
security
degree of protection against threats identified in a security policy

Note 1 to entry: A system is secure if it is protected to the degree specified in the security policy.

3.27
server connected coordinator
SCC
network coordinator that terminates the wireless protocol described in this document and is connected to control servers

3.28
tag
any device type associated with the device and capable of joining the network

3.29
tag access
process in which tags establish communication links with interrogators according to certain communication protocols

4 Symbols and abbreviated terms

ACK	acknowledgement
ALW	always
CCITT	Comité Consultatif International Téléphonique et Télégraphique
Cht	Carrier high level tolerance
Clt	Carrier low level tolerance
CRC	cyclical redundancy check
CSI	cryptographic suite identifier
CSMA	carrier sense multiple access
DBPSK	differential binary phase shift keying
DF	dedicated file

DSSS	direct sequence spread spectrum
EBV	extensible bit vectors
EF	element file
EIRP	equivalent isotropic radiated power
EVM	error vector magnitude
$f_{bitrate}$	base frequency of the bit rate of Manchester code without bit changes
f_c	frequency of operating field (carrier frequency)
FCF	frame control field
FCS	frame check sequence
FHSS	frequency hopping spread spectrum
LSB	least significant bit
M	Modulation
Ma	Modulation overshoot
MAC	medium access control
Mb	Modulation undershoot
MF	main directory file
MIC	message integrity code
MIN	Manufacturing Identification Number
Mlt	Modulation lower tolerance
MSB	most significant bit
Mut	Modulation upper tolerance
N	total number of time-slots in access frame
NAK	no-acknowledgement
NDB	network discovery beacon
NEV	never
NSM	network status message
OID	object identifier
O-QPSK	offset quadrature phase shift keying
ppm	parts per million
PWD	password
Q	parameter that controls tag access number

QPSK	quad-phased shift keying
RFID	radio frequency identification
RID	interrogator identifier
RNr	random number generated by interrogators
RNt	random number generated by tags
RN16	16-bit random number
RN8	8-bit random number
RTLS	real time locating system
SN_{slot}	serial number of time-slot
$SN_{success}$	serial number of successful access
Tbmf	Manchester fall time
Tbmr	Manchester rise time
Tcf	carrier fall time
Tcr	carrier rise time
Tcs	carrier steady time
TDMA	time division multiple access
Tf	fall time
Tfhf	carrier FHSS fall time
Tfhr	carrier FHSS rise time
Tfhs	carrier FHSS steady time
Tflb	forward link bit time
Tr	rise time
Trlb	return link bit time
TID	tag identifier
TTL	tag talk last
UII	unique item identifier
Wslot	width of time-slot
$\oplus$	bitwise XOR
||	series connection
•	point multiplication
+	binary addition

5 General items on 2,45 GHz RFID protocols that support this document

5.1 Protocols

This clause describes the general items of the ISO/IEC 18000-4 2,45 GHz RFID command/data level communication protocols. These protocols facilitate communication between compliant tag and compliant interrogator. The timing parameters and signal characteristics for the protocols are defined in the physical link specifications in each mode. Details of the Modes of various protocols are described in Clauses 6, 7, 8 and 9.

5.2 Frequency

5.2.1 General

This document is intended to address RFID devices operating in the 2 450 MHz Industrial, Scientific and Medical (ISM) frequency band.

5.2.2 Interface definitions

This document supports standard parameters and standard air interface implementations for wireless, non-contact information system equipment for Item Management applications. Typical applications operate at ranges greater than 1 m.

5.2.2.1 RFID system definition

The radio frequency identification (RFID) system shall include a host system and RFID equipment (interrogator and tags). The host system runs an application program, which controls interfaces with the RFID. The RFID equipment shall be composed of two principal components: tags and interrogators. The tag is intended for attachment to an item, which a user wishes to manage. It is capable of storing a tag ID number and other data regarding the tag or item and of communicating this information to the interrogator. The interrogator is a device, which communicates to tags in its field of view. Additionally, the interrogator can use its transmitted RF carrier to power the tag. Systems, which rely on the transmitted interrogator carrier for powering the tag, are typically referred to as passive tag systems. The interrogator controls the protocol, reads information from the tag, directs the tag to store data in some cases, and ensures message delivery and validity.

5.2.2.2 Minimum features

RFID systems defined by this document provide the following minimum features:

— identify tag in range;

— read data;

— write data or handle read-only systems gracefully;

— selection by group or address;

— graceful handling of multiple tags in the field of view;

— error detection.

5.2.2.3 Conformance

To claim conformance with this document, an RFID system shall comply with one of the physical/data link implementations described in Clauses 6, 7, 8 and 9.

The rules for RFID device conformity evaluation are given in ISO/IEC 18047-4.

5.3 Tag identification number

A tag identification number shall be included in commands directed to a specific tag unless the protocol provides other means like Tag Talks First (TTF) protocols. This document mandates that each tag shall include a manufacturer's tag identification number as defined in Annex A for mode 1, in Annex C for mode 2, in 8.5.1 for mode 3 and in 9.4.6.1 for mode 4.

A separate User Tag Identification is not mandatory, but is an option. When a UserTagID is used, it shall consist of the number of bytes required by the user application. This number and other application data shall be accessed as user data fields on the tag. These fields can be accessed via the API using the driver's field name resolution mechanism. The UserTagID is a user-defined tag identifier and is not necessarily unique.

5.4 Potential interference

Standards developers have a duty to ensure that no "significant interference" exists between Standardized modes. "Significant Interference" exists if a system of one Standardized mode (working within the most widespread regulated power emissions) is likely to impede the successful operation of a system of another Standardized mode (working within the most widespread regulated power emissions), *in likely expected operating situations.*

Marginal measurable interference that does not impede operation *in likely expected operating situations*, or that could be avoided by simple and inexpensive design improvement, shall not be considered cause to reject a mode.

— Therefore, TTF modes are clearly identified as such in this document.

— Therefore, installers of RFID systems are advised that they should make best efforts to be a good neighbour in installing any systems, bearing in mind that there may be other systems sharing the same bandwidth, and are advised to take precautions to minimize interference to other systems. Installers are equally advised to be prepared to handle interference within the bandwidth from other users up to transmission powers permitted by local regulations.

6 MODE 1: Passive backscatter RFID system

6.1 MODE 1: General

The FHSS backscatter option or the narrow band operation RFID system shall include an interrogator that runs the FHSS backscatter option 1 RFID protocol or in narrow band operation, as well as one or more tags within the interrogation zone.

When placed in the RF field of an interrogator, the tag shall begin to power up. If the field is adequate, the tag shall execute a power-on reset and shall be ready to receive commands. Each command shall begin with a preamble and start delimiters that, taken together, enable the tag to perform clock and data recovery on the incoming signal. Data to and from the tag are checked for errors using a Cyclic Redundancy Code (CRC). Therefore, CRC fields are present in all interrogator interrogations and in all tag responses. Additional data protection is provided by Manchester encoding on the forward (interrogator to tag) link and FM0 encoding on the return (tag to interrogator) link.

By using the FHSS backscatter option 1 RFID command set or in narrow band operation, the interrogator can execute a number of functions on tags in its field. For example, the interrogator can send a command sequence, which allows it to identify multiple tags simultaneously in its RF field. Alternately, it can select a subset of the tags in the field based on tag memory contents. It can also read data stored on a tag in its field, as well as write or lock data to such a tag.

The description of the RFID tag command set in the following clause shall provide detail regarding the command field and return data/acknowledgement fields, if any. In addition, it shall cover additional high-level elements of the FHSS backscatter option RFID protocol, including how the multiple item

identification algorithm works and byte ordering requirements. The more general aspects of the protocol (preambles, CRC-16, etc.) are covered in detail in 6.2.7.

This portion of this document describes a passive backscatter RFID system that supports the following system capabilities:

System protocol

— Identify and communicate with multiple tags in the field

— Select a subgroup of tags to identify or communicate with based on information that the user has stored in the tag

— Read from and write or rewrite data many times to individual tags

— User controlled permanent lock memory

Data integrity protection

— Manchester bit-wise encoding and CRC-16 packet-level protection are applied to the forward link (interrogator-to-tag) data.

— FM0 bit-wise encoding and CRC-16 packet-level protection are applied to the return link (tag-to-interrogator) data.

In this RFID system, interrogators both power and communicate with the tags that are within their range. Tags receive data as on-off key amplitude modulation of the power/data signal from the interrogator. During the time that the tag communicates back to the interrogator, the interrogator broadcasts a steady radio frequency power level, and the tag modulates the impedance of its radio frequency load attached to the tag antenna terminals. The interrogator then receives the data back from the tag as a variation in reflection of its transmitted power.

6.2 Physical layer and data coding

6.2.1 Interrogator power-up waveform

The interrogator power-up waveform shall comply with the mask specified in Figure 1 and Table 1.

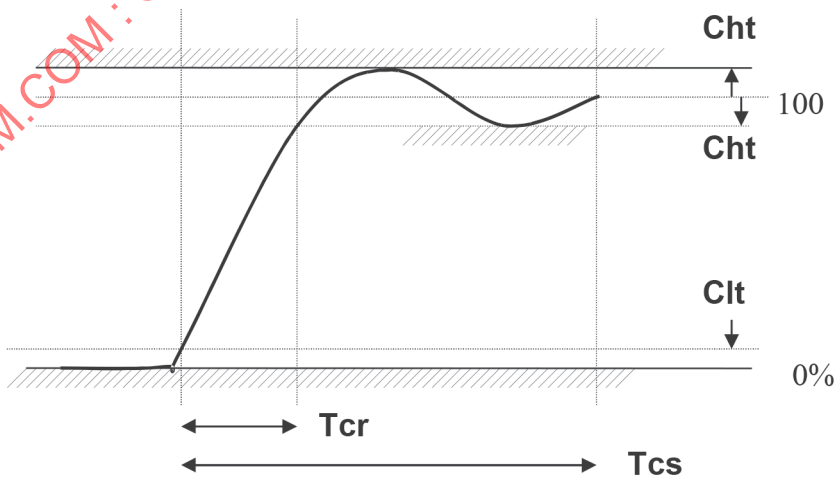


Figure 1 — Interrogator power-up waveform

Table 1 — Interrogator power-up waveform parameter values

Parameter	Min	Max
Tcs		400 µs
Tcr	0 µs	30 µs
Cht		3 %
Clt		1 %

6.2.2 Interrogator power-down

Once the carrier level has dropped below the ripple limit Cht, power down shall be monotonic and of duration Tcf, as specified in Figure 2 and Table 2.

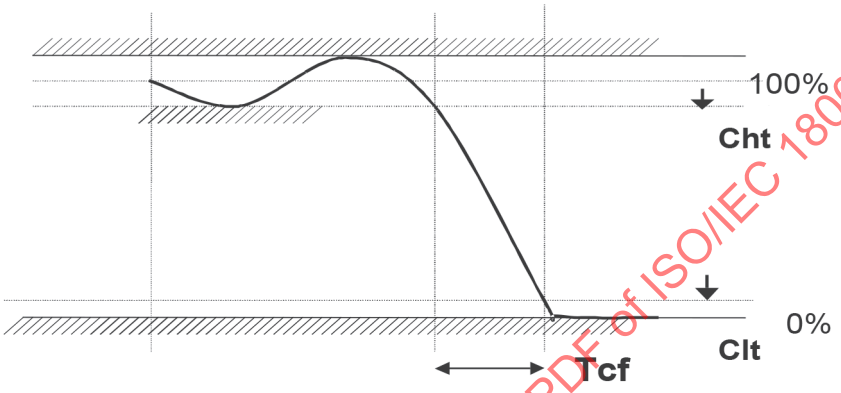


Figure 2 — Interrogator power-down waveform

Table 2 — Interrogator power-down timings

Parameter	Min	Max
Tcf	1 µs	500 µs
Cht		3 %
Clt		1 %

6.2.3 Frequency hopping carrier rise and fall times

When the interrogator operates in the frequency hopping spread spectrum (FHSS) mode, the carrier rise and fall times shall conform to the characteristics specified in Figure 3 and Table 3.

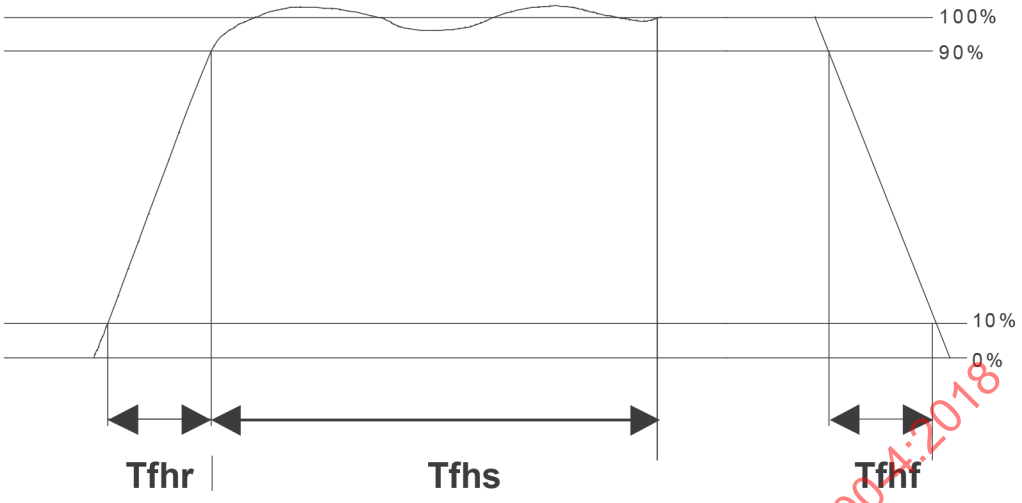


Figure 3 — FHSS carrier rise and fall characteristics

Table 3 — FHSS carrier rise and fall parameters

Parameter	Min	Max
Tfhr		15 µs
Tfhs	400 µs	
Tfhf		15 µs

NOTE The numbers in Table 3 are an example for current FCC regulations only.

6.2.4 Forward link

6.2.4.1 Carrier modulation

The data transmission from the interrogator to the tag is achieved by modulation of the carrier (ASK). The data coding is performed by generating pulses that create a Manchester coding.

The example of 40 kbit/s signal of Manchester coding as per Figure 4.

The parameter for 99 % Modulation of Manchester coding is specified in Table 4.

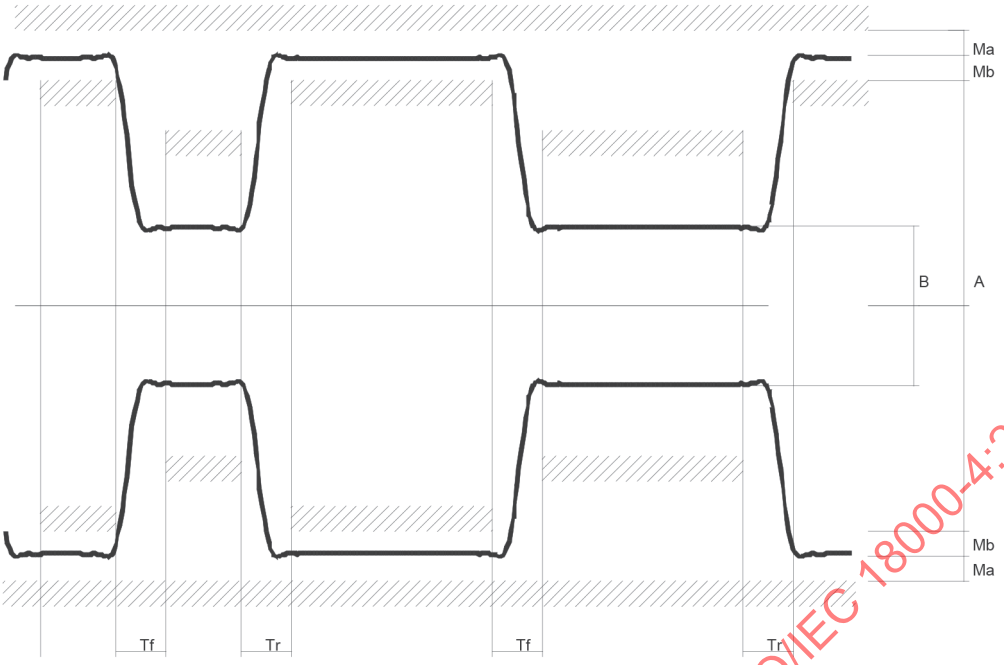


Figure 4 — Example of 40 kbit/s signal

Table 4 — Parameter for 99 % Modulation

Parameter	Minimum	Nominal	Maximum
M = (A – B)/(A + B)	90	99	100
Ma	0		0,03 (A – B)
Mb	0		0,03 (A – B)
Tr	0 µs	1,8 µs	$0,1/f_{bitrate}$
Tf	0 µs	1,8 µs	$0,1/f_{bitrate}$

6.2.4.2 Bit coding of forward link fields

Data are Manchester encoded as per Figure 5.

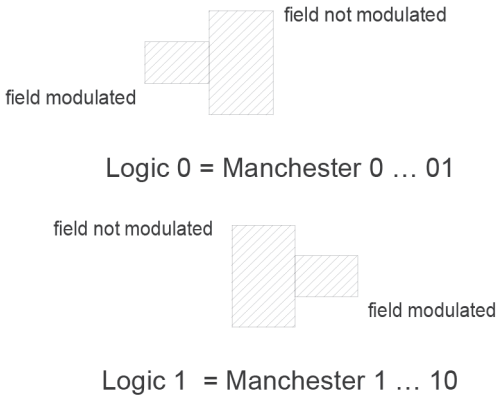


Figure 5 — Forward link bit coding

6.2.5 FM0 return link

6.2.5.1 General

The tag transmits information to the interrogator by modulating the incident energy and reflecting it back to the interrogator (backscatter).

6.2.5.2 Modulation

The tag switches its reflectivity between two states. The “space” state is the normal condition in which the tag is powered by the interrogator and is able to receive and decode the forward link. The “mark” state is the alternative condition created by changing the antenna configuration or termination.

6.2.5.3 Data rate

The return link data rate is derived from the forward link data rate and is typically 40 kbit/s. For details, see Table 5.

6.2.5.4 Data coding

Data are coded using the FM0 technique, also known as Bi-Phase Space.

One symbol period Trlb is allocated to each bit to be sent. In FM0 encoding, data transitions occur at all bit boundaries. In addition, data transitions occur at the mid-bit of logic 0 being sent.

Table 5 — Return link parameters

Data rate	Trlb	Tolerance
30 kbit/s to 40 kbit/s	25 µs to 33 µs	±15 %

Coding of data is MSB first. Figure 6 illustrates the coding for the 8 bits of ‘B1’.

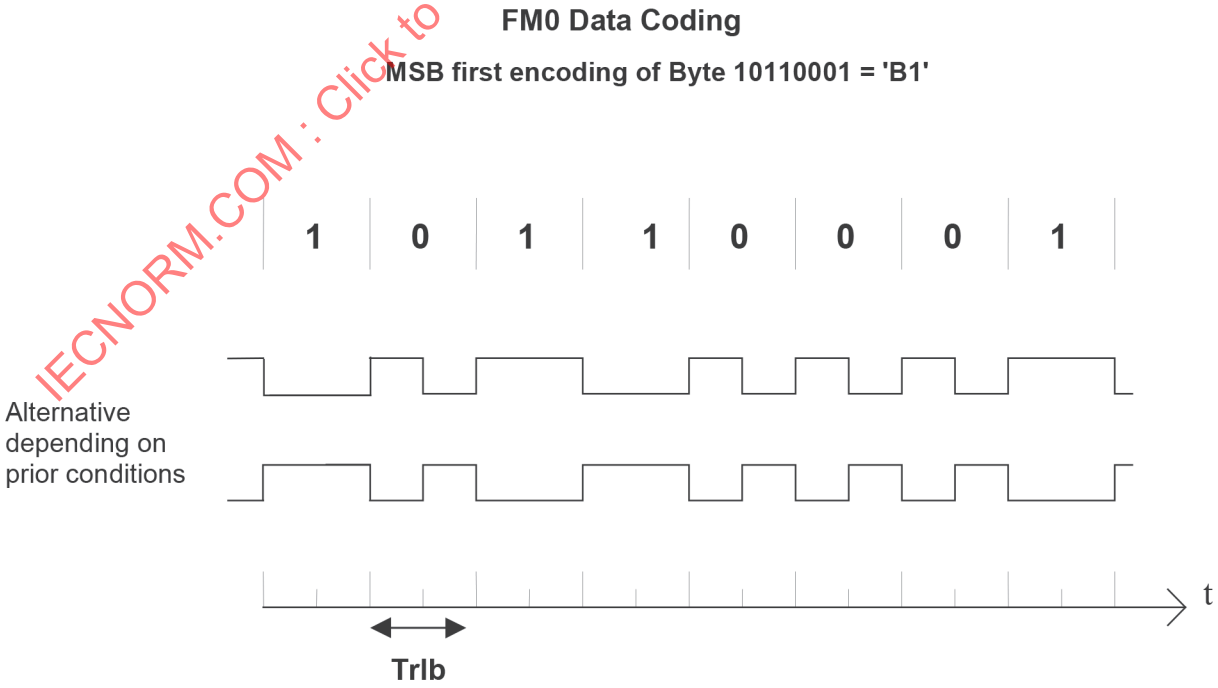


Figure 6 — Tag to interrogator data coding

6.2.5.5 Message format

A Return Link Message consists of n data bits preceded by the Preamble and followed by the tag data. The data bits are sent MSB first.

The Preamble enables the interrogator to lock to the tag data clock and begin decoding of the message. It consists of 16 bits as shown in Table 6. There are multiple code violations (sequence not conforming to FM0 rules) that act as a frame marker for the transition from Preamble to Data.

6.2.5.6 Return preamble

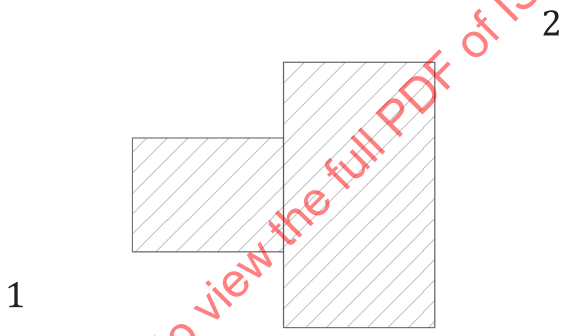
The return preamble is a sequence of backscatter modulation specified in Table 6.

Table 6 — Return preamble

00 00 01 01 01 01 01 01 01 01 00 01 10 11 00 01

Data '0' is represented by the tag's modulator being in the high impedance state, Data '1' is represented by the tag's modulator switching to the low impedance state, thereby causing a change in the incident energy to be back-scattered.

The tag shall execute backscatter, a half-bit 1 and half-bit 0 sent by the tag defined as follows in Figure 7.



NOTE 1 = low impedance (backscatter), 0 = high impedance (no backscatter).

Figure 7 — Return link preamble

6.2.6 Cyclic redundancy check (CRC)

When sending a command to the tag, the interrogator shall attach an inverted CRC to the message packet. On receiving a command from the interrogator, the tag shall verify that the checksum or the CRC value is valid. If it is invalid, it shall discard the frame, shall not respond and shall not take any other action.

The 16-bit CRC applies for both communication directions: from interrogator to tag and from tag to interrogator.

The polynomial used to calculate the CRC is $X^{16} + X^{12} + X^5 + 1$. The 16-bit register shall be preloaded with 'FFFF'. The resulting CRC value shall be inverted, attached to the end of the packet and transmitted.

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

At the tag, the incoming CRC bits are inverted and then clocked into the register. After the LSB bit is clocked into the tag, the 16-bit CRC register should contain all zeroes.

The 16-bit CRC shall be calculated on all data bits up to, but not including, the first CRC bit.

On receiving of a response from the tag, it is recommended that the interrogator verifies that the CRC value is valid. If it is invalid, appropriate remedial action is the responsibility of the interrogator designer.

The CRC 16 bits and bytes transmission rules is specified in Table 7.

Table 7 — CRC 16 bits and bytes transmission rules

MSByte		LSByte	
MSB	LSB	MSB	LSB
CRC 16 (8 bits)		CRC 16 (8 bits)	

↑ first transmitted bit of the inverted CRC

6.2.7 Protocol concept

Data are encoded and presented in slightly different ways in the constituent fields. For interrogator-to-tag communication (forward link), data are sent using an on-off key format. The radio frequency field being on corresponds to 1, while the radio frequency field being off corresponds to 0. The on-off ratio specification is defined in 6.2.4. In the case of Manchester coding, a Manchester 1 is a 1 to 0 transition, while a Manchester 0 is a 0 to 1 transition.

For tag-to-interrogator communication (return link), data are sent using backscatter techniques. This requires that the interrogator provide steady power to the tag during the return link. While the interrogator powers the tag, the tag shall change alternately the effective impedance of the tag front end and thus changing the overall radio frequency reflectivity of the tag as seen by the interrogator. During this time, the interrogator shall not modulate the carrier. During the WAIT field (when tags write data into their memory), the interrogator shall also provide steady power to the tag and shall not modulate the carrier. The transmission protocol defines the mechanism to exchange instructions and data between the interrogator and the tag, in both directions.

It is based on the concept of "interrogator talks first".

This means that any tag shall not start transmitting (modulating) unless it has received and properly decoded an instruction sent by the interrogator.

The protocol is based on an exchange of a command from the interrogator to the tag and a response from the tag(s) to the interrogator.

The conditions under which the tag sends a response are defined in 6.3.6.

Each command and each response are contained in a frame. The frame is specified in 6.2.7.

Each command consists of the following fields:

Preamble

Delimiter

Command code

Parameter fields, depending on the command

Application data fields, depending on the command

CRC

Each response consists of the following fields:

Return preamble

Application data fields

CRC

The protocol is bit-oriented. The number of bits transmitted in a frame is a multiple of eight (8), i.e. an integer number of bytes. However, the frame itself is not based on an integer number of bytes.

In all byte fields, the MSB shall be transmitted first, proceeding to the LSB. In all word (8-byte) data fields, the MSB shall be transmitted first.

The MSB shall be the byte at the specified address. The LSB shall be the byte at the specified address plus 7 (i.e. bytes are transmitted in incrementing address order).

The byte significance is relevant to data transmission and to the GROUP_SELECT and GROUP_UNSELECT greater than and less than comparisons.

The MSB of the byte mask shall correspond to the most significant data byte, the byte at the specified address.

Word (8-byte) addresses are not required to be on an 8-word boundary and may be on any byte boundary.

RFU bits and bytes shall be set to zero (0).

6.2.8 Command format

6.2.8.1 General

The command consists of the following fields:

Preamble

Delimiter

Command

Parameter and data files

CRC

The general command format is specified in Table 8.

Table 8 — General command format

Preamble detect	Preamble	Delimiter	Command	Parameter	Data	CRC

6.2.8.2 Preamble detect field

The preamble detect field consists of a steady carrier (no modulation) during a time of at least 400 µs. This corresponds to 16 bits for a communication rate of 40 kbit/s.

6.2.8.3 Preamble

The preamble is equivalent to 9 bits of Manchester 0.

010101010101010101

6.2.8.4 Delimiter

6.2.8.4.1 Start delimiter 1

In NRZ format; includes Manchester errors; spaces ignored

11 00 11 10 10 Delimiter 1

6.2.8.5 CRC

See 6.2.6 and Annex B.

6.2.9 Response format

6.2.9.1 General

The response consists of the following fields:

Quiet

Return preamble

Data fields

CRC

The general response format is specified in Table 9.

Table 9 — General response format

Quiet	Return preamble	Data	CRC

The tag shall use a backscatter technique to communicate data to the interrogator. The interrogator shall be steadily powering the tag as well as listening to the tag response throughout the tag-to-interrogator (backscatter) communication. This applies to all fields in the return link.

6.2.9.2 QUIET

The tag shall not backscatter for 16 × Tflb − 0,75 × Tflb. The duration of the quiet time is determined by the communication speed of the forward and return link.

6.2.9.3 CRC

See 6.2.6 and Annex B.

6.2.10 WAIT

During the WAIT field, the interrogator provides steady power to the tag for the duration of at least 15 ms. No on-off key data may be sent during the write operation.

When a tag receives a write command, it shall execute a write operation. (The details of the conditions under which a write will occur are described in 6.3.6.2.5.4.) If a write operation is executed, the final field in the overall field sequence shall always be WAIT.

During the WAIT field, when the tag is writing data into the EEPROM, the interrogator shall steadily power the tag. On-off key data shall not be sent during this time.

Details are shown in Figure 8 and in Figure 9.

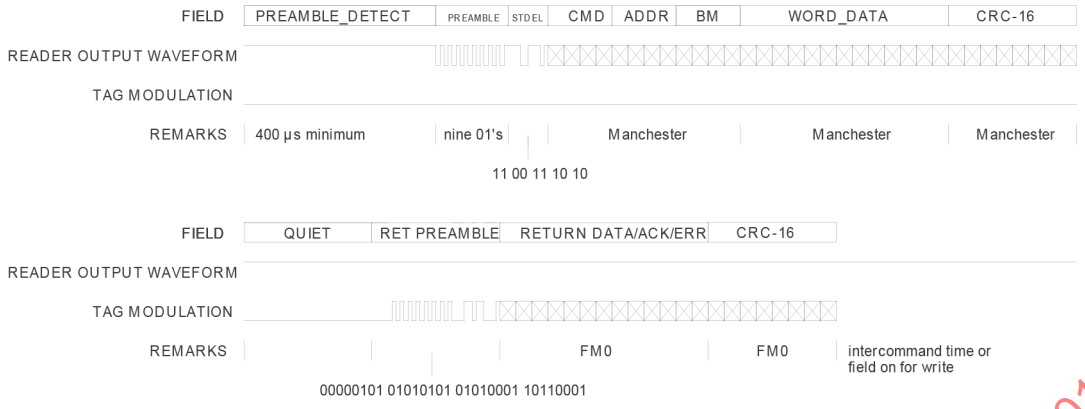


Figure 8 — Sample command/response packets for GROUP_SELECT (40 kbit/s on forward and return link)

Figure 9 — Sample command/response packets for WRITE (40 kbit/s on forward and return link)

6.2.11 Communication sequences at packet level

Figure 10 and Figure 11 show several examples of communication sequences at the packet level. Figure 10 depicts a packet sequence that includes a write command. The sequence includes a wait for write time, which provides the necessary time for the chip to complete its write operation. In addition, following the wait for write time, the interrogator issues a tag resync signal. This signal is composed of 10 consecutive 01 signals. The purpose of the tag resync signal is to initialize the tag data recovery circuitry. It is required after a write because the interrogator may output spurious edges during the wait for write time. Without the tag resync, tags may miscalibrate as a result of the spurious signals that may be generated.

Figure 11 depicts a packet sequence in which a frequency hop between commands is included. The tag resync signal is again required after the hop because spurious signals may be generated during the hop time.

In order to ensure that tags do not get confused, frequency hops between command and response should be avoided.

Action	COMMAND	RESPONSE	WAIT FOR WRITE	TAG RESYNC	COMMAND	RESPONSE
Component execution action	Interrogator	Tag	Interrogator	Interrogator	Interrogator	Tag
Notes	---	---	15 ms minimum	ten 01's	---	---

Figure 10 — Command sequence (including a write) with no hopping

Action	COMMAND	RESPONSE	HOP	TAG RESYNC	COMMAND	RESPONSE
Component execution action	Interrogator	Tag	Interrogator	Interrogator	Interrogator	Tag
Notes	---	---	< 26 µs	ten 01's	---	---

Figure 11 — Command sequence with a hop between response and next command

6.3 Protocol and collision arbitration

6.3.1 Definition of data elements, bit and byte ordering

6.3.1.1 Unique ID

See A.2.

6.3.1.2 CRC

See 6.2.6 and Annex B.

6.3.1.3 FLAGS

The tag shall support a field of eight flags. This field is called FLAGS.

The FLAGS is specified in Table 10.

Table 10 — FLAGS

Bit	Name
FLAG1 (LSB)	DE_SB (Data_Exchange Status Bit)
FLAG2	WRITE_OK
FLAG3	BATTERY_POWERED
FLAG4	BATTERY_OK
FLAG5	0 (RFU)

Table 10 *(continued)*

Bit	Name
FLAG6	0 (RFU)
FLAG7	0 (RFU)
FLAG8 (MSB)	0 (RFU)

6.3.1.3.1 Data Exchange Status Bit (DE_SB)

The tag shall set this bit when the tag goes into the DATA_EXCHANGE state and keep it set unless it moves into the POWER-OFF state.

When the DE_SB is set and the tag comes into the POWER-OFF state, then the tag shall trigger a timer that will reset the DE_SB bit after t_{DE_SB}.

t_{DE_SB} shall be at least 2 s in the temperature range 30 °C to 60 °C.

t_{DE_SB} shall be at least 4 s in the temperature range 0 °C to 50 °C.

When the tag receives the INITIALIZE command, then it shall reset the DE_SB immediately.

6.3.1.3.2 WRITE_OK

The WRITE_OK bit shall be set after a successful write access to the memory (e.g. WRITE, LOCK).

The WRITE_OK bit is cleared after execution of the command following the write command.

6.3.1.3.3 BATTERY_POWERED

The BATTERY_POWERED bit shall be set when the tag should have a battery. It shall be cleared for passive tags.

6.3.1.3.4 BATTERY_OK

The BATTERY_OK bit shall be set when the battery has enough power to support the tag. It shall be cleared for passive tags.

6.3.2 Tag memory organization

The functional memory shall be organized in blocks of one byte.

Up to 256 blocks of one byte can be addressed.

This leads to a maximum memory capacity of up to 2 kbit.

NOTE This structure allows future extensions of the maximum memory capacity.

6.3.3 Block security status

Each byte shall have a corresponding lock bit. The lock bits may be locked by use of the LOCK command. The status of the lock bit may be read by the QUERY_LOCK command. The tag shall not be allowed to reset any lock bit after leaving the final production site. In most cases, this is the production site that defines the unique ID.

6.3.4 Overall protocol description

6.3.4.1 Tag states

The tag has four major states:

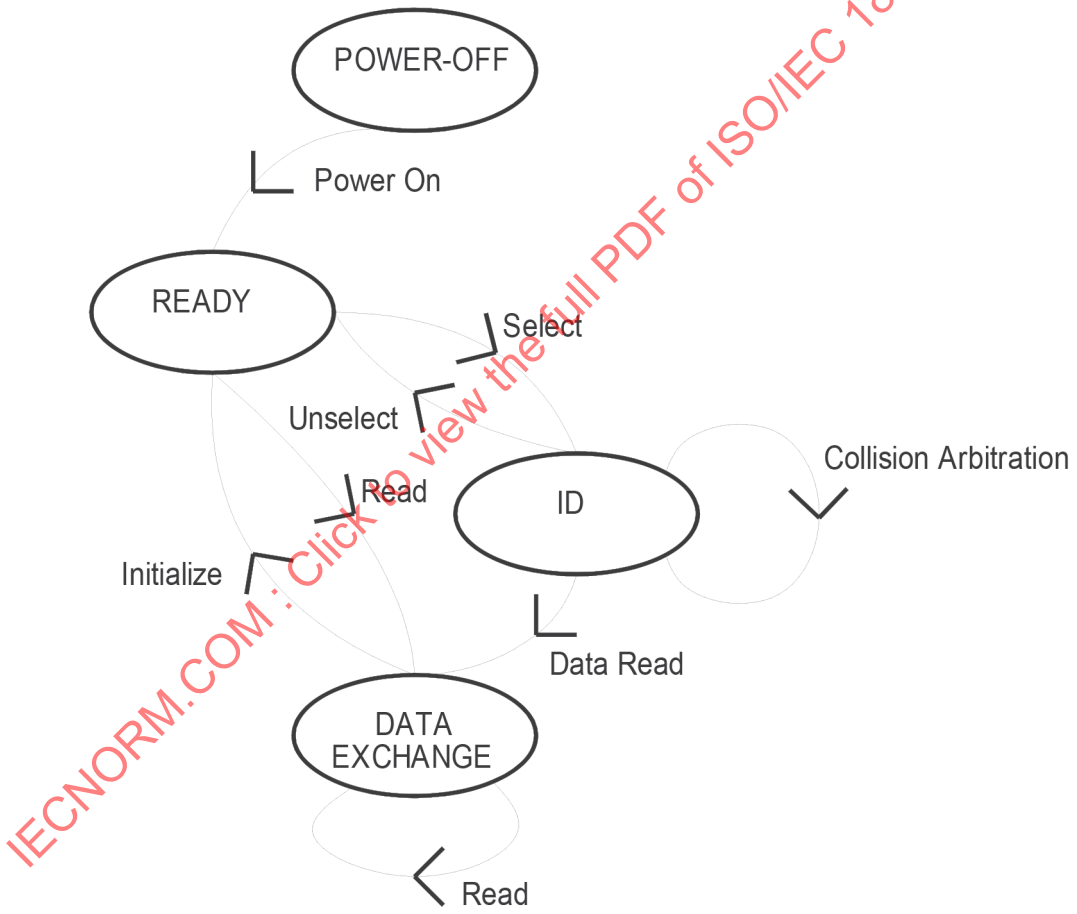
POWER-OFF The tag is in the POWER-OFF state when the interrogator cannot activate it. (For battery-assisted tags, it means that the level of RF excitation is insufficient to turn on the tag circuits.)

READY The tag is in the READY state when the interrogator first powers it up.

ID The tag is in the ID state when it is trying to identify itself to the interrogator.

DATA_EXCHANGE The tag is in the DATA_EXCHANGE state when it is known to the interrogator and was selected.

The state diagram as per Figure 12.



NOTE This diagram does not show that the tag goes into POWER-OFF from all states in the case that the interrogator field is permanently turned off.

Figure 12 — State diagram

The state diagram only shows an overview of the possible transition. Details are specified in Table 12.

Power-On:

State change when interrogator field is turned on.

Select

State change due to selection of tag by GROUP_SELECT or READ commands.

Unselect

State change due to deselection of tag by GROUP_UNSELECT commands or INITIALIZE command.

Collision Arbitration

No state change during collision arbitration until single tag is identified.

Data_Read

State change due to first read access in collision arbitration process.

Read

State change due to read access independent of collision arbitration process.

Initialize

State change due to deselection of tag by INITIALIZE command.

The detailed command processing is specified in Table 11.

The transition between these states is specified in Table 12.

6.3.4.2 Detailed command processing

Commands shall be active in states marked with "X" and neither causes a state change nor cause a response in the other states.

Table 11 — Detailed command processing

COMMAND	States		
	READY	ID	DATA EXCHANGE
GROUP_SELECT_EQ	X	X	
GROUP_SELECT_NE	X	X	
GROUP_SELECT_GT	X	X	
GROUP_SELECT_LT	X	X	
GROUP_SELECT_EQ_FLAGS	X	X	
GROUP_SELECT_NE_FLAGS	X	X	
GROUP_UNSELECT_EQ		X	
GROUP_UNSELECT_NE		X	
GROUP_UNSELECT_GT		X	
GROUP_UNSELECT_LT		X	
GROUP_UNSELECT_EQ_FLAGS		X	
GROUP_UNSELECT_NE_FLAGS		X	
MULTIPLE_UNSELECT		X	
FAIL		X	
SUCCESS		X	
RESEND		X	
INITIALIZE	X	X	X
READ	X	X	X

Table 11 *(continued)*

COMMAND	States		
	READY	ID	DATA EXCHANGE
DATA_READ		X	X
READ_VERIFY	X	X	X
READ_VERIFY4BYTE	X	X	X
WRITE	X	X	X
WRITE4BYTE	X	X	X
WRITE4BYTE_MULTIPLE		X	X
WRITE_MULTIPLE		X	X
LOCK			X
QUERY_LOCK	X	X	X

Table 12 — State transition table

Current state	Command	Condition	New state
POWER-OFF	ANY COMMAND		POWER OFF
POWER-OFF	“Power up”		READY
READY	GROUP_SELECT_EQ	≠	READY
READY	GROUP_SELECT_NE	=	READY
READY	GROUP_SELECT_GT	≤	READY
READY	GROUP_SELECT_EQ_FLAGS	flag not set	READY
READY	GROUP_SELECT_NE_FLAGS	flag set	READY
READY	GROUP_SELECT_LT	≥	READY
READY	GROUP_SELECT_EQ	=	ID
READY	GROUP_SELECT_NE	≠	ID
READY	GROUP_SELECT_GT	>	ID
READY	GROUP_SELECT_LT	<	ID
READY	GROUP_SELECT_EQ_FLAGS	flag set	ID
READY	GROUP_SELECT_NE_FLAGS	flag not set	ID
READY	INITIALIZE		READY
READY	READ	ID no match	READY
READY	READ	ID match	DATA_EXCHANGE
READY	READ_VERIFY	ID no match or not WRITE_OK	READY
READY	READ_VERIFY	ID match and WRITE_OK	DATA_EXCHANGE
READY	READ_VERIFY4BYTE	ID no match or not WRITE_OK	READY
READY	READ_VERIFY4BYTE	ID match and WRITE_OK	DATA_EXCHANGE
READY	WRITE	ID no match	READY
READY	WRITE	ID match	DATA_EXCHANGE
READY	WRITE4BYTE	ID no match	READY
READY	WRITE4BYTE	ID match	DATA_EXCHANGE
READY	QUERY_LOCK	ID no match	READY
READY	QUERY_LOCK	ID match	DATA_EXCHANGE
ID	GROUP_UNSELECT_EQ	≠	ID

Table 12 *(continued)*

Current state	Command	Condition	New state
ID	GROUP_UNSELECT_NE	=	ID
ID	GROUP_UNSELECT_GT	≤	ID
ID	GROUP_UNSELECT_LT	≥	ID
ID	GROUP_UNSELECT_EQ_FLAGS	flag not set	ID
ID	GROUP_UNSELECT_NE_FLAGS	flag set	ID
ID	GROUP_UNSELECT_EQ	=	READY
ID	GROUP_UNSELECT_NE	≠	READY
ID	GROUP_UNSELECT_GT	>	READY
ID	GROUP_UNSELECT_LT	<	READY
ID	GROUP_UNSELECT_EQ_FLAGS	flag set	READY
ID	GROUP_UNSELECT_NE_FLAGS	flag not set	READY
ID	MULTIPLE_UNSELECT	≠ or not WRITE_OK	ID
ID	MULTIPLE_UNSELECT	= and WRITE_OK	READY
ID	GROUP_SELECT_EQ		ID
ID	GROUP_SELECT_NE		ID
ID	GROUP_SELECT_GT		ID
ID	GROUP_SELECT_LT		ID
ID	GROUP_SELECT_EQ_FLAGS		ID
ID	GROUP_SELECT_NE_FLAGS		ID
ID	FAIL		ID
ID	SUCCESS		ID
ID	RESEND		ID
ID	INITIALIZE		READY
ID	READ	ID no match	ID
ID	READ	ID match	DATA_EXCHANGE
ID	DATA_READ	ID no match	ID
ID	DATA_READ	ID match	DATA_EXCHANGE
ID	READ_VERIFY	ID no match or not WRITE_OK	ID
ID	READ_VERIFY	ID match and WRITE_OK	DATA_EXCHANGE
ID	READ_VERIFY4BYTE	ID no match or not WRITE_OK	ID
ID	READ_VERIFY4BYTE	ID match and WRITE_OK	DATA_EXCHANGE
ID	WRITE	ID no match	ID
ID	WRITE	ID match	DATA_EXCHANGE
ID	WRITE4BYTE	ID no match	ID
ID	WRITE4BYTE	ID match	DATA_EXCHANGE
ID	WRITE_MULTIPLE		ID
ID	WRITE4BYTE_MULTIPLE		ID
ID	QUERY_LOCK	ID no match	ID
ID	QUERY_LOCK	ID match	DATA_EXCHANGE
DATA_EXCHANGE	INITIALIZE		READY
DATA_EXCHANGE	READ		DATA_EXCHANGE
DATA_EXCHANGE	DATA_READ		DATA_EXCHANGE

Table 12 *(continued)*

Current state	Command	Condition	New state
DATA_EXCHANGE	READ_VERIFY		DATA_EXCHANGE
DATA_EXCHANGE	READ_VERIFY4B YTE		DATA_EXCHANGE
DATA_EXCHANGE	WRITE		DATA_EXCHANGE
DATA_EXCHANGE	WRITE4BYTE		DATA_EXCHANGE
DATA_EXCHANGE	WRITE4BYTE_MULTIPLE		DATA_EXCHANGE
DATA_EXCHANGE	WRITE_MULTIPLE		DATA_EXCHANGE
DATA_EXCHANGE	LOCK		DATA_EXCHANGE
DATA_EXCHANGE	QUERY_LOCK		DATA_EXCHANGE

6.3.5 Collision arbitration

The interrogator may use the GROUP_SELECT and GROUP_UNSELECT commands to define all or a subset of tags in the field to participate in the collision arbitration. It then may use the identification commands to run the collision arbitration algorithm.

For the collision arbitration, the tag shall support two pieces of hardware on the tag:

— an 8-bit counter COUNT;

— a random 1 or 0 generator.

In the beginning, a group of tags is moved to the ID state by GROUP_SELECT commands and shall set their internal counters to 0. Subsets of the group may be unselected by GROUP_UNSELECT commands back to the READY state. Other groups can be selected before the identification process begins. Simulation results show no advantage in identifying one large group or a few smaller groups.

After the above described selection, the following loop should be performed:

a) All tags in the ID state with the counter COUNT at 0 shall transmit their ID. This set initially includes all the selected tags.

b) If more than one tag transmitted, the interrogator receives an erroneous response. The FAIL command shall be sent.

c) All tags receiving a FAIL command with COUNT not equal to 0 shall to increment COUNT. That is, they move further away from being able to transmit.

   All tags receiving FAIL a count of 0 (those that just transmitted) shall generate a random number. Those that roll a 1 shall increment COUNT and shall not transmit. Those that roll a 0 shall keep COUNT at 0 and shall send their UID again.

One of four possibilities now occurs:

d) If more than one tag transmits the FAIL command, step b) repeats (Possibility 1).

e) If all tags roll a 1, none transmits. The interrogator receives nothing. It sends the SUCCESS command. All the counters decrement, and the tags with a count of 0 transmit. Typically, this returns to step b) (Possibility 2).

f) If only one tag transmits and the ID is received correctly, the interrogator shall send the DATA_READ command with the ID. If the DATA_READ command is received correctly, that tag shall move to the DATA_EXCHANGE state and shall transmit its data.

   The interrogator shall send SUCCESS. All tags in the ID state shall decrement COUNT.

g) If only one tag has a count of 1 and transmits, step e) or f) is repeated. If more than one tag transmits, step b) is repeated (Possibility 3).

h) If only one tag transmits and the ID is received with an error, the interrogator shall send the RESEND command. If the ID is received correctly, step e) is repeated. If the ID is received again some variable number of times (this number can be set based on the level of error handling desired for the system), it is assumed that more than one tag is transmitting and step b) is repeated (Possibility 4).

6.3.6 Commands

Commands are divided into four functional groups:

— selection commands;

— identification commands;

— data transfer commands;

— multiple commands.

Further, commands have one of the following types:

— mandatory;

— optional;

— custom;

— proprietary.

6.3.6.1 Command types

All tags with the same IC manufacturer code and the same IC version number shall behave the same.

6.3.6.1.1 Mandatory

The command codes range from '00' to '0A', '0C', '15', '1E', '1F' and '20' to '3F'.

A mandatory command shall be supported by all tags that claim to be compliant. Interrogators which claim compliance shall support all mandatory commands.

6.3.6.1.2 Optional

The command codes range from '0B', '0D' to '0F', '11' to '13', '17' to '1C', '1D' and from '40' to '9F'.

Optional commands are commands that are specified within the International Standard. Interrogators shall be technically capable of performing all optional commands that are specified in the International Standard (although need not be set up to do so). Tags may or may not support optional commands. If an optional command is used, it shall be implemented in the manner specified in the International Standard.

If the tag does not support an optional command, it shall remain silent.

NOTE The commands whose code ranges from '17' to '1C' are optional and not essential to operate the tag. However, their support by the tag is recommended for appropriate performance. To reflect this, they are reported as "recommended" in Table 13.

6.3.6.1.3 Custom

The command codes range from 'A0' to 'DF'.

Custom commands may be enabled by an International Standard, but they shall not be specified in that International Standard. A custom command shall not solely duplicate the functionality of any mandatory or optional command defined in the International Standard by a different method.

The only fields that can be customized are the parameters and the data fields.

Any custom command contains as its first parameter the IC manufacturer code. This allows IC manufacturers to implement custom commands without risking duplication of command codes and thus misinterpretation.

If the tag does not support a custom command, it shall remain silent.

6.3.6.1.4 Proprietary

The command codes are '10', '14', '16' and they range from 'E0' to 'FF'.

Proprietary commands may be enabled by an International Standard, but they shall not be specified in that International Standard. A proprietary command shall not solely duplicate the functionality of any mandatory or optional command defined in the International Standard by a different method.

These commands are used by IC and tag manufacturers for various purposes such as tests, programming of system information, etc. The IC manufacturer may at its option document them or not. It is allowed that these commands are disabled after IC and/or tag manufacturing.

6.3.6.2 Command codes and format

Table 13 — Command codes and format

Command code	Type	Command name	Parameters		
'00'	Mandatory	GROUP_SELECT_EQ	ADDRESS	BYTE_MASK	WORD_DATA
'01'	Mandatory	GROUP_SELECT_NE	ADDRESS	BYTE_MASK	WORD_DATA
'02'	Mandatory	GROUP_SELECT_GT	ADDRESS	BYTE_MASK	WORD_DATA
'03'	Mandatory	GROUP_SELECT_LT	ADDRESS	BYTE_MASK	WORD_DATA
'04'	Mandatory	GROUP_UNSELECT_EQ	ADDRESS	BYTE_MASK	WORD_DATA
'05'	Mandatory	GROUP_UNSELECT_NE	ADDRESS	BYTE_MASK	WORD_DATA
'06'	Mandatory	GROUP_UNSELECT_GT	ADDRESS	BYTE_MASK	WORD_DATA
'07'	Mandatory	GROUP_UNSELECT_LT	ADDRESS	BYTE_MASK	WORD_DATA
'08'	Mandatory	FAIL	None	None	None
'09'	Mandatory	SUCCESS	None	None	None
'0A'	Mandatory	INITIALIZE	None	None	None
'0B'	Optional	DATA_READ	ID	ADDRESS	None
'0C'	Mandatory	READ	ID	ADDRESS	None
'0D'	Recommended	WRITE	ID	ADDRESS	BYTE_DATA
'0E'	Recommended	WRITE_MULTIPLE	None	ADDRESS	BYTE_DATA
'0F'	Recommended	LOCK	ID	ADDRESS	None
'10'	Proprietary	IC manufacturer dependent			
'11'	Recommended	QUERY_LOCK	ID	ADDRESS	None
'12'	Recommended	READ_VERIFY	ID	ADDRESS	None
'13'	Recommended	MULTIPLE_UNSELECT	ADDRESS	BYTE_DATA	None
'14'	Proprietary	IC manufacturer dependent			
'15'	Mandatory	RESEND	None	None	None
'16'	Proprietary	IC manufacturer dependent			

Table 13 *(continued)*

Command code	Type	Command name	Parameters		
'17'	Recommended	GROUP_SELECT_EQ_ FLAGS	None	BYTE_MASK	BYTE_DATA
'18'	Recommended	GROUP_SELECT_NE_ FLAGS	None	BYTE_MASK	BYTE_DATA
'19'	Recommended	GROUP_UNSELECT_EQ_ FLAGS	None	BYTE_MASK	BYTE_DATA
'1A'	Recommended	GROUP_UNSELECT_NE_ FLAGS	None	BYTE_MASK	BYTE_DATA
'1B'	Recommended	WRITE4BYTE	ID	ADDRESS	BYTE_MASK
'1C'	Recommended	WRITE4BYTE_MULTIPLE	BYTE_ MASK	ADDRESS	4BYTE_DATA
'1D'	Recommended	READ_VERIFY4BYTE	ID	ADDRESS	
'1E' to '1F'	Mandatory	RFU			
'20' to '3F'	Mandatory	RFU			
'40' to '9F'	Optional	RFU			
'A0' to 'DF'	Custom	IC manufacturer dependent			
'E0' to 'FF'	Proprietary	IC manufacturer dependent			

6.3.6.2.1 Command fields

The command fields are specified in Table 14.

Table 14 — Command fields

Field name	Field size
COMMAND	1 byte
ADDRESS	1 byte
BYTE_MASK	1 byte
ID	8 bytes
WORD_DATA	8 bytes
BYTE_DATA	1 byte
4BYTE_DATA	4 bytes

6.3.6.2.2 Tag responses

The tag responses are specified in Table 15.

Table 15 — Tag responses

Response code	Response name	Response size
'00'	ACKNOWLEDGE	1 byte
	ACKNOWLEDGE_NOK	1 byte
'01'	ACKNOWLEDGE_OK	1 byte
'FE'	ERROR_NOK	1 byte
'FF'	ERROR	1 byte
	ERROR_OK	1 byte

Table 15 *(continued)*

Response code	Response name	Response size
Not applicable	WORD_DATA	8 bytes
Not applicable	BYTE_DATA	1 byte
'02' to 'FD'	RFU	

6.3.6.2.3 Selection commands

Selection commands define a subset of tags in the field to be identified or written to and may be used as part of the collision arbitration.

6.3.6.2.3.1 Data comparison for selection command on memory

Each select command of the commands GROUP_SELECT_EQ, GROUP_SELECT_NE, GROUP_SELECT_GT, GROUP_SELECT_LT, GROUP_UNSELECT_EQ, GROUP_UNSELECT_NE, GROUP_UNSELECT_GT, GROUP_UNSELECT_LT has three arguments (parameter and data):

ADDRESS

BYTE_MASK

WORD_DATA

and the tag shall make one of four possible comparisons:

EQ M EQUAL D

NE M NOT EQUAL D

GT M GREATER THAN D

LT M LOWER THAN D

The arguments of the comparison are

M7 MSB	M6	M5	M4	M3	M2	M1	M0 LSB
Tag memory content at ADDRESS+0	Tag memory content at ADDRESS+1	Tag memory content at ADDRESS+2	Tag memory content at ADDRESS+3	Tag memory content at ADDRESS+4	Tag memory content at ADDRESS+5	Tag memory content at ADDRESS+6	Tag memory content at ADDRESS+7

$$M = M0 + M1 \times 2^{8} + M2 \times 2^{16} + M3 \times 2^{24} + M4 \times 2^{32} + M5 \times 2^{40} + M6 \times 2^{48} + M7 \times 2^{56}$$

and the argument of the command

D7 MSB	D6	D5	D4	D3	D2	D1	D0 LSB
First byte after command							Last byte after command

$$D = D0 + D1 \times 2^{8} + D2 \times 2^{16} + D3 \times 2^{24} + D4 \times 2^{32} + D5 \times 2^{40} + D6 \times 2^{48} + D7 \times 2^{56}$$

The argument BYTE_MASK defines what bytes to be considered for comparison.

The data masking for Group Select and Group Unselect commands are specified in Table 16.

Table 16 — Data masking for Group_Select and Group_Unselect commands

BYTE_MASK	WORD_DATA
Bit 7 (MSB) is set	Consider D7 and M7 for comparison
Bit 6 is set	Consider D6 and M6 for comparison
Bit 5 is set	Consider D5 and M5 for comparison
Bit 4 is set	Consider D4 and M4 for comparison
Bit 3 is set	Consider D3 and M3 for comparison
Bit 2 is set	Consider D2 and M2 for comparison
Bit 1 is set	Consider D1 and M1 for comparison
Bit 0 (LSB) is set	Consider D0 and M0 for comparison
Bit 7 (MSB) is cleared	Ignore D7 and M7 for comparison
Bit 6 is cleared	Ignore D6 and M6 for comparison
Bit 5 is cleared	Ignore D5 and M5 for comparison
Bit 4 is cleared	Ignore D4 and M4 for comparison
Bit 3 is cleared	Ignore D3 and M3 for comparison
Bit 2 is cleared	Ignore D2 and M2 for comparison
Bit 1 is cleared	Ignore D1 and M1 for comparison
Bit 0 (LSB) is cleared	Ignore D0 and M0 for comparison

6.3.6.2.3.2 Data comparison for selection command on flags

Each select command of the commands GROUP_SELECT_EQ_FLAGS, GROUP_SELECT_NE_FLAGS, GROUP_UNSELECT_EQ_FLAGS, GROUP_UNSELECT_NE_FLAGS has two arguments (parameter and data):

BYTE_MASK

BYTE_DATA

and the tag shall make of two possible comparisons:

EQ FLAGS EQUAL D

NE FLAGS NOT EQUAL D

The arguments of the comparison are FLAGS, as defined in 6.3.1.3, and the argument of the command D, consisting of the bits D7 (MSB) to D0 (LSB).

The argument BYTE_MASK defines what bits to be considered for comparison.

The data masking for Group_Select_Flags and Group_Unselect_Flags are specified in Table 17.

Table 17 — Data masking for Group_Select_Flags and Group_Unselect_Flags

BYTE_MASK	BYTE_DATA
Bit 7 (MSB) is set	Consider D7 and FLAG7 for comparison
Bit 6 is set	Consider D6 and FLAG6 for comparison
Bit 5 is set	Consider D5 and FLAG5 for comparison
Bit 4 is set	Consider D4 and FLAG4 for comparison
Bit 3 is set	Consider D3 and FLAG3 for comparison
Bit 2 is set	Consider D2 and FLAG2 for comparison
Bit 1 is set	Consider D1 and FLAG1 for comparison
Bit 0 (LSB) is set	Consider D0 and FLAG0 for comparison

Table 17 *(continued)*

BYTE_MASK	BYTE_DATA
Bit 7 (MSB) is cleared	Ignore D7 and FLAG7 for comparison
Bit 6 is cleared	Ignore D6 and FLAG6 for comparison
Bit 5 is cleared	Ignore D5 and FLAG5 for comparison
Bit 4 is cleared	Ignore D4 and FLAG4 for comparison
Bit 3 is cleared	Ignore D3 and FLAG3 for comparison
Bit 2 is cleared	Ignore D2 and FLAG2 for comparison
Bit 1 is cleared	Ignore D1 and FLAG1 for comparison
Bit 0 (LSB) is cleared	Ignore D0 and FLAG0 for comparison

Formula describing the EQUAL function:

The EQUAL comparison passes if (!B7+(D7=FLAG7)) * (!B6+(D6=FLAG6)) * (!B5+(D5=FLAG5)) * * (!B4+(D4=FLAG4)) * (!B3+(D3=FLAG3)) * (!B2+(D2=FLAG2)) * (!B1+(D1=FLAG1)) * (!B0+(D0=FLAG0)) is true.

Formula describing the UNEQUAL function:

The UNEQUAL comparison passes if B7*(D7!=FLAG7) + B6*(D6!=FLAG6) + B5*(D5!=FLAG5) + B4*(D4!=FLAG4) + B3*(D3!=FLAG3) + B2*(D2!=FLAG2) + B1*(D1!=FLAG1) + B0*(D0!=FLAG0) is true.

6.3.6.2.3.3 GROUP_SELECT_EQ

Command code = '00'

On receiving a GROUP_SELECT_EQ command, a tag which is in the READY state shall read the 8-byte memory content beginning at the specified address and compare it with the WORD_DATA sent by the interrogator. In the case that the memory content is equal to the WORD_DATA, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and go into the state ID.

On receiving a GROUP_SELECT_EQ command, a tag which is in the ID state shall set its internal counter COUNT to 0, read its UID and send back the UID, and stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_SELECT_EQ command is specified in Table 18.

The GROUP_SELECT_EQ response in the case of NO error is specified in Table 19.

Table 18 — GROUP_SELECT_EQ command

Preamble	Delimiter	COMMAND	ADDRESS	MASK	WORD_DATA	CRC
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 19 — GROUP_SELECT_EQ response in the case of NO error

Preamble	ID	CRC
	64 bits	16 bits

NOTE If the byte mask is zero, GROUP_SELECT_EQ selects all tags.

6.3.6.2.3.4 GROUP_SELECT_NE

Command code = '01'

On receiving a GROUP_SELECT_NE command, a tag which is in the READY state shall read the 8-byte memory content beginning at the specified address and compare it with the WORD_DATA sent by the interrogator. In the case that the memory content is not equal to the WORD_DATA, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and go into the state ID.

On receiving a GROUP_SELECT_NE command, a tag which is in the ID state shall set its internal counter COUNT to 0, read its UID and send back the UID, and stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_SELECT_NE command is specified in Table 20.

The GROUP_SELECT_NE response in the case of NO error is specified in Table 21.

Table 20 — GROUP_SELECT_NE command

Preamble	Delimiter	COMMAND	ADDRESS	BYTE_MASK	WORD_DATA	CRC
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 21 — GROUP_SELECT_NE response in the case of NO error

Preamble	ID	CRC
	64 bits	16 bits

6.3.6.2.3.5 GROUP_SELECT_GT

Command code = '02'

On receiving a GROUP_SELECT_GT command, a tag which is in the READY state shall read the 8-byte memory content beginning at the specified address and compare it with the WORD_DATA sent by the interrogator. In the case that the memory content is greater than the WORD_DATA, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and go into the state ID.

On receiving a GROUP_SELECT_GT command, a tag which is in the ID state shall set its internal counter COUNT to 0, read its UID and send back the UID, and stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_SELECT_GT command is specified in Table 22.

The GROUP_SELECT_GT response in the case of NO error is specified in Table 23.

Table 22 — GROUP_SELECT_GT command

Preamble	Delimiter	COMMAND	ADDRESS	MASK	WORD_DATA	CRC
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 23 — GROUP_SELECT_GT response in the case of NO error

Preamble	ID	CRC
	64 bits	16 bits

6.3.6.2.3.6 GROUP_SELECT_LT

Command code = '03'

On receiving a GROUP_SELECT_LT command, a tag which is in the READY state shall read the 8-byte memory content beginning at the specified address and compare it with the WORD_DATA sent by the interrogator. In the case that the memory content is lower than the WORD_DATA, the tag shall set its

internal counter COUNT to 0, read its UID and send back the UID and go into the state ID, and stay in the ID state.

On receiving a GROUP_SELECT_LT command, a tag which is in the ID state shall set its internal counter COUNT to 0, read its UID and send back the UID, and stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_SELECT_LT command is specified in Table 24.

The GROUP_SELECT_LT response is specified in Table 25.

Table 24 — GROUP_SELECT_LT command

Preamble	Delimiter	COMMAND	ADDRESS	BYTE_MASK	WORD_DATA	CRC
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 25 — GROUP_SELECT_LT response

Preamble	ID	CRC
	64 bits	16 bits

6.3.6.2.3.7 GROUP_UNSELECT_EQ

Command code = '04'

On receiving a GROUP_UNSELECT_EQ command, a tag which is in the ID state shall read the 8-byte memory content beginning at the specified address and compare it with the WORD_DATA sent by the interrogator. In the case that the memory content is equal to the WORD_DATA, the tag shall go into the state READY and not send any reply. In the case that the comparison fails, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and shall stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_UNSELECT_EQ command is specified in Table 26.

The GROUP_UNSELECT_EQ response is specified in Table 27.

Table 26 — GROUP_UNSELECT_EQ command

Preamble	Delimiter	COMMAND	ADDRESS	BYTE_MASK	WORD_DATA	CRC
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 27 — GROUP_UNSELECT_EQ response

Preamble	ID	CRC
	64 bits	16 bits

NOTE If the byte mask is zero, GROUP_UNSELECT_EQ unselects all tags.

6.3.6.2.3.8 GROUP_UNSELECT_NE

Command code = '05'

On receiving a GROUP_UNSELECT_NE command, a tag which is in the ID state shall read the 8-byte memory content beginning at the specified address and compare it with the WORD_DATA sent by the interrogator. In the case that the memory content is not equal to the WORD_DATA, the tag shall go into the state READY and not send any reply. In the case the comparison fails, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and shall stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_UNSELECT_NE command is specified in Table 28.

The GROUP_UNSELECT_NE response is specified in Table 29.

Table 28 — GROUP_UNSELECT_NE command

Preamble	Delimiter	COMMAND	ADDRESS	BYTE_MASK	WORD_DATA	CRC
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 29 — GROUP_UNSELECT_NE response

Preamble	ID	CRC
	64 bits	16 bits

6.3.6.2.3.9 GROUP_UNSELECT_GT

Command code = '06'

On receiving a GROUP_UNSELECT_GT command, a tag which is in the ID state shall read the 8-byte memory content beginning at the specified address and compare it with the WORD_DATA sent by the interrogator. In the case that the memory content is greater than the WORD_DATA, the tag shall go into the state READY and not send any reply. In the case that the comparison fails, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and shall stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_UNSELECT_GT response is specified in Table 30.

The GROUP_UNSELECT_GT response in the case of NO error and comparison fails is specified in Table 31.

Table 30 — GROUP_UNSELECT_GT command

Preamble	Delimiter	COMMAND	ADDRESS	BYTE_MASK	WORD_DATA	CRC
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 31 — GROUP_UNSELECT_GT response in the case of NO error and comparison fails

Preamble	ID	CRC
	64 bits	16 bits

6.3.6.2.3.10 GROUP_UNSELECT_LT

Command code = '07'

On receiving a GROUP_UNSELECT_LT command, a tag which is in the ID state shall read the 8-byte memory content beginning at the specified address and compare it with the WORD_DATA sent by the interrogator. In the case that the memory content is lower than the WORD_DATA, the tag shall go into the state READY and not send any reply. In the case that the comparison fails, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and shall stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_UNSELECT_LT command is specified in Table 32.

The GROUP_UNSELECT_LT response is specified in Table 33.

Table 32 — GROUP_UNSELECT_LT command

Preamble	Delimiter	COMMAND	ADDRESS	BYTE_MASK	WORD_DATA	CRC
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 33 — GROUP_UNSELECT_LT response

Preamble	ID	CRC
	64 bits	16 bits

6.3.6.2.3.11 MULTIPLE_UNSELECT

Command code = '13'

On receiving a MULTIPLE_UNSELECT command, a tag which is in the ID state shall read the 1-byte memory content beginning at the specified address and compare it with the BYTE_DATA sent by the interrogator. In the case that the memory content is equal to the BYTE_DATA and the flag WRITE_OK is set, then the tag shall go into the state READY and not send any reply. In the case that the comparison fails, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and shall stay in the ID state.

In all other cases, the tag shall not send a reply.

The MULTIPLE_UNSELECT command is specified in Table 34.

The MULTIPLE_UNSELECT response is specified in Table 35.

Table 34 — MULTIPLE_UNSELECT command

Preamble	Delimiter	COMMAND	ADDRESS	BYTE_DATA	CRC
		8 bits	8 bits	8 bits	16 bits

Table 35 — MULTIPLE_UNSELECT response

Preamble	ID	CRC
	64 bits	16 bits

This command may be used to unselect all tags that had a successful write, while tags that had a weak write or write problems stay selected.

6.3.6.2.3.12 GROUP_SELECT_EQ_FLAGS

Command code = '17'

On receiving a GROUP_SELECT_EQ_FLAGS command, a tag which is in the READY state shall compare the FLAGS with the BYTE_DATA sent by the interrogator. In the case that the FLAGS are equal to the BYTE_DATA, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and go into the state ID.

On receiving a GROUP_SELECT_EQ_FLAGS command, a tag which is in the ID state shall set its internal counter COUNT to 0, read its UID and send back the UID, and stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_SELECT_EQ_FLAGS command is specified in Table 36.

The GROUP_SELECT_EQ_FLAGS response is specified in Table 37.

Table 36 — GROUP_SELECT_EQ_FLAGS command

Preamble	Delimiter	COMMAND	BYTE_MASK	BYTE_DATA	CRC
		8 bits	8 bits	8 bits	16 bits

Table 37 — GROUP_SELECT_EQ_FLAGS response

Preamble	ID	CRC
	64 bits	16 bits

NOTE If the byte mask is zero, GROUP_SELECT_EQ_FLAGS selects all tags.

6.3.6.2.3.13 GROUP_SELECT_NE_FLAGS

Command code = '18'

On receiving a GROUP_SELECT_NE_FLAGS command, a tag which is in the READY state shall compare the FLAGS with the BYTE_DATA sent by the interrogator. In the case that the FLAGS are not equal to the BYTE_DATA, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and go into the state ID.

On receiving a GROUP_SELECT_NE_FLAGS command, a tag which is in the ID state shall set its internal counter COUNT to 0, read its UID and send back the UID, and stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_SELECT_NE_FLAGS command is specified in Table 38.

The GROUP_SELECT_NE_FLAGS response is specified in Table 39.

Table 38 — GROUP_SELECT_NE_FLAGS command

Preamble	Delimiter	COMMAND	BYTE_MASK	BYTE_DATA	CRC
		8 bits	8 bits	8 bits	16 bits

Table 39 — GROUP_SELECT_NE_FLAGS response

Preamble	ID	CRC
	64 bits	16 bits

6.3.6.2.3.14 GROUP_UNSELECT_EQ_FLAGS

Command code = '19'

On receiving a GROUP_UNSELECT_EQ_FLAGS command, a tag which is in the ID state shall compare the FLAGS with the BYTE_DATA sent by the interrogator. In the case that the FLAGS are equal to the BYTE_DATA, the tag shall go into the state READY and not send any reply. In the case that the comparison fails, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and shall stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_UNSELECT_EQ_FLAGS command is specified in Table 40.

The GROUP_UNSELECT_EQ_FLAGS response is specified in Table 41.

Table 40 — GROUP_UNSELECT_EQ_FLAGS command

Preamble	Delimiter	COMMAND	BYTE_MASK	BYTE_DATA	CRC
		8 bits	8 bits	8 bits	16 bits

Table 41 — GROUP_UNSELECT_EQ_FLAGS response

Preamble	ID	CRC
	64 bits	16 bits

NOTE If the byte mask is zero, GROUP_UNSELECT_EQ_FLAGS unselects all tags.

6.3.6.2.3.15 GROUP_UNSELECT_NE_FLAGS

Command code = '1A'

On receiving a GROUP_UNSELECT_NE_FLAGS command, a tag which is in the ID state shall compare the FLAGS with the BYTE_DATA sent by the interrogator. In the case that the FLAGS are not equal to the BYTE_DATA, the tag shall go into the state READY and not send any reply. In the case that the comparison fails, the tag shall set its internal counter COUNT to 0, read its UID and send back the UID, and shall stay in the ID state.

In all other cases, the tag shall not send a reply.

The GROUP_UNSELECT_NE_FLAGS command is specified in Table 42.

The GROUP_UNSELECT_NE_FLAGS response is specified in Table 43.

Table 42 — GROUP_UNSELECT_NE_FLAGS command

Preamble	Delimiter	COMMAND	BYTE_MASK	BYTE_DATA	CRC
		8 bits	8 bits	8 bits	16 bits

Table 43 — GROUP_UNSELECT_NE_FLAGS response

Preamble	ID	CRC
	64 bits	16 bits

6.3.6.2.4 Identification commands

Identification commands are used to perform to run the multiple tag identification protocol.

6.3.6.2.4.1 FAIL

Command code = '08'

The identification algorithm uses FAIL when more than one tag tried to identify itself at the same time. Some tags back off and some tags retransmit.

A tag shall only accept a FAIL command if it is in the ID state. In the case that its internal counter COUNT is not zero or the random generator result is 1, then COUNT shall be increased by 1, unless it is FF.

If the resulting COUNT value is 0, then the tag shall read its UID and send it back in the response.

The FAIL command is specified in Table 44.

The FAIL response is specified in Table 45.

Table 44 — FAIL command

Preamble	Delimiter	COMMAND	CRC
		8 bits	16 bits

Table 45 — FAIL response

Preamble	ID	CRC
	64 bits	16 bits

6.3.6.2.4.2 SUCCESS

Command code = '09'

SUCCESS initiates identification of the next set of tags. It is used in two cases:

— When all tags receiving FAIL backed off and did not transmit, SUCCESS causes those same tags to transmit again.

— After, for example, a DATA_READ moves an identified tag to DATA_EXCHANGE, SUCCESS causes the next subset of selected but unidentified tags to transmit.

A tag shall only accept a SUCCESS command if it is in the ID state. In the case that its internal counter COUNT is not zero, it shall be decreased by 1.

If the resulting COUNT value is 0, then the tag shall read its UID and send it back in the response.

The SUCCESS command is specified in Table 46.

The SUCCESS response is specified in Table 47.

Table 46 — SUCCESS command

Preamble	Delimiter	COMMAND	CRC
		8 bits	16 bits

Table 47 — SUCCESS response

Preamble	ID	CRC
	64 bits	16 bits

6.3.6.2.4.3 RESEND

Command code = '15'

The identification algorithm uses RESEND when only one tag transmitted but the UID was received in error. The tag that transmitted resends its UID.

A tag shall only accept a RESEND command if it is in the ID state. If the COUNT value is 0, then the tag shall read its UID and send it back in the response.

The RESEND command is specified in Table 48.

The RESEND response is specified in Table 49.

Table 48 — RESEND command

Preamble	Delimiter	COMMAND	CRC
		8 bits	16 bits

Table 49 — RESEND response

Preamble	ID	CRC
	64 bits	16 bits

6.3.6.2.4.4 INITIALIZE

Command code = '0A'

On receiving an INITIALIZE command, a tag shall go into the READY state and reset the Data_Exchange_Status_Bit.

It shall not send any response.

The INITIALIZE command is specified in Table 50.

Table 50 — INITIALIZE command

Preamble	Delimiter	COMMAND	CRC
		8 bits	16 bits

6.3.6.2.5 Data Transfer commands

Data Transfer commands are used to read or write data from or to the tag memory.

For memory lock functionality, a tag shall provide the opportunity to mark an address lockable. An ADDRESS is marked lockable by commands as described in this clause. No ADDRESS shall be marked lockable after the tag has been in the POWER-OFF state. A tag shall not support more than one ADDRESS to be marked lockable at the same time.

6.3.6.2.5.1 READ

Command code = '0C'

On receiving the READ command, the tag shall compare the sent ID with its UID. In the case that the ID is equal to the UID, the tag shall from any state move to the DATA_EXCHANGE state, read the 8-byte memory content beginning at the specified address and send back its content in the response. In the case that the ID is not equal to the UID or any other error, the tag shall not send a reply. Further, the tag makes the byte of ADDRESS lockable.

The address is numbered from '00' to 'FF' (0 to 255).

The Read command is specified in Table 51.

The Read response is specified in Table 52.

Table 51 — Read command

Preamble	Delimiter	COMMAND	ID	ADDRESS	CRC
		8 bits	64 bits	8 bits	16 bits

Table 52 — Read response

Preamble	WORD_DATA	CRC
	64 bits	16 bits

6.3.6.2.5.2 DATA_READ

Command code = '0B'

On receiving the DATA_READ command, the tag shall only if it is in either the state ID or the state DATA_EXCHANGE compare the sent ID with its UID. In the case that the ID is equal to the UID, the tag shall from any state except READY move to the DATA_EXCHANGE state, read the 8-byte memory content beginning at the specified address and send back its content in the response. In the case that the ID is not equal to the UID or any other error, the tag shall not send a reply. The tag also shall send no reply when it is in the state READY, independent of the value of ID. Further, the tag makes the byte of ADDRESS lockable.

The address is numbered from '00' to 'FF' (0 to 255).

The DATA_READ command is specified in Table 53.

The DATA_READ response is specified in Table 54.

Table 53 — DATA_READ command

Preamble	Delimiter	COMMAND	ID	ADDRESS	CRC
		8 bits	64 bits	8 bits	16 bits

Table 54 — DATA_READ response

Preamble	WORD_DATA	CRC
	64 bits	16 bits

6.3.6.2.5.3 READ_VERIFY

Command code = '12'

On receiving the READ_VERIFY command, the tag shall compare the sent ID with its UID. In the case that the ID is equal to the UID and the WRITE_OK flag is set, the tag shall from any state move to the DATA_EXCHANGE state, read the 1-byte memory content beginning at the specified address and send back its content in the response. Further, the tag shall mark the byte at ADDRESS lockable. In the case that the ID is not equal to the UID, WRITE_OK is not set, or any other error, the tag shall not send a reply. Further, the tag makes the byte of ADDRESS lockable.

The address is numbered from '00' to 'FF' (0 to 255).

The READ_VERIFY command is specified in Table 55.

The READ_VERIFY response is specified in Table 56.

Table 55 — READ_VERIFY command

Preamble	Delimiter	COMMAND	ID	ADDRESS	CRC
		8 bits	64 bits	8 bits	16 bits

Table 56 — READ_VERIFY response

Preamble	BYTE_DATA	CRC
	8 bits	16 bits

6.3.6.2.5.4 READ_VERIFY4BYTE

Command code = '1D'

On receiving the READ_VERIFY4BYTE command, the tag shall compare the sent ID with its UID. In the case that the ID is equal to the UID and the WRITE_OK flag is set, the tag shall from any state move to the DATA_EXCHANGE state, read the 4-byte memory content beginning at the specified address and send back its content in the response.

In the case that the ID is not equal to the UID, WRITE_OK is not set, or any other error, the tag shall not send a reply.

The address is numbered from '00' to 'FF' (0 to 255).

The READ_VERIFY4BYTE command is specified in Table 57.

The READ_VERIFY4BYTE response is specified in Table 58.

Table 57 — READ_VERIFY4BYTE command

Preamble	Delimiter	COMMAND	ID	ADDRESS	CRC-16
		8 bits	64 bits	8 bits	16 bits

Table 58 — READ_VERIFY4BYTE response

Preamble	4BYTE_DATA	CRC-16
	32 bits	16 bits

6.3.6.2.5.5 WRITE

Command code = '0D'

On receiving the WRITE command, the tag shall compare the sent ID with its UID. In the case that the ID is equal to the UID, the tag shall from any state move to the DATA_EXCHANGE state and read the lock information for the byte on the specified memory content beginning at the specified address. In the case that the memory is locked, it shall send back the ERROR response. In the case that the memory is unlocked, it shall send back the ACKNOWLEDGE and program the data into the specified memory address. Further, the tag makes the byte of ADDRESS lockable.

In all other cases, the tag will not send any reply.

In the case that the write access was successful, the tag shall set the WRITE_OK bit. Otherwise, it shall reset it.

The address is numbered from '00' to 'FF' (0 to 255).

The WRITE command is specified in Table 59.

The WRITE response in the case of unlocked memory is specified in Table 60.

The WRITE response in the case of locked memory is specified in Table 61.

Table 59 — WRITE command

Preamble	Delimiter	COMMAND	ID	ADDRESS	BYTE_DATA	CRC
		8 bits	64 bits	8 bits	8 bits	16 bits

Table 60 — WRITE response in the case of unlocked memory

Preamble	ACKNOWLEDGE	CRC
	8 bits	16 bits

Table 61 — WRITE response in the case of locked memory

Preamble	ERROR	CRC
	8 bits	16 bits

6.3.6.2.5.6 WRITE4BYTE

Command code = '1B'

On receiving the WRITE4BYTE command, the tag shall compare the sent ID with its UID. In the case that the ID is equal to the UID, the tag shall from any state move to the DATA_EXCHANGE state and read the lock information for the 4 bytes on the specified memory content beginning at the specified address. In the case that one of the bytes specified by the BYTE_MASK is locked, it shall send back the ERROR response. In the case that all bytes are unlocked, it shall send back the ACKNOWLEDGE and program the data into the specified memory.

In all other cases, the tag will not send any reply.

Executing WRITE4BYTE, a tag shall only write those bytes that are selected by the BYTE_MASK, which means that write could be done to 1 to 4 bytes (using the mask bits in the BYTE_MASK field).

BYTE_MASK of the command	
ADDRESS	Bit of BYTE_MASK to select whether byte should be written
[ADDR+0]	B7
[ADDR+1]	B6
[ADDR+2]	B5
[ADDR+3]	B4

In the case that the write access was successful, the tag shall set the WRITE_OK bit. Otherwise, it shall reset it.

The address is numbered from '00' to 'FF' (0 to 255). The starting address for the WRITE4BYTE command shall be on a 4-byte page boundary.

The WRITE4BYTE command is specified in Table 62.

The WRITE4BYTE response in the case of unlocked memory is specified in Table 63.

The WRITE4BYTE response in the case of locked memory is specified in Table 64.

Table 62 — WRITE4BYTE command

Preamble	Delimiter	COMMAND	ID	ADDRESS	BYTE_MASK	4BYTEDATA	CRC
		8 bits	64 bits	8 bits	8 bits	32 bits	16 bits

Table 63 — WRITE4BYTE response in the case of unlocked memory

Preamble	ACKNOWLEDGE	CRC
	8 bits	16 bits

Table 64 — WRITE4BYTE response in the case of locked memory

Preamble	ERROR	CRC
	8 bits	16 bits

6.3.6.2.5.7 LOCK

Command code = '0F'

On receiving a LOCK command, a tag which is in the DATA_EXCHANGE state shall read its UID and compare it with the ID sent by the interrogator. In the case that the UID is equal to the ID, the ADDRESS is within the valid address range and the byte at ADDRESS is marked lockable, then the tag shall send back the ACKNOWLEDGE and program the lock bit of the specified memory address.

In all other cases, the tag shall not send a reply.

In the case that the write access was successful, the tag shall set the WRITE_OK bit. Otherwise, it shall reset it.

The address is numbered from '00' to 'FF' (0 to 255).

The LOCK command is specified in Table 65.

The LOCK response in the case that locking was possible and performed is specified in Table 66.

Table 65 — LOCK command

Preamble	Delimiter	COMMAND	ID	ADDRESS	CRC
		8 bits	64 bits	8 bits	16 bits

Table 66 — LOCK response in the case that locking was possible and performed

Preamble	ACKNOWLEDGE	CRC
	8 bits	16 bits

6.3.6.2.5.8 QUERY_LOCK

Command code = '11'

On receiving a QUERY_LOCK command, a tag shall read its UID and compare it with the ID sent by the interrogator. In the case that the UID is equal to the ID and the ADDRESS is within the valid address range, then the tag shall move into the DATA_EXCHANGE state. Further, the tag shall read the lock bit for the memory byte at ADDRESS. In the case that this memory is not locked, then it shall response ACKNOWLEDGE_OK if WRITE_OK is set and ACKNOWLEDGE_NOK if WRITE_OK is cleared. In the case that this memory is locked, then it shall respond ERROR_OK if WRITE_OK is set and ERROR_NOK if WRITE_OK is cleared. Further, the tag shall mark the byte of ADDRESS lockable unless it is already locked.

In all other cases, the tag shall not send a reply.

The address is numbered from '00' to 'FF' (0 to 255).

The QUERY_LOCK command is specified in Table 67.

That QUERY_LOCK response if memory address is not locked and WRITE_OK is set is specified in Table 68.

That QUERY_LOCK response if memory address is not locked and WRITE_OK is cleared is specified in Table 69.

That QUERY_LOCK response if memory address is locked and WRITE_OK is set is specified in **Table 70**.

That QUERY_LOCK response if memory address is locked and WRITE_OK is cleared is specified in Table 71.

Table 67 — QUERY_LOCK command

Preamble	Delimiter	COMMAND	ID	ADDRESS	CRC
		8 bits	64 bits	8 bits	16 bits

Table 68 — QUERY_LOCK response if memory address is not locked and WRITE_OK is set

Preamble	ACKNOWLEDGE_OK	CRC
	8 bits	16 bits

Table 69 — QUERY_LOCK response if memory address is not locked and WRITE_OK is cleared

Preamble	ACKNOWLEDGE_NOK	CRC
	8 bits	16 bits

Table 70 — QUERY_LOCK response if memory address is locked and WRITE_OK is set

Preamble	ERROR_OK	CRC
	8 bits	16 bits

Table 71 — QUERY_LOCK response if memory address is locked and WRITE_OK is cleared

Preamble	ERROR_NOK	CRC
	8 bits	16 bits

6.3.6.2.5.9 WRITE_MULTIPLE

Command code = '0E'

Write Multiple commands are used to write to and to verify multiple tags in parallel.

On receiving the WRITE_MULTIPLE command, a tag which is in the ID state or the DATA_EXCHANGE state shall read the lock information for the byte on the specified memory content beginning at the specified address. In the case that the memory is locked, it shall do nothing. In the case that it is unlocked, it shall program the data into the specified memory.

The tag shall not send any response.

In the case that the write access was successful, the tag shall set the WRITE_OK bit. Otherwise, it shall reset it.

The address is numbered from '00' to 'FF' (0 to 255).

The WRITE_MULTIPLE command is specified in Table 72.

Table 72 — WRITE_MULTIPLE command

Preamble	Delimiter	COMMAND	ADDRESS	BYTE_DATA	CRC
		8 bits	8 bits	8 bits	16 bits

6.3.6.2.5.10 WRITE4BYTE_MULTIPLE

Command code = '1C'

Write Multiple commands are used to write to and to verify multiple tags in parallel.

On receiving the WRITE4BYTE_MULTIPLE command, a tag which is in the ID state or the DATA_EXCHANGE state shall read the lock information for the 4 bytes on the specified memory content beginning at the specified address. In the case that one of the bytes of the 4-byte block is locked, it shall do nothing. In the case that all bytes are unlocked, it shall program the data into the specified memory.

The tag shall not send any response.

Executing WRITE4BYTE, a tag shall only write those bytes that are selected by the BYTE_MASK, which means that write could be done to 1 to 4 bytes (using the mask bits in the BYTE_MASK field).

BYTE_MASK of the command WRITE4BYTE_MULTIPLE	
ADDRESS	Bit of BYTE_MASK to select whether byte should be written
[ADDR+0]	B7
[ADDR+1]	B6
[ADDR+2]	B5
[ADDR+3]	B4

In the case that the write access was successful, the tag shall set the WRITE_OK bit. Otherwise, it shall reset it.

The address is numbered from '00' to 'FF' (0 to 255). The starting address for the WRITE4BYTE command shall be on a 4-byte page boundary.

The WRITE4BYTE_MULTIPLE command is specified in Table 73.

Table 73 — WRITE4BYTE_MULTIPLE command

Preamble	Delimiter	COMMAND	ADDRESS	BYTE_MASK	4BYTEDATA	CRC
		8 bits	8 bits	8 bits	32 bits	16 bits

6.3.6.2.6 Response description (Binary Tree Protocol Type)

6.3.6.2.6.1 ACKNOWLEDGE

ACKNOWLEDGE indicates a successful acceptance of the WRITE or LOCK.

6.3.6.2.6.2 ERROR

ERROR indicates an error in the WRITE, e.g. a write to locked memory area.

6.3.6.2.6.3 ACKNOWLEDGE_OK

ACKNOWLEDGE_OK is the response to a QUERY_LOCK and indicates an unlocked memory byte and a successful preceding write command.

6.3.6.2.6.4 ACKNOWLEDGE_NOK

ACKNOWLEDGE_NOK is the response to a QUERY_LOCK and indicates an unlocked memory byte and an unsuccessful preceding write command.

6.3.6.2.6.5 ERROR_OK

ERROR_OK is the response to a QUERY_LOCK and indicates a locked memory byte and a successful preceding write command.

6.3.6.2.6.6 ERROR_NOK

ERROR_NOK is the response to a QUERY_LOCK and indicates as locked memory byte and an unsuccessful preceding write command.

6.3.6.2.6.7 WORD_DATA

WORD_DATA is 8 bytes returned in response to a READ or DATA_READ command.

6.3.6.2.6.8 ID

ID is 8 bytes returned in response to a GROUP_SELECT, GROUP_UNSELECT, FAIL, SUCCESS or RESEND,

6.3.6.2.6.9 BYTE_DATA

BYTE_DATA is 1 byte returned in response to the READ_VERIFY command.

6.3.6.2.6.10 4BYTEDATA

4BYTEDATA is 4 bytes used as argument for commands WRITE4BYTE, WRITE4BYTE_MULTIPLE and READ_VERIFY4BYTE.

6.3.7 Transmission errors

There are two types of transmission errors: modulation coding errors (detectable per bit) and CRC errors (detectable per command). Both errors cause any command to be aborted. The tag shall not respond.

For all CRC errors, the tag returns to the ready state.

For all coding errors, the tag returns to the READY state if a valid start delimiter had been detected. Otherwise, it maintains its current state.

7 MODE 2: Long range high data rate RFID system

7.1 MODE 2: General

This clause describes an RFID system offering a gross data rate up to 384 kbit/s at the air interface in case of Read/Write (R/W) tag. In case of Read Only (R/O) tag, the data rate is 76,8 kbit/s. By using of battery powered tags, such a system is well designed for long-range RFID applications. This air interface description does not explicitly claim for battery assistance in the tag.

7.2 Modulation and coding

General: To avoid transmissions errors in case of data word 0 (data bits 0 have also CRC bits 0) on the air interface, every transmitted data word from the interrogator shall be multiplied byte by byte with B9hex. This operation is done by a simple XOR. To minimize the hardware in the tag and to get the original data out of the tag, the same operation shall also be used in the interrogator after receiving data from a tag.

The Multiplication word is specified in Table 74.

Table 74 — Multiplication word

Multiplication word (8 bits)							
B7	B6	B5	B4	B3	B2	B1	B0
1	0	1	1	1	0	0	1

At tag manufacturing, every data stored in the tag shall be pre-processed by that operation (this applies also for R/O-tags).

7.2.1 Forward link (only for R/W-tag)

The forward link modulation and coding format can be seen in Figure 13. Two carriers, a CW carrier and a GMSK modulated carrier, shall be transmitted at the same time to the tag. This minimizes the

hardware on the tag because the local oscillator for the down converter on the tag is generated in the interrogator.

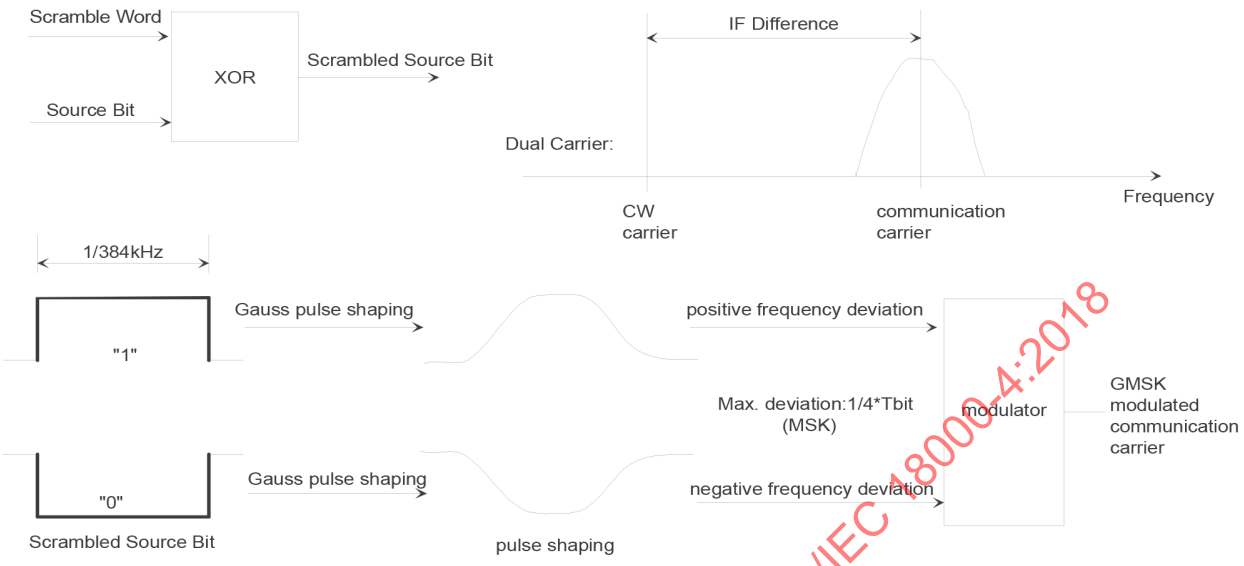


Figure 13 — Forward link modulation and coding

7.2.2 Return link for notification (for both types of the tag)

The return link modulation and coding format during the notification can be seen in Figure 14.

In case of R/O-tag, an OOK modulation instead of BPSK modulation may also be used. Even in the case of OOK, the whole pre-processing like inverting, subcarrier modulation and differential encoding shall be applied.

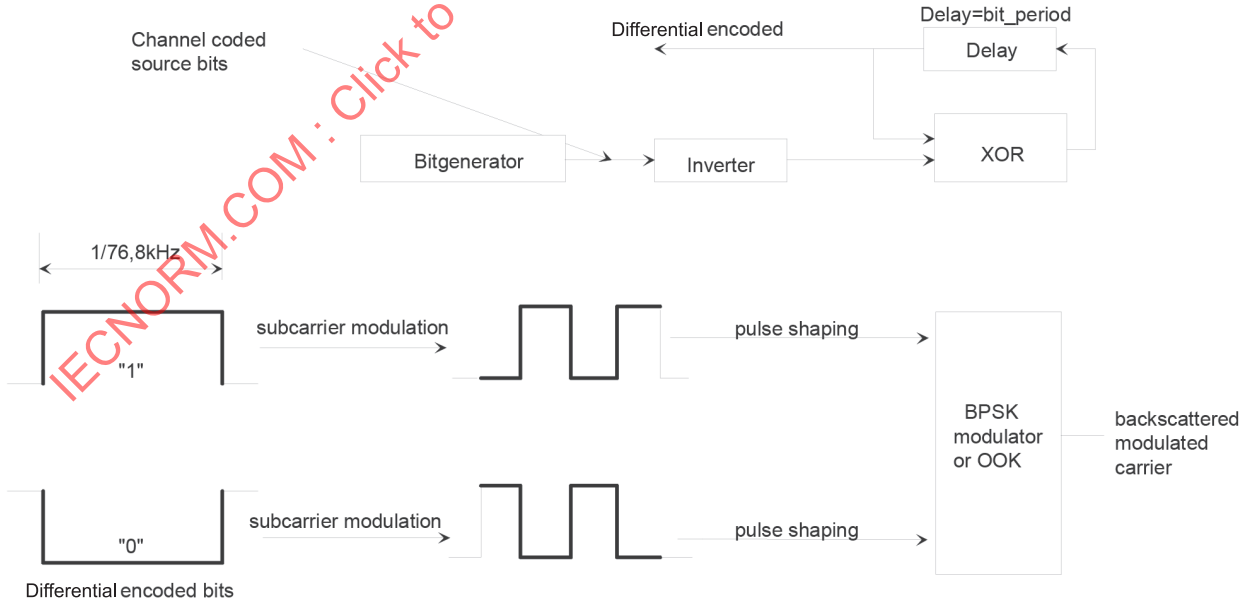


Figure 14 — Return link modulation and coding during notification

7.2.3 Return link for communication (only for R/W-tag)

The return link modulation and coding format during the communication can be seen in Figure 15.

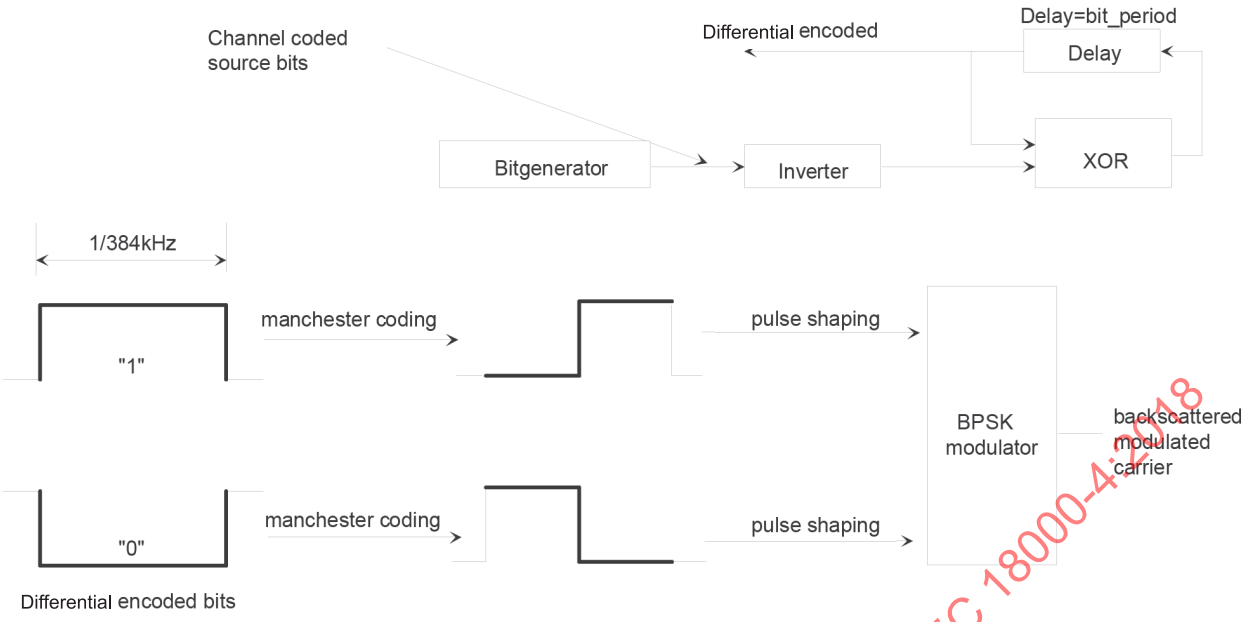


Figure 15 — Return link modulation and coding during communication

7.3 General system description

The system shall consist of an interrogator and at least one of three types of tags:

— an R/W-tag having read capability from and write capability to the tag;

— an R/O-tag behaving the same as an R/W-tag but only read capability from the tag;

— a special version of the R/O-tag with a short notification channel N-CH being useful in high-speed applications.

Mixed operation with different types of tags at the same time shall be possible. The interrogator shall operate at least with normal R/O-tags.

To be able to operate all tag types, a TTF concept shall be used. Therefore, all tags shall backscatter a fixed sequence (notification sequence) starting with synchronization information and the tag data (including UserTagID, MfrTagID and MemoryID) to establish a communication. Depending on the content of the synchronization information, the interrogator shall evaluate the tag type. The total length of the sequence may be fixed or is depending on the MemoryID.

The repetition time may be set over the air interface individually according to application parameters like tag speed, tag identification rate or total amount of tags in the field of the interrogator. Anticollision is done by randomizing the repetition time. Individual tags shall backscatter their sequence at an average repetition time but randomized (i.e. duty cycle of tag wake-up procedure and sleep time are random).

For the forward link of R/W tags, two carriers are necessary: one modulated by GMSK (BT = 0,5) and the other carrier is CW. For the return link, the tag uses backscatter modulation of the communication carrier. Thus, the system uses two carriers of constant frequency difference. In case of R/O-tag, there is only one carrier necessary. However, while the difference remains fixed, the two carriers may hop in the allowed frequency band as the communication takes place to reduce the impact of in-band interference on system performance. The hopping is governed by power measurements in free frequency channels between communication periods.

Both the interrogator and the tag operate during small time intervals only when idle, the former to avoid interference for neighbouring RFID systems and the latter to reduce power consumption. In these short

overlapping wake-up periods, the interrogator listens if a tag modulates the transmitted reference carrier with a notification sequence. If this is the case, the interrogator initiates the notification procedure by synchronizing to the individual tag's signal. After the notification procedure is completed and the interrogator accepts the tag, the system enters the communication mode to perform exchange of data. To do so, and to enable or sustain communication with additional tags in the identification field, the interrogator is forced to synchronize to an arriving tag before the end of the notification mode. After completing the notification, the interrogator turns back to the communication mode and continues the existing transmission between the interrogator and tags. This allows the interrogator to organize all R/W-tags in the field in a time frame structure for subsequent service. All tag information from an R/O-tag shall be completely read out during the notification process. No additional communication is necessary. Generally, the communication between interrogator and tag is based on time division duplexing/time division multiplexing (TDD/TDM). The interrogator time-multiplexes communications between an interrogator and several tags (TDM). The information exchange between interrogator and tag is based on time division multiplexing (TDD). Hence, data transmission is performed in time slots. Up to 64 subframes, each consisting of 14 time slots, are combined to a frame. The number of subframes actually used is fixed at system installation, but may be changed on maintenance occasions. During communication between tag and interrogator, a subframe is assigned permanently and exclusively to a tag.

7.4 Frame structure

7.4.1 Hierarchical structure

The Protocol is frame-based. Each frame contains 1 to 64 subframes and each subframe contains 14 slots (see Figure 16). Each slot can be used for transmitting either 200 bits at a data rate of 384 kbit/s or 40 bits at a data rate of 76,8 kbit/s. The length of a subframe is consequently (1/384 kHz) × 200 × 14 = 7,29 ms.

The length of a frame is SW-configurable and ranges from 1 subframe (approx. 7,3 ms) to 64 subframes (approx. 466,6 ms). This configuration is made by installation. It is possible to reconfigure this structure, but a dynamical reconfiguration during the operation is not possible. In the case of a forward link, there is no guard time between the slots. In the case of a return link, protection bits are inserted between the slots. The number of protection bits depends on the physical channel type.

The communication between interrogator and tag is based on time division duplexing/time division multiplexing (TDD/TDM). The interrogator time-multiplexes the communication between an interrogator and several tags. The information exchange between interrogator and tag is based on time division multiplexing. While the communication is going on, a subframe is assigned permanently to a tag. It is not possible to assign a subframe to more than one tag.

The same frame structure is being used for communication and spectrum check channels. For a notification channel, the frame structure is adapted to that of the service requesting tag, for as long as it takes until the notification channel is terminated. Once the notification channel is terminated, the original frame structure of the interrogator shall be re-established.

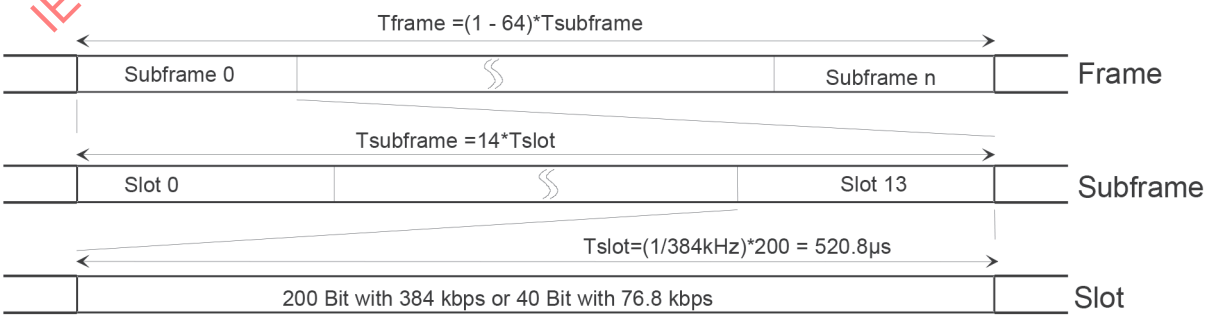


Figure 16 — Frame structure

7.4.2 Logical channels

Definition: Logical channels are the assignments between subframes and the tasks to be performed and are controlled by the interrogator. There are three main groups of logical channels:

— notification channel (N-CH);

— communication channel (C-CH);

— spectrum check channel (SC-CH).

Logical channels contain physical channels, which are explained in the following clauses. It is possible to chain logical channels, enabling the transmission of a super ordinate access to several frames at once (e.g. write 2kBytes). Such a chain shall always start with a notification channel and end with a communication channel (in the case of R/W-tag). The last logical channel has to transmit an End of Communication (EOC) signal on the physical level.

In case of R/O-tag, the whole information transmission from tag to interrogator shall take place in the notification channel. There is no communication channel to be built up. A spectrum check channel may or may not be used.

7.4.2.1 Notification channel (for both types of tags): N-CH

Function: On the notification channel, a new tag is inserted into the interrogator slot structure to carry out the bi-directional communication if it is an R/W-tag (see Figure 17), or all the information shall be read out from the tag if it is an R/O-tag (see Figure 18 and Figure 19). The notification channel shall be started at least in slot0 if a tag slot structure is detected. If neither communication nor spectrum check channel is used, the notification channel can be started in every slot. The notification channel is terminated when the tag reads the first command in case of R/W-tag, or after all information is read out from an R/O-tag. The first command is transmitted in the subframe assigned to and reserved for the communication channel.

Notification shall be cancelled if:

— The retrieved TagID is blacklisted (e.g. the interrogator does not want to communicate with the tag).

— The retrieved TagID contains errors (non-correctable CRC errors). The information on the logical confirmation channel contains errors (non-correctable CRC errors).

— The first command is not read and interpreted correctly by an R/W-tag (non-correctable CRC errors). In that case, the interrogator does not get a response from this tag.

— All the subframes are fully assigned (no command shall be transmitted).

7.4.2.2 Communication channel (only for R/W-tag): C-CH

Function: The communication channel is the medium where the read and write access operations between interrogator and tag are carried out (see Figure 20). Once the connection is set up, it is also possible to query the TagID. This channel shall be started after the tag reads the first command and shall be terminated when the interrogator sets the EOC (End of Communication) signal.

7.4.2.3 Spectrum check channel: SC-CH

Function: The spectrum check channel shall be used for searching free frequency channels (see Figure 19 and Figure 20). This channel may be activated if no notification or communication channel is being operated.

7.4.2.4 Priorities between the various logical channels

a) The first command is transmitted on the communication channel (same as termination of a notification).

Priorities when the first command is transmitted on the communication channels are specified in Table 75.

Table 75 — Priorities when the first command is transmitted on the communication channels

Channel	Priority
Notification	2
Communication	1
Spectrum check	3

This implies that a started notification shall be terminated before a new one can be started.

b) The first command is not transmitted on the communication channel.

Priorities when the first command is not transmitted on the communication channels are specified in Table 76.

Table 76 — Priorities when the first command is not transmitted on the communication channels

Channel	Priority
Notification	1
Communication	2
Spectrum check	3

This means that a notification shall interrupt the processing of the other two channels, unless it is the first command that is being transmitted on the communication channel [see a) above]. In that case, an ARQ shall be initiated for the interrupted communication channel. Spectrum checks can be carried out only in an empty subframe.

7.4.2.5 Frame structure for the notification channel in case of an R/W-tag

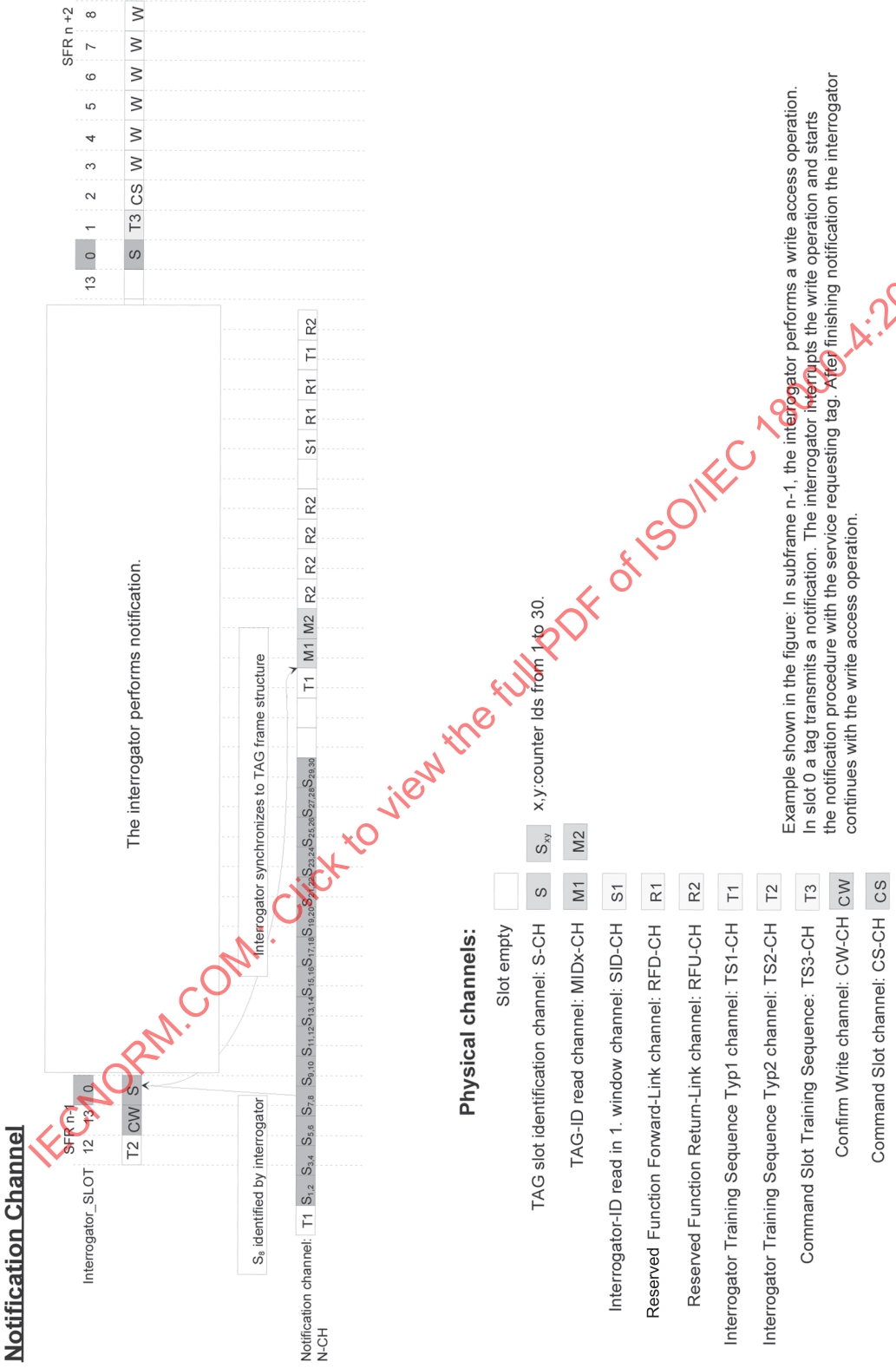


Figure 17 — Frame structure of the notification channel in case of an R/W-tag

7.4.2.6 Frame structure for the notification channel in case of an R/O-tag

Figure 18 — Frame structure of the notification channel in case of an R/O-tag

7.4.2.7 Frame structure for the notification channel in case of an R/O-tag for high speed applications

Figure 19 — Frame structure of the notification channel in case of an R/O-tag for high speed applications

7.4.2.8 Frame structure for the communication and spectrum check channels

Figure 20 — Frame structure of the communication and spectrum check channels

7.4.3 Physical channels

Definition: Physical channels are the assignments between slots and modulation and coding procedures. As mentioned before, the logical channels can contain physical channels. Table 77 shows the structure of the logical and the physical channels.

Table 77 — Logical and physical channels

Logical channel groups	Logical channel	Physical channel	Function
Notification channel N-CH	Tag Slot identification channel: S-CH (return link)	Tag Slot identification channel: S-CH (return link)	Tag sends in this channel synchronization information that could be read by interrogator during search slot (e.g. slot0).
	TagID read channel: MID-CH (return link)	TagID read channel part 1: MID1-CH (return link)	This channel is used to transmit the first part of the 32-bit TagID.
		TagID read channel part 2: MID2-CH (return link)	This channel is used to transmit the second part of the 32-bit TagID.
		TagID read channel part 1: MID3-CH (return link)	This channel is used to transmit the first part of the 32-bit TagID.
		TagID read channel part 2: MID4-CH (return link)	This channel is used to transmit the second part of the 32-bit TagID.
	TagData read channel: MIN-CH (return link)	TagData read channel: MIN1-CH (return link)	This channel is used to transmit the first part of the 32-bit tag data.
		TagData read channel: MIN2-CH (return link)	This channel is used to transmit the second part of the 32-bit tag data.
		TagData read channel: MIN3-CH (return link)	This channel is used to transmit the first part of the second 32-bit tag data.
		TagData read channel: MIN4-CH (return link)	This channel is used to transmit the second part of the second 32-bit tag data.
	Interrogator-ID read channel: SID-CH (forward link)	Interrogator-ID read channel: SID-CH (forward link)	This channel is used to transmit the 10-bit long interrogator-ID and a 15-bit counter value to the tag.
	Reserved Function Forward link channel: RFD-CH (forward link)	Reserved Function Forward link channel: RFD-CH (forward link)	Reserved for proprietary future use.
	Reserved Function Return link channel: RFU-CH (return link)	Reserved Function Return link channel: RFU-CH (return link)	Reserved for proprietary future use.
		Interrogator Training Sequence Type 1 channel: TS1-CH (return link)	This channel is used only to ease the implementation of the hardware.

Table 77 *(continued)*

Logical channel groups	Logical channel	Physical channel	Function
Communication channel C-CH	Command Slot channel: CS-CH (forward link)	Command Slot channel: CS-CH (forward link)	This channel is used to transmit the following commands from the interrogator to the tag: — write; — long read; — short read; — init; — wait; — EOC.
	Read More than 84 Byte channel: RM-CH (return link)	Read channel: R-CH (return link)	On this channel, up to 108 bytes can be transmitted in a single subframe from tag to interrogator.
	Read Less or equal than 84 byte channel: RL-CH (return link)	Read channel: R-CH (return link)	On this channel, a maximum of 84 bytes can be transmitted in a single subframe from tag to interrogator.
	Write channel: W-CH (forward link)	Write channel: W-CH (forward link)	On this channel, a maximum of 144 bytes can be transmitted in a single subframe from interrogator to tag.
	Confirm Write channel: CW-CH (return link)	Confirm Write channel: CW-CH (return link)	Signals whether the transmission of data from interrogator to tag in a given subframe was free of errors or not.
		Interrogator Training Sequence Type 2 channel: TS2-CH (return link)	This channel is used only to ease the implementation of the hardware.
		Command Slot Training Sequence: TS3-CH (forward link)	This channel is used only to ease the implementation of the hardware.
Spectrum check channel SC-CH	Spectrum check channel SC-CH	Spectrum check channel SC-CH	The interrogator uses this channel to measure the RSSI values in the allowed frequency band within the allowed channels.

With regard to slot assignment, the bits are assigned as follows: MSB to LSB: from left to right. MSB is transmitted first, then LSB.

7.4.3.1 Tag Slot identification channel: S-CH (return link)

Function: During a slot identification channel (at minimum in slot0 of the interrogator), an interrogator shall read the synchronization information of a new tag (if there is a new tag in the field). The S-CH is structured in such a way that the information required for synchronization (time offset and time counters) is present twice in one slot. For each S-CH, there are two blocks which each take half a slot long. The two blocks differ only by one increment of the time counter (the first block corresponds to the lower value). This ensures that this information shall be available for evaluation for any time position between tag and interrogator slot structure. For R/W-tags and for standard R/O-tags, the time counter runs from 1 to 30, which means that 15 S-CH are sent consecutively on the N-CH. This ensures that the information required for synchronization shall be available in two subsequent slots0. In case of R/O-tags for high speed applications, the counter values of the two subsequent half slots are 31 followed

by 0. For this type of applications, the S-CH is not fixed to slot0. For a graphical representation, refer to Figure 21. Additionally, the correlator word is the same sequence for an R/W-tag and for R/O-tags but inverted. Due to that fact, the interrogator shall decide during the S-CH which type of tag starts communication.

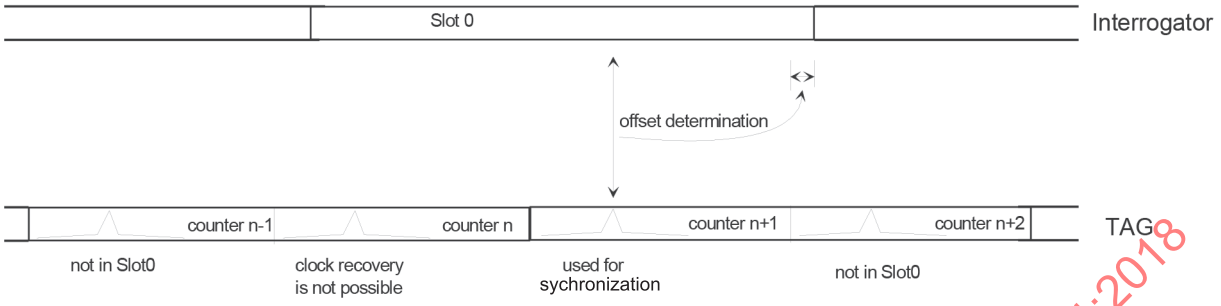


Figure 21 — S-CH position with regard to slot0

Data transmission: return link with 76,8 kbit/s.

The subframe assignment for S-CH in case of standard applications is specified in Table 78.

The subframe assignment for S-CH in case of R/O applications or if no communication or spectrum check channel is established is specified in Table 79.

Table 78 — Subframe assignment for S-CH in case of standard applications

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
•													

Table 79 — Subframe assignment for S-CH in case of R/O applications or if no communication or spectrum check channel is established

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
•	•	•	•	•	•	•	•	•	•	•	•	•	•

Slot assignment for R/W-tag:

Inverted time-reversed Barker sequence with a length of 13, which already contains the subsequence 0101 (B19...B16) for clock recovery. A parity bit is added after the 5 bit time counters.

The Slot assignment for S-CH in case of R/W-tag is specified in Table 80.

Table 80 — Slot assignment for S-CH in case of R/W-tag

Correlator													Time counters	Parity	Tail
Clock recovery															
B19	B18	B17	B16	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6...B2	B1	B0
0	1	0	1	0	0	1	1	0	0	0	0	0			0

B6...B2: Time counters from 1 to 30.

Slot assignment for R/O-tag:

Time-reversed Barker sequence with a length of 13, which already contains the subsequence 0101 for clock recovery (B18...B15). A parity bit is added after the 5 bit time counters.

The Slot assignment for S-CH in case of R/O-tag is specified in Table 81.

Table 81 — Slot assignment for S-CH in case of R/O-tag

Correlator													Time counters	Parity	Tail
Clock recovery															
B19	B18	B17	B16	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6...B2	B1	B0
1	0	1	0	1	1	0	0	1	1	1	1	1			0

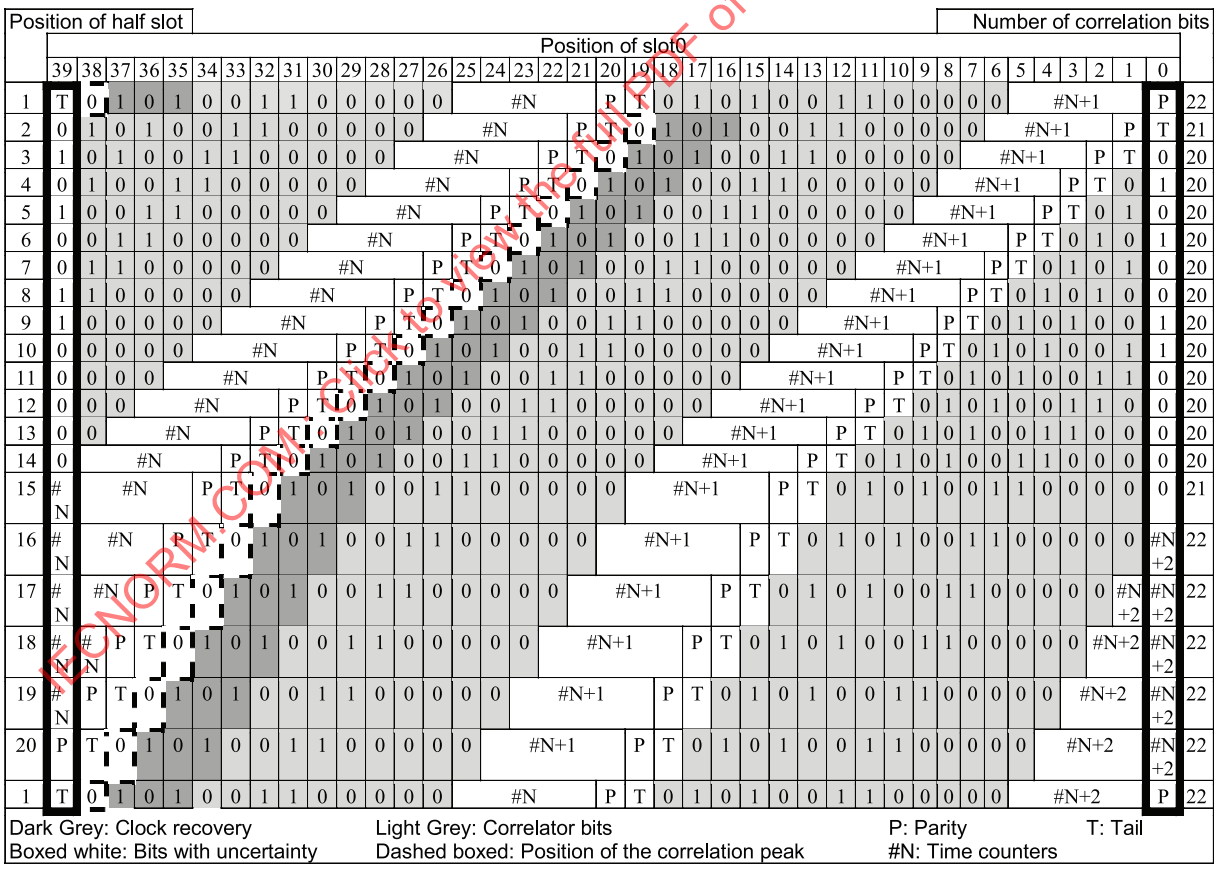
B6...B2: Time counters from 1 to 30 for standard R/O-tags, 31,0 for R/O-tags for high-speed applications.

Channel coding: Not applicable.

B1: supplements B6...B2 to achieve even number parity (identical for both tag types).

Decoding: By means of a correlator. The correlation is executed in two steps to keep false alarms to a minimum. In the first step, only the bits in the correlator word (B19...B7) are included in the correlation. In the second step, the remaining known bits in an interrogator slot (e.g. slot0) are included in the correlation, too. S-CH is half as long as slot0; therefore, there are still known bits. The number of known bits depends on where the correlation peak was found in the first step. This implies that the second correlation word has to be established dynamically, according to Table 82. In the table, only the values for the R/W-tag can be seen. The values for an R/O-tag are just simple inverted in the correlation field (in the field of the time-reversed Barker sequence).

Table 82 — Second level correlation scheme



7.4.3.2 TagID read channel: MIDx-CH (return link, for both tag types)

Function: This channel is used to transmit the 32-bit ID_{31}, ..., ID_0 in two subsequent slots.

Data transmission: Return link with 76,8 kbit/s.

Subframe assignment: Not relevant since MIDx-CH is a part of the notification channel. The position in the subframe is not synchronous with the interrogator frame structure.

Slot assignment: A time-reversed Barker sequence with a length of 11 is used for clock and word recovery.

The Slot assignment for MID1 (for both types of tags)/MID3 (only for R/O-tags) is specified in Table 83.

The Slot assignment for MID2 (for both types of tags)/MID4 (only for R/O-tags) is specified in Table 84.

Table 83 — Slot assignment for MID1 (for both types of tags)/MID3 (only for R/O-tags)

Word synch.											27 bit TagID	Tail
Clock recovery												
B39	B38	B37	B36	B35	B34	B33	B32	B31	B30	B29	B28...B2	B1...B0
0	1	0	0	1	0	0	0	1	1	1	First part of TagID ID_{31}, ..., ID_5	0

Table 84 — Slot assignment for MID2 (for both types of tags)/MID4 (only for R/O-tags)

Word synch.											5 bit TagID	CRC over 32 bits	Tail
Clock recovery													
B39	B38	B37	B36	B35	B34	B33	B32	B31	B30	B29	B28...B24	B23...B2	B1...B0
0	1	0	0	1	0	0	0	1	1	1	ID_4,..., ID_0	CRC_{21}, ..., CRC_0	0

For a description of the memory mapping of the TagID, refer to Annex C.

Channel coding: A shortened Fire code is used for TagID channel coding (54,32). After generation, the coded 54 bits (ID_{31}, ..., ID_0, CRC_{21}, ..., CRC_0) are transmitted in slots MID1/MID3 and MID2/MID4 beginning with the MSB ID_{31}. Generator polynomial:

$$g(x) = x^{22} + x^{17} + x^{13} + x^9 + x^4 + 1$$

The Channel Coding algorithm is as follows:

For encoding:

— Initialize the CRC accumulator to all zeroes $0.....0_h$

— Divide in GF(2) the polynomial

$ID_{31}x^{53} + ID_{30}x^{52} + ... + ID_0x^{22}$

by the generator polynomial

$x^{22} + x^{17} + x^{13} + x^9 + x^4 + 1$,

obtain as remainder the polynomial

$CRC_{21}x^{21} + ... + CRC_0x^0$

— Attach the CRC bits (CRC_{21}, ..., CRC_0) to the end of the TagID bits (ID_{31}, ..., ID_0) and transmit the 54 codebits (ID_{31}, ..., ID_0, CRC_{21}, ..., CRC_0) MSB first

For decoding:

— Divide the code polynomial

$ID_{31}x^{53} + ... + ID_0x^{22} + CRC_{21}x^{21} + ... + CRC_0x^0$

pre-multiplied with a certain factor (to account for the shortened code)

by the generator polynomial

$x^{22} + x^{17} + x^{13} + x^9 + x^4 + 1$,

and use the remainder polynomial for error correction and error detection

7.4.3.3 TagData read channel: MINx-CH (return link, only for R/O-tag)

Function: This channel is used to transmit 64 bits tag data in four subsequent slots. This can be done by sending two pairs of two consecutive slots.

Data transmission: Return link with 76,8 kbit/s.

Subframe assignment: Not relevant since MINx-CH is a part of the notification channel. The position in the subframe is not synchronous with the interrogator frame structure.

Slot assignment: A time-reversed Barker sequence with a length of 11 is used for clock and word recovery.

The Slot assignment for MIN1/MIN3 is specified in Table 85.

The Slot assignment for MIN2/MIN4 is specified in Table 86.

Table 85 — Slot assignment for MIN1/MIN3

Word synch.											27 bit tag data	Tail
Clock recovery												
B39	B38	B37	B36	B35	B34	B33	B32	B31	B30	B29	B28...B2	B1...B0
0	1	0	0	1	0	0	0	1	1	1	First part of TagData D_{31}, ..., D_5	0

Table 86 — Slot assignment for MIN2/MIN4

Word synch.											5 bit tag data	CRC over 32 bits	Tail
Clock recovery													
B39	B38	B37	B36	B35	B34	B33	B32	B31	B30	B29	B28...B24	B23...B2	B1...B0
0	1	0	0	1	0	0	0	1	1	1	D_4, ..., D_0	CRC_{21}, ..., CRC_0	0

TagData bits: DATA63...DATA0

NOTE These bits can be used to extend the UserTagID or to store application related data in the tag.

Channel coding: For each pair of two consecutive slots transporting 32 data bits, a shortened Fire code is used for tag data channel coding (54,32). After generation, the coded 54 bits (D_{31}, ..., D_0, CRC_{21}, ..., CRC_0) are transmitted in slots MIN1/MIN3 and MIN2/MIN4 beginning with the MSB D_{31}. Generator polynomial:

$g(x) = x^{22} + x^{17} + x^{13} + x^9 + x^4 + 1$

The Channel Coding algorithm is as follows:

For encoding:

— Initialize the CRC accumulator to all zeroes 0.....0_h

— Divide in GF(2) the polynomial

$D_{31}x^{53} + D_{30}x^{52} + ...+ D_0x^{22}$

by the generator polynomial

$x^{22} + x^{17} + x^{13} + x^9 + x^4 + 1$,

obtain as remainder the polynomial

$CRC_{21}x^{21} + ... + CRC_0x^0$

— Attach the CRC bits (CRC_{21}, ..., CRC_0) to the end of the data bits (D_{31}, ..., D_0) and transmit the 54 codebits (D_{31}, ..., D_0, CRC_{21}, ..., CRC_0) MSB first

For decoding:

— Divide the code polynomial

$D_{31}x^{53} + ... + D_0x^{22} + CRC_{21}x^{21} + ... + CRC_0x^0$

pre-multiplied with a certain factor (to account for the shortened code)

by the generator polynomial

$x^{22} + x^{17} + x^{13} + x^9 + x^4 + 1$,

and use the remainder polynomial for error correction and error detection

7.4.3.4 Interrogator-ID read channel: SID-CH (forward link, only for R/W-tag)

Function: This channel is used to transmit the 10 bit long interrogator-ID and a 15-bit counter value to the tag. The counter value enables communication: it shows where the tag has to expect the first command on the communication channel.

Data transmission: Forward link with 384 kbit/s.

Subframe assignment: Not relevant since SID-CH is a part of the notification channel. The position in the subframe is not synchronous with the interrogator frame structure.

Slot assignment: Only the first 112 bits out of 200 are assigned. The remaining 88 bits are not evaluated by the tag. For word synchronization, a sequence (TSC1) with a length of 16 is used. The default correlator threshold for word synchronization is 13 (the value shall exceed 13: two bit errors maximum).

The Slot assignment for SID-CH part 1 is specified in Table 87.

The Slot assignment for SID-CH part 2 is specified in Table 88.

The Slot assignment for SID-CH part 3 is specified in Table 89.

Table 87 — Slot assignment for SID-CH part 1

Level detector (20 bits)								Wake-up and clock recovery (36 bits)							
B199	B198	B197	B196	....	B182	B181	B180	B179	B178	B177	B176	....	B146	B145	B144
0	1	0	1	....	1	0	1	0	1	0	1	....	1	0	1

Table 88 — Slot assignment for SID-CH part 2

Word synch.: 16 bit sequence TSC1 (16 bits)															
B143	B142	B141	B140	B139	B138	B137	B136	B135	B134	B133	B132	B131	B130	B129	B128
1	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0

Table 89 — Slot assignment for SID-CH part 3

Interrogator-ID (10 bits)	Counter value (15 bits)	CRC over 25 bits (15 bits)	Not evaluated by tag
B127...B118	B117...B103	B102...B88	B87...B0
D_{24}, ..., D_{15}	D_{14}, ..., D_0	CRC_{14}, ..., CRC_0	

NOTE The sequence "0 1" is repeated for B195, B183 and B175, and B147.

Channel coding: A shortened Fire code is used for SID-CH channel coding (40,25). After generation, the coded 40 bits (D_{24}, ..., D_0, CRC_{14}, ..., CRC_0) are transmitted beginning with the MSB D_{24}. Generator polynomial:

$$g(x) = x^{15} + x^{10} + x^9 + x^6 + x + 1$$

The Channel Coding algorithm is as follows:

For encoding:

— Initialize the CRC accumulator to all zeroes $0.....0_h$

— Divide in GF(2) the polynomial

$D_{24}x^{39} + D_{23}x^{38} + ...+ D_0x^{15}$

by the generator polynomial

$x^{15} + x^{10} + x^9 + x^6 + x + 1$,

obtain as remainder the polynomial

$CRC_{14}x^{14} + ...+ CRC_0x^0$

— Attach the CRC bits (CRC_{14}, ..., CRC_0) to the end of the data bits (D_{24}, ..., D_0) and transmit the 40 codebits (D_{24}, ..., D_0, CRC_{14}, ..., CRC_0) MSB first

For decoding:

— Divide the code polynomial

$D_{24}x^{39} + ...+ D_0x^{15} + CRC_{14}x^{14} + ...+ CRC_0x^0$

pre-multiplied with a certain factor (to account for the shortened code)

by the generator polynomial

$x^{15} + x^{10} + x^9 + x^6 + x + 1$

and use the remainder polynomial for error correction and error detection

7.4.3.5 Reserved function forward link channel: RFD-CH (forward link, only for R/W tags)

Function: Reserved for proprietary future use.

7.4.3.6 Reserved function return link channel: RFU-CH (return link, for both types of tag)

Function: Reserved for proprietary future use. The functionality can be different for the two tag types.

7.4.3.7 Interrogator training sequence type1 channel: TS1-CH (return link/without logical channel)

Function: This channel is used to ease the implementation of the hardware.

Data transmission: Return link with a data rate out of 200 to 400 kbit/s. An alternating series of 0 and 1 started with 0. This signal is not differentially pre-coded, with the exception of the last bit. The last bit has to be differentially pre-coded to enable differential demodulation in the subsequent slot.

Subframe assignment: Set up before the S-CH, MID-CH, and RFU-CH channels.

Slot assignment: Not relevant.

7.4.3.8 Command slot channel: CS-CH (forward link)

Function: This channel is used to transmit the commands from the interrogator to the tag. The total amount per slot is 200 bits, with a net data bit content of 120 bits. The remaining bits are used for clock recovery, word synchronization and for error protection.

Data transmission: Forward link with 384 kbit/s.

The Subframe assignment for CS-CH in case of W-CH, RM-CH, MID-CH is specified in Table 90.

The Subframe assignment for CS-CH in case of RL-CH is specified in Table 91.

The Slot assignment for CS-CH part 1 is specified in Table 92.

The Slot assignment for CS-CH part 2 is specified in Table 93.

The Slot assignment for CS-CH part 3 is specified in Table 94.

The Slot assignment for CS-CH part 4 is specified in Table 95.

Table 90 — Subframe assignment for CS-CH in case of W-CH, RM-CH, MID-CH

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
		•											

Table 91 — Subframe assignment for CS-CH in case of RL-CH

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
		•										•	

Slot assignment: 12 bits out of 200 are used for clock recovery. For word synchronization, a sequence (TSC1) with a length of 26 is used. The default correlator threshold for word synchronization is 24 (the value shall exceed 24: two bit errors maximum). The remaining 162 bits are coded net data bits.

Table 92 — Slot assignment for CS-CH part 1

Clock recovery (12 bits)											
B199	B198	B197	B196	B195	B194	B193	B192	B191	B190	B189	B188
0	1	0	1	0	1	0	1	0	1	0	1

Table 93 — Slot assignment for CS-CH part 2

B187...B162: Word synch. (TSC1) (26 bits)																								
0	0	1	0	0	1	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	1

Table 94 — Slot assignment for CS-CH part 3

Command type	EOC	Interrogator frame structure	TagID	CRC protection over bits B161...B132
(4 bits)	(1 bit)	(7 bits)	(18 bits)	(44 bits)
B161...B158	B157	B156...B150	B149...B132	B131...B88
D_{29}, ..., D_{26}	D_{25}	D_{24}, ..., D_{18}	D_{17}, ..., D_0	CRC_{43}, ..., CRC_0

Table 95 — Slot assignment for CS-CH part 4

Block length	Reserve	Start address	Reserve	CRC protection over bits B87...B58	Reserve
(8 bits)	(1 bit)	(18 bits)	(3 bits)	(44 bits)	(14 bits)
B87...B80	B79	B78...B61	B60...B58	B57...B14	B13...B0
D_{29}, ..., D_{22}	D_{21}	D_{20}, ..., D_3	D_2, ..., D_0	CRC_{43}, ..., CRC_0	

Description of fields:

Command type: Refer to 7.6.1 and 7.6.3.

EOC: End_Of_Communication (EOC) is signalled with one bit in the command field.

Interrogator frame structure: Indicates the number of subframes contained in a frame.

TagID: Only the ID31...ID14 range shall be transmitted in the command slot.

Block length: Indicates how many bytes the transmitted block contains.

Start address: Indicates where the first byte in the transmitted block should be written or read to. B79 shall be set to 0.

Channel coding: A shortened Fire code (74,30) is repeatedly used for coding the CS-CH data bits. After generation, the coded 74 bits (D_{29}, ..., D_0, CRC_{43}, ..., CRC_0) are transmitted beginning with the MSB D_{29}. Generator polynomial (same as for R-CH):

$$g(x) = x^{44} + x^{30} + x^{29} + x^{15} + x + 1$$

The Channel Coding algorithm is as follows:

For encoding:

— Initialize the CRC accumulator to all zeroes 0.....0_h

— Divide in GF(2) the polynomial

$D_{29}x^{73} + D_{28}x^{72} + ...+ D_0x^{44}$

by the generator polynomial

$x^{44} + x^{30} + x^{29} + x^{15} + x + 1$

obtain as remainder the polynomial

$CRC_{43}x^{43} + ...+ CRC_0x^0$

— Attach the CRC bits (CRC_{43}, ..., CRC_0) to the end of the data bits (D_{29}, ..., D_0) and transmit the 74 codebits (D_{29}, ..., D_0, CRC_{43}, ..., CRC_0) MSB first

For decoding:

— Divide the code polynomial

$D_{29}x^{73} + ... + D_0x^{44} + CRC_{43}x^{43} + ... + CRC_0x^0$

pre-multiplied with a certain factor (to account for the shortened code)

by the generator polynomial

$x^{44} + x^{30} + x^{29} + x^{15} + x + 1$

and use the remainder polynomial for error correction and error detection

7.4.3.9 Read channels: R-CH (return link)

Function: These channels are used to transmit the tag net data to the interrogator. The total amount per slot is 200 bits, with a net data bit content of 96 bits. The remaining bits are used for clock recovery, word synchronization and for error protection.

Data transmission: Return link with 384 kbit/s.

Subframe assignment:

The Subframe assignment for R-CH in case of RM-CH is specified in Table 96.

The Subframe assignment for R-CH in case of RL-CH is specified in Table 97.

The Slot assignment for R-CH part 1 is specified in Table 98.

The Slot assignment for R-CH part 2 is specified in Table 99.

Table 96 — Subframe assignment for R-CH in case of RM-CH

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
				•	•	•	•	•	•	•	•	•	

Table 97 — Subframe assignment for R-CH in case of RL-CH

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
				•	•	•	•	•	•	•			

Slot assignment: A time-reversed Barker sequence with a length of 11 is used for clock and word recovery. The remaining 189 bits are coded net data bits, split up into two identical parts.

Table 98 — Slot assignment for R-CH part 1

Word synch.										
Clock recovery										
B199	B198	B197	B196	B195	B194	B193	B192	B191	B190	B189
0	1	0	0	1	0	0	0	1	1	1

Table 99 — Slot assignment for R-CH part 2

Net data bits (48 bits)	**CRC over previous 48 bits** (44 bits)	**Net data bit** (48 bits)	**CRC over bits B96...B49** (44 bits)	**Tail bit**
B188...B141 D_{47}, ..., D_0	B140...B97 CRC_{43}, ... ,CRC_0	B96...B49 D_{47}, ..., D_0	B48...B5 CRC_{43}, ..., CRC_0	B4...B0

Channel coding: A shortened Fire code (92,48) is repeatedly used for coding the R-CH data bits. After generation, the coded 92 bits (D_{47}, ..., D_0, CRC_{43}, ..., CRC_0) are transmitted beginning with the MSB D_{47}. Generator polynomial:

$$g(x) = x^{44} + x^{30} + x^{29} + x^{15} + x + 1$$

The Channel Coding algorithm is as follows:

For encoding:

— Initialize the CRC accumulator to all zeroes $0.....0_h$

— Divide in GF(2) the polynomial

$D_{47}x^{91} + D_{46}x^{90} + ...+ D_0x^{44}$

by the generator polynomial

$x^{44} + x^{30} + x^{29} + x^{15} + x + 1$

obtain as remainder the polynomial

$CRC_{43}x^{43} + ...+ CRC_0x^0$

— Attach the CRC bits (CRC_{43}, ..., CRC_0) to the end of the data bits (D_{47}, ..., D_0) and transmit the 92 codebits (D_{47}, ..., D_0, CRC_{43}, ..., CRC_0) MSB first

For decoding:

— Divide the code polynomial

$D_{47}x^{91} + ...+ D_0x^{44} + CRC_{43}x^{43} + ...+ CRC_0x^0$

pre-multiplied with a certain factor (to account for the shortened code)

by the generator polynomial

$x^{44} + x^{30} + x^{29} + x^{15} + x + 1$

and use the remainder polynomial for error correction and error detection

7.4.3.10 Write channel: W-CH (forward link)

Function: This channel is used to transmit net data from interrogator to tag. The total amount per slot is 200 bits, with a net data bit content of 128 bits. The remaining bits are used for clock recovery, word synchronization and for error protection.

Data transmission: Forward link with 384 kbit/s.

The Subframe assignment for W-CH is specified in Table 100.

The Slot assignment for W-CH part 1 is specified in Table 101.

The Slot assignment for W-CH part 2 is specified in Table 102.

The Slot assignment for W-CH part 3 is specified in Table 103.

Table 100 — Subframe assignment for W-CH

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
			•	•	•	•	•	•	•	•	•		

Slot assignment: 12 bits out of 200 are used for clock recovery. For word synchronization, a sequence (TSC1) with a length of 16 is used. The remaining 172 bits are coded net data bits.

Table 101 — Slot assignment for W-CH part 1

Clock recovery (12 bits)											
B199	B198	B197	B196	B195	B194	B193	B192	B191	B190	B189	B188
0	1	0	1	0	1	0	1	0	1	0	1

Table 102 — Slot assignment for W-CH part 2

Word synch.: 16 bit sequence TSC1															
B187	B186	B185	B184	B183	B182	B181	B180	B179	B178	B177	B176	B175	B174	B173	B172
1	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0

Table 103 — Slot assignment for W-CH part 3

Net data bits (128 bits)	**CRC over previous 128 bits** (44 bits)
B171...B44	B43...B0
D_{127}, ..., D_0	CRC_{43}, ..., CRC_0

Channel coding: A shortened Fire code is used for W-CH channel coding (172,128). After generation, the coded 172 bits (D_{127}, ..., D_0, CRC_{43}, ..., CRC_0) are transmitted beginning with the MSB D_{127}. Generator polynomial (same as R-CH):

$$g(x) = x^{44} + x^{30} + x^{29} + x^{15} + x + 1$$

The Channel Coding algorithm is as follows:

For encoding:

— Initialize the CRC accumulator to all zeroes $0.....0_h$

— Divide in GF(2) the polynomial

$D_{127}x^{171} + D_{126}x^{170} + ...+ D_0x^{44}$

by the generator polynomial

$x^{44} + x^{30} + x^{29} + x^{15} + x + 1$

obtain as remainder the polynomial

$CRC_{43}x^{43} + ...+ CRC_0x^0$

— Attach the CRC bits (CRC_{43}, ..., CRC_0) to the end of the data bits (D_{127}, ..., D_0) and transmit the 172 codebits (D_{127}, ..., D_0, CRC_{43}, ..., CRC_0) MSB first

For decoding:

— Divide the code polynomial

$D_{127}x^{171} + ...+ D_0x^{44} + CRC_{43}x^{43} + ...+ CRC_0x^0$

pre-multiplied with a certain factor (to account for the shortened code)

by the generator polynomial

$x^{44} + x^{30} + x^{29} + x^{15} + x + 1$

and use the remainder polynomial for error correction and error detection

The CRC definitions and calculation example are shown in Annex D.

7.4.3.11 Confirm write channel: CW-CH (return link)

Function: This channel signals whether the transmission of data from interrogator to tag in a given subframe was free of errors or not. For a transmission to be error-free, all the slots shall have been received without errors. A "not error-free" signal results in an ARQ procedure for this communication.

Data transmission: Return link with 384 kbit/s.

The Subframe assignment for CW-CH is specified in Table 104.

The Slot assignment for CW-CH part 1is specified in Table 105.

The Slot assignment for CW-CH part 2 is specified in Table 106.

The Slot assignment for CW-CH part 3 is specified in Table 107.

Table 104 — Subframe assignment for CW-CH

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
													•

Slot assignment: A time-reversed Barker sequence with a length of 11 is used for clock and word recovery. In the case of W-CH, the CRCs are evaluated on a slot-by-slot basis. If all the slot CRCs are error-free (or feature correctable errors only), the CRC_OK bit shall be set. This bit is transmitted to the interrogator with 2 bits × 26 bits consecutively (sequence TSC1). If the received CRC was not okay, a word containing nothing but 0's shall be generated with a length of 2 bits × 26 bits. The interrogator does not evaluate the remaining bits.

Table 105 — Slot assignment for CW-CH part 1

Word synch.										
Clock recovery										
B199	B198	B197	B196	B195	B194	B193	B192	B191	B190	B189
0	1	0	0	1	0	0	0	1	1	1

Table 106 — Slot assignment for CW-CH part 2

							B188...B163: CRC_OK: true (TSC1) (26 bits)																	
0	0	1	0	0	1	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	1
							B188...B163: CRC_OK: false																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 107 — Slot assignment for CW-CH part 3

B162...B137: CRC_OK: true (TSC1) (26 bits)																									B136...B0
0	0	1	0	0	1	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	1	—
B162...B137: CRC_OK: false																										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Channel coding: None.

Decoding: By means of correlator. The default correlator threshold for CRC_OK word detection is 50 (the value shall exceed 50: 2 bit errors maximum in 52 bits).

7.4.3.12 Interrogator training sequence type2 channel: TS2-CH (return link/without logical channel)

Function: This channel is used only to ease the implementation of the hardware.

Data transmission: Return link with a data rate out of 200 to 400 kbit/s. An alternating series of 0 and 1 started with 0.

This signal is not differentially pre-coded with the exception of the last bit. The last bit has to be differentially pre-coded to enable differential demodulation in the subsequent slot.

Subframe assignment: Always transmitted before the return link slot if no physical return link slot was sent before the "first" return link slot. That means that this channel shall be inserted before the first return link slot, containing “real” information that shall be sent.

The Subframe assignment for TS2-CH in case of W-CH is specified in Table 108.

The Subframe assignment for TS2-CH in case of RM-CH, RL-CH, MID-CH is specified in Table 109.

The Slot assignment for TS2-CH is specified in Table 110.

Table 108 — Subframe assignment for TS2-CH in case of W-CH

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
												•	

Table 109 — Subframe assignment for TS2-CH in case of RM-CH, RL-CH, MID-CH

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
			•										

Table 110 — Slot assignment for TS2-CH

Interrogator training sequence (200 bits)				
B199	B198	...	B1	B0
0	1	...	0	1

7.4.3.13 Command slot training sequence: TS3-CH (forward link/without logical channel)

Function: This channel is used only to ease the implementation of the hardware.

Data transmission: Forward link with 384 kbit/s. The last 30 bits are an alternating series of 2 zeroes and 2 ones, starting with 2 ones.

Subframe assignment: Always transmitted only before CS-CH if CS-CH transmitted in slot2.

The Subframe assignment for TS3-CH in case of C-CH is specified in Table 111.

The Slot assignment for TS3-CH is specified in Table 112.

Table 111 — Subframe assignment for TS3-CH in case of C-CH

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
	•												

Table 112 — Slot assignment for TS3-CH

Command slot training sequence (30 bits)									
B199…B30	B29	B28	B27	B26	…	B3	B2	B1	B0
—	1	1	0	0	…	0	0	1	1

7.4.3.14 Spectrum check channel: SC-CH (return link/without carrier)

Function: The interrogator uses this channel to measure the RSSI values in the allowed frequency band within the allowed channels. The stored values are used for determining free frequencies for notification and communication.

Data transmission: None.

Subframe assignment: Only if there is neither communication nor notification.

The Subframe assignment for SC-CH is specified in Table 113.

Table 113 — Subframe assignment for SC-CH

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
		•	•	•	•	•	•	•	•	•	•	•	

Slot assignment: Not applicable.

Channel coding: None.

Decoding: Not relevant.

7.5 Channel coding and sequences

7.5.1 Synchronization and CRC patterns

The Clock and word synchronization words for the physical channels is specified in Table 114.

The CRC parameterization for the physical channels is specified in Table 115.

Table 114 — Clock and word synchronization words for the physical channels

Channel type		Clock	Word	Description
N-CH	S-CH (R/W-tag)	0101	001100000	Word incl. clock
	S-CH[a] (R/O-tag)	1010	110011111	Word incl. clock
	MID-CH	0100	1000111	Word incl. clock
	MIN-CH	0100	1000111	Word incl. clock
	SID-CH	010101…010101 (36 bits)	1011100001000100	
	TS1-CH	Not relevant	Not relevant	
C-CH	R-CH	0100	1000111	Word incl. clock
	W-CH	010101010101	1011100001000100	
	CW-CH	0100	1000111	Word incl. clock
	TS2-CH	Not relevant	Not relevant	

[a] The clock recovery can work also with the capital letters: 0101: the same sequence as for an R/W-tag.

Table 114 *(continued)*

Channel type		Clock	Word	Description
	CS-CH	010101010101	00100101110000100010010111	
	TS3-CH	Not relevant	Not relevant	
SC-CH		Not relevant	Not relevant	

[a] The clock recovery can work also with the capital letters: 0101: the same sequence as for an R/W-tag.

Table 115 — CRC parameterization for the physical channels

Channel type		n'	k'	Generator polynomial	Remarks
N-CH	S-CH	6	5	Even parity only	n': total number of bits in a CRC block. k': total number of net bits in a CRC block. In case of correction, only the data field shall be corrected. No wrap around shall be applied for the correction.
	MID-CH	54	32	$x^{22} + x^{17} + x^{13} + x^9 + x^4 + 1$	
	MIN-CH	54	32	$x^{22} + x^{17} + x^{13} + x^9 + x^4 + 1$	
	SID-CH	40	25	$x^{15} + x^{10} + x^9 + x^6 + x + 1$	
C-CH	TS1-CH	—	—	—	
	R-CH	92	48	$x^{44} + x^{30} + x^{29} + x^{15} + x + 1$	
	W-CH	172	128	$x^{44} + x^{30} + x^{29} + x^{15} + x + 1$	
	CW-CH	52	2	52 bit correlator	
	TS2-CH	—	—	—	
	CS-CH	74	30	$x^{44} + x^{30} + x^{29} + x^{15} + x + 1$	
	TS3-CH	—	—	—	
SC-CH		—	—	—	

7.6 Command set for the command slot channel: CS-CH (only for R/W-tag)

7.6.1 Command types

All tags with the same IC manufacturer code and the same IC version number shall behave the same.

7.6.1.1 Mandatory

Mandatory command shall be supported by all R/W-tags that claim to be compliant. Interrogators which claim compliance for R/W-operation shall support all mandatory commands.

7.6.1.2 Optional

If the tag does not support an optional command, it shall remain silent.

Optional commands are commands that are specified within the International Standard. Interrogators which claim compliance for R/W-operation shall be technically capable of performing all optional commands that are specified in the International Standard (although need not be set up to do so). R/W-tags may or may not support optional commands. If an optional command is used, it shall be implemented in the manner specified in the International Standard.

7.6.1.3 Custom

Custom commands may be enabled by an International Standard, but they shall not be specified in that International Standard. A custom command shall not solely duplicate the functionality of any mandatory or optional command defined in the International Standard by a different method.

An interrogator shall only send a custom command to a tag if the manufacturer of the tag specifies such a command.

During the notification process, the TagID is sent to the interrogator. All custom commands shall be addressed individually to specific manufactured tags. This allows IC manufacturers to implement custom commands without risking duplication of command codes and thus misinterpretation.

7.6.1.4 Proprietary

Proprietary commands may be enabled by an International Standard, but they shall not be specified in that International Standard. A proprietary command shall not solely duplicate the functionality of any mandatory or optional command defined in the International Standard by a different method.

IC and tag manufacturers use these commands for various purposes such as tests, programming of system information, etc. They are not specified in this document. The IC manufacturer may at its option document them or not. It is allowed that these commands are disabled after IC and/or tag manufacturing.

7.6.2 Command set

General notes:

If a command cannot be decoded, and provided this is not the first command (in order to repeat a read attempt, the information on the interrogator frame structure is needed, and this information is available only with the first correctly decoded command), the tag shall try ten more times (in the ten subsequent frames) to decode the command. If the tag does not succeed in decoding a command, it shall return to the sleep mode. If the tag cannot decode a first command, it shall return to the sleep mode immediately, without trying to repeat the operation.

7.6.2.1 Write

Function: This command shall transmit a maximum of 144 bytes in a subframe to the tag.

NOTE This command requires only bits B161 to B14 in CS-CH to be evaluated (the command has arguments).

7.6.2.2 Long_Read

Function: This command shall transmit more than 84 bytes in a subframe to the interrogator.

NOTE For information on how to proceed further, refer to RM-CH. This command requires only bits B161 to B14 in CS-CH to be evaluated (the command has arguments).

7.6.2.3 Short_Read

Function: This command shall transmit a maximum of 84 bytes in a subframe to the interrogator.

NOTE For information on how to proceed further, refer to RL-CH. This command requires only bits B161 to B14 in CS-CH to be evaluated (the command has arguments). The only exception is when EOC is detected to be active in slot2. In that case, only bits B161 to B88 need to be evaluated.

7.6.2.4 Init

Function: This command shall transmit one byte to the tag. In the tag, this byte is written into all RAM cells.

NOTE On the protocol, **Init** behaves in the same way as **Write**. That is why the interrogator expects a CW-CH in slot13. This command requires only bits B161 to B88 in CS-CH to be evaluated (the command has no arguments).

7.6.2.5 Wait

Function: This command signals to the tag that the tag has to wait – for the length of one frame – for a new command.

NOTE On the protocol, **Wait** behaves in the same way as **Short_Read** without a data field. This command requires only bits B161 to B88 in CS-CH to be evaluated (the command has no arguments).

7.6.3 Command codes

The Command codes in slot2 are specified in Table 116.

The Command codes in slot12 are specified in Table 117.

Table 116 — Command codes in slot2

Name	Type	Command code B161...B158				EOC B157	Function
Wait	Mandatory	0	0	0	0	x	Subframe is not filled with data. An EOC in slot12 is not to be expected. This is an NOP command. The tag shall decode the next command in the next frame. It shall not be possible to terminate the communication.
Short_Read	Mandatory	0	0	0	1	0	Subframe filled with a maximum of 84 bytes of read data. An EOC is to be expected in slot12. After EOC has been received in slot12, the tag shall return to the sleep mode. If there is no EOC in slot12, the tag shall wait for a command to arrive in the next frame.
						1	Confirmation that a Long_Read or Wait command was successfully transmitted in the previous frame. The tag shall immediately return to the sleep mode. This command requires only bits B161 to B88 to be evaluated (the command has no arguments).
Long_Read	Mandatory	0	0	1	1	x	Subframe filled with read data. The communication in this subframe cannot be terminated.
Write	Mandatory	1	1	0	0	0	Subframe filled with write data. The communication in this subframe shall not be terminated. A feedback on the validity of the data received by tag in this subframe shall be sent to the interrogator on the CW-CH channel.
						1	The communication in this subframe shall be terminated when the CRCs signal valid data for all the slots. A feedback on the validity of the data received in this subframe shall be sent to the interrogator on the CW-CH channel. Once the data have been written to RAM, the tag shall return to the sleep mode.
Init	Optional	1	1	1	1	x	A feedback on the validity of the data received in this subframe (INIT byte) shall be sent to the interrogator on the CW-CH channel. The communication cannot be terminated during initialization. During initialization, the tag shall be polled in each respective subframe to find out whether or not the initialization has been terminated.
Reserved for future use	Optional	0 1	1 0	0 1	1 0	x	
IC Mfg dependent	Custom	0 0 1 1 1	0 1 0 1 1	1 0 1 0 1	0 0 1 1 0	x	

Table 116 *(continued)*

Name	Type	Command code				EOC	Function
		B161...B158				B157	
IC Mfg dependent	Proprietary	0	1	1	0	x	
		0	1	1	1		
		1	0	0	0		
		1	0	0	1		

Table 117 — Command codes in slot12

Name	Type	Command code				EOC	Function
		B161...B158				B157	
EOC	Mandatory	0	0	0	1	1	Confirmation that a Short_Read command was successfully transmitted in this frame. The tag shall immediately return to the sleep mode. This command requires only bits B161 to B88 to be evaluated (the command has no arguments). EOC = 0 shall not be used (invalid operation).
Reserved for future use	Optional	0	1	0	1	x	
		1	0	1	0		
IC Mfg dependent	Custom	0	0	0	0	x	
		0	0	1	0		
		0	0	1	1		
		0	1	0	0		
		1	0	1	1		
		1	1	0	0		
		1	1	0	1		
		1	1	1	0		
		1	1	1	1		
IC Mfg dependent	Proprietary	0	1	1	0	x	
		0	1	1	1		
		1	0	0	0		
		1	0	0	1		

In the case of command decoding error, the tag shall try 10 more times to decode the command. After the eleventh unsuccessful attempt, the tag shall return to the sleep mode. The exception to this rule shall be the "first" command. If the tag fails to decode the first command, it shall return to the sleep mode immediately.

For mandatory and optional commands: in the case of 'x', the EOC bit shall not be evaluated.

8 MODE 3: Active RFID ITF network

8.1 General

The mode describes the basis for a wireless network standard for devices used in the shipping and logistics industry. It describes a standardized wireless network for freight containers (and subsequently other shipping conveyance types, e.g. intermodal containers, trucks, rail cars, etc.) and allows the market to determine the type(s) of devices to build and integrate using the network based on commercial requirements and/or government concerns.

A common, globally ubiquitous network architecture allows the commercial sector to create and build devices for the intermodal freight container (maritime and transportation) industry. With this known/standardized infrastructure, device manufacturers may build products that support commercial endeavours secured by the fact that the product will work globally and be interoperable among carriers, between ports/terminals, and between nations on all continents.

Autonomous devices (such as active RFID tags for example attached to containers, chassis, or other equipment) characterized by moving around with the object they are attached to will benefit from one globally available and consistent network. They are generally small and independent from any power supply other than the power integrated into the device itself (battery, power scavenging capability) and in general have an expected life span of between 3 years and 5 years (or more). Low power usage is critical.

8.2 Operational requirements

This mode describes the implementation for a wireless network protocol for devices used in the transportation and maritime shipping and logistics industry. It describes a wireless network for freight containers (and subsequently other shipping conveyance types, e.g. intermodal containers, trucks, rail cars, etc.) and allows the market to determine the type(s) of devices to build and integrate using the network based on commercial requirements and/or government concerns.

The purpose of the protocol is to support compliant devices as required by shippers, ocean carriers, marine terminals and port operators, logistics providers, and intermodal carriers.

The network should ensure interoperability between countries, terminals, and ports given ocean carriers use multiple ports and shall be assured that any investment in compliant devices and/or compliant hardware work at all terminals (carrier owned or not) in the same manner. Moreover, terminal operators shall be assured that any compliant network technology can be installed at all ports in a similar fashion regardless of geographic location.

The network standard shall support one standard intermodal freight container air interface for compliant devices globally.

The network standard shall define how information moves securely and reliably across the network.

The network shall meet the safety requirements.

The network standard is designed to use the minimal amount of installed infrastructure.

Where required, the network standard shall provide data definition.

The network standard is designed to be globally deployed at a very low cost.

The network standard provides a method to establish a mesh network.

The network shall allow multiple compliant device types. Device categories can include, but are not limited to, tracking devices, container security devices, sensor devices, location devices, and infrastructure devices such as network coordinators and handhelds.

The network standard shall co-exist with other wireless protocols such as Wi-Fi and other wireless communication and RTLS networks.

8.3 Network Physical Layer description

The Network Physical Layer is a radio protocol based on ISO/IEC/IEEE 8802-15-4.

Any ISO/IEC/IEEE 8802-15-4 integrated circuit (IC) may be utilized to implement this document.

This document utilizes 16 channels (channel 11 to channel 26) over the 2,405 to 2,483 GHz spectrum, supporting a 250 000 bits per second data rate using QPSK modulation (see IEEE 802.15.4:2006, Clause 10).

Channel conflict resolution shall be attained utilizing Carrier Sense Multiple Access (CSMA) with a hold off up to 10 ms (see IEEE 802.15.4:2006, 5.1.1.4).

This Mode supports sleep modes. There is a network discovery method which utilizes an NDB that includes wake cycles parameters and user-controlled channel allocation.

8.4 Network description

8.4.1 General

The network shall be made up of one or more network coordinators, each of which is associated with at least one server connected coordinator (SCC). The Network Coordinator and SCC may be the same device.

8.4.2 Network topology

The following list summarizes the network topology:

a) Ocean container devices described in this document are Tag Talk Last (TTL).

b) Tags and remote devices will not initiate network connections unless a network coordinator emits a broadcasted command or NDB packet.

c) After associating, a tag, also named device, may emit transmissions unilaterally for certain applications (e.g. low latency alarms, exits, arrivals, etc.)

d) This network specification shall support multiple topologies. All devices shall support star, trunk, and peer-to-peer topologies.

   1) Point to multi-point topologies (star topology) as shown in Figure 22.

   2) Trunk coordinator configurations as shown in Figure 23.

   3) Peer-to-peer topology as shown in Figure 24.

   4) Mesh topology as shown in Figure 25.

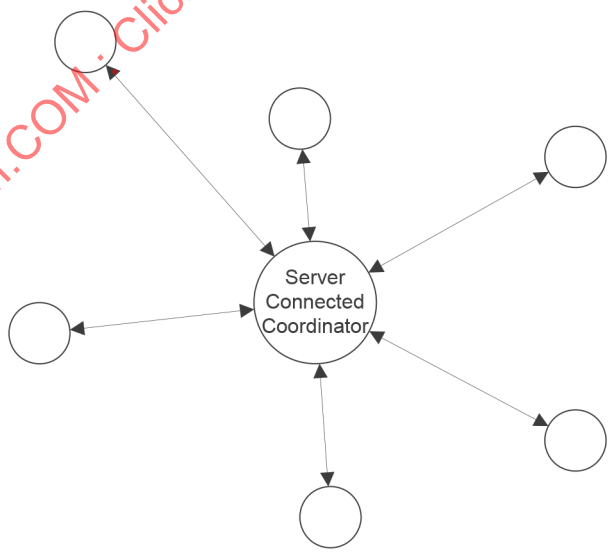


Figure 22 — Star topology

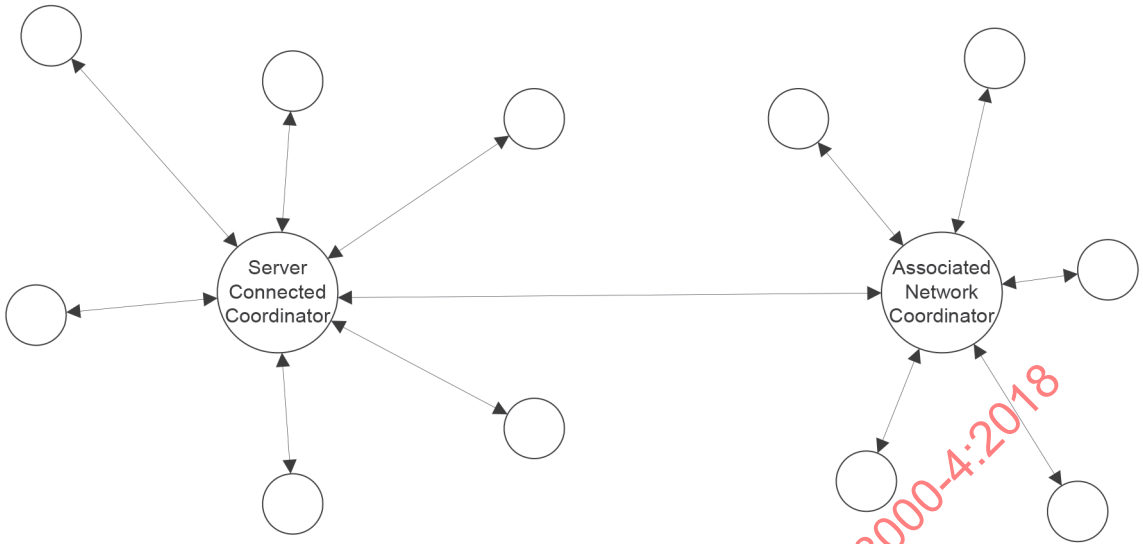


Figure 23 — Trunk topology

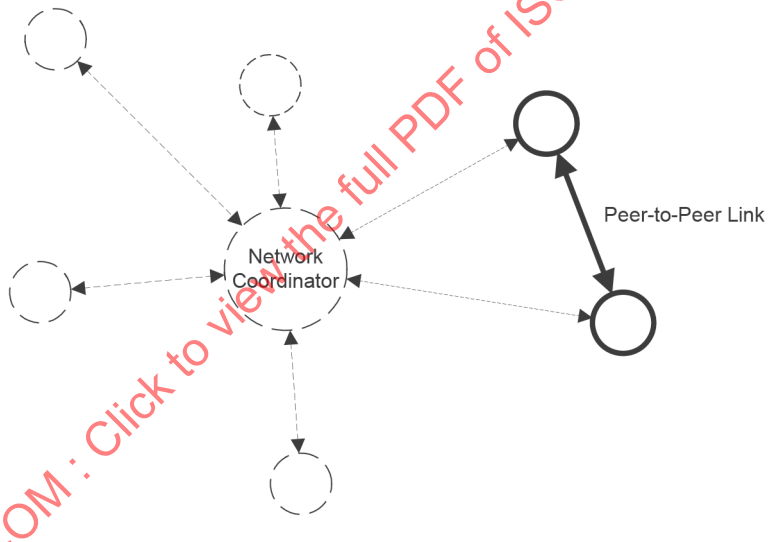


Figure 24 — Peer-to-peer topology

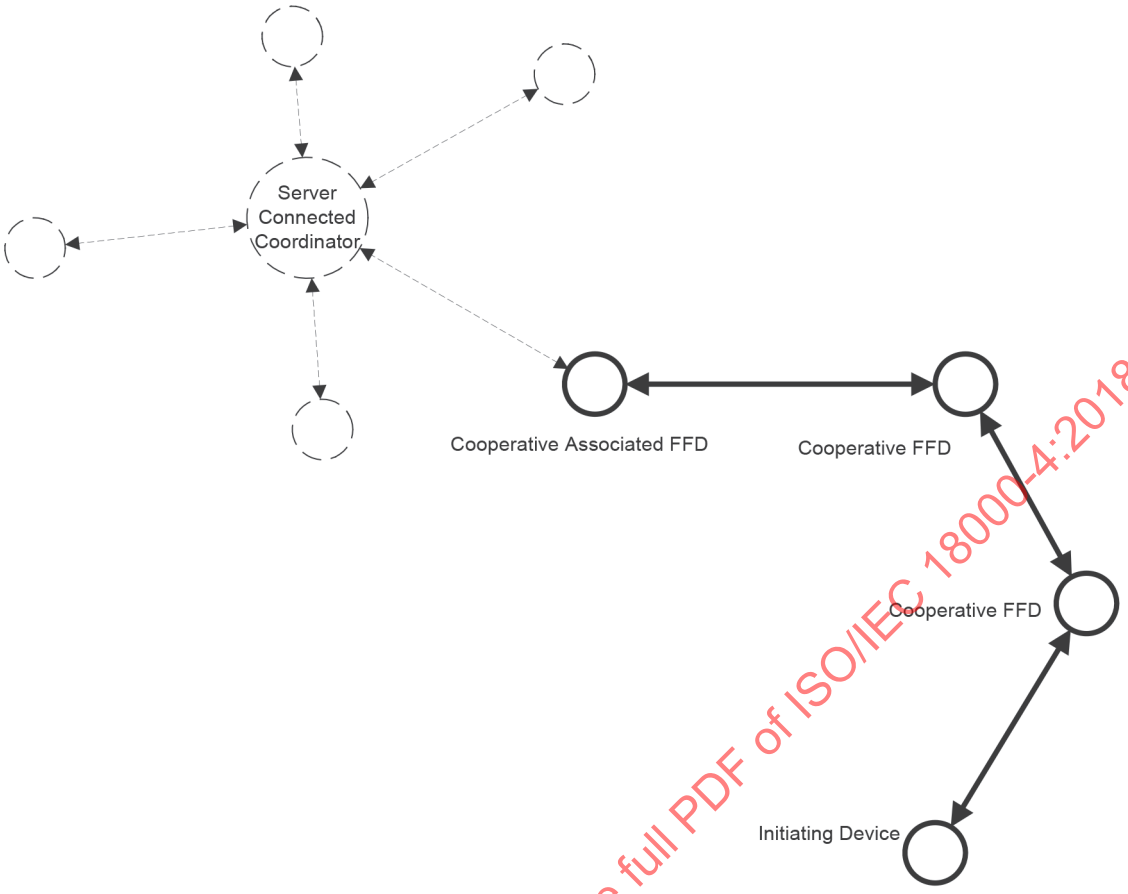


Figure 25 — Mesh topology

8.5 Star topology

8.5.1 General

All devices shall comply with a star topology requirement.

There are two types of network connections for star topology networks.

Short-term connections: The network supports rapid network discovery and data transmission protocols. Network access is resolved in milliseconds. Status data can be transferred to a device specified server within 1 s.

Maintained connections: Network connections associate a remote device with a network coordinator using negotiated message exchanges. The association process authenticates and authorizes a connection. An associated connection speeds data uploads and may be link encrypted for security or private operation within local area networks.

The connection type (short term or maintained) is determined by the remote device. Remote devices request association during the network discovery process. A remote device may use the SCC MAC address to identify whether a short-term or maintained network is appropriate.

All devices messages shall include the IEEE EUI 64 address and Manufacturing Identification Number (MIN) as unique identifier.

8.5.2 Star topology data flow

In a star topology, the network coordinator is an SCC. The network coordinator shall emit a series of periodic NDB packets to initiate network connections. When a device receives an NDB, it responds with a network status message (NSM). The NSM contains information about the remote server for the device's data. All data sent from this device may be forwarded to the indicated remote server.

A network coordinator may be secured. In this case, if the network coordinator is secure, a device may change behaviour if it is in the vicinity of a secured network coordinator (secured location). To facilitate reliable radio communication with less interference, multiple channels may be used. The NDB will be broadcast on what is referred to as a hailing channel. In order to reduce power usage from scanning many channels, this document defines four channels for NDB transmission. These channels are 15, 17, 21, and 23. In the NDB data, an alternate network channel may be specified. The alternate network channel may be less busy and may provide higher reliability with a channel not utilized for network discovery.

8.6 Trunk topology

All devices shall comply with a trunk topology requirement.

A device operates in a trunk network in the same manner as a star topology network.

In a trunk configuration, there are two or more coordinators in which each coordinator operates in its own star network topology. There is only one coordinator that is an SCC.

Trunk communications extend the radio range of a network by relaying radio traffic between coordinators. This reduces cost of additional network infrastructure or allows additional infrastructure to be installed at remote locations.

The SCC operates in a manner that provides association services to additional trunk coordinators; see ANC (see 8.10.4).

8.6.1 Trunk coordinator requirements

Before a trunk coordinator may operate in a star network, it shall first associate with an SCC or secondary trunk coordinator. A trunk coordinator shall associate as described in associated network connection (ANC).

If the trunk coordinator is able to associate with the SCC, it will immediately send an NSM to its SCC. This NSM shall indicate that it wishes to associate as a trunk coordinator. Any trunk coordinator associated with an SCC shall broadcast its own NDB. Additional trunk coordinators may request an association with that trunk coordinator. The same type of association request MAC command and NSM shall be used for this secondary and any additionally chained trunk coordinator networks.

8.6.2 Data flow in a trunk topology

All data from an associated device are sent to its coordinator. If that coordinator is a trunk coordinator, the trunk coordinator shall forward the device packet to that trunk coordinator's coordinator address and the data shall be passed to the next trunk coordinator until the SCC receives the data. The source address shall remain unchanged throughout this process.

When an SCC receives data from its server, it shall check the destination address of that data for a matching address from its list of associated devices. If the destination is an associated device, the message is sent to that device; otherwise, the message is forwarded to all trunk coordinators that are associated. The source address shall be that of the SCC. The receiving associated trunk coordinators shall change the source address from the parent coordinator to their own address and send the message to an associated device if a matching address is found or send the message to all associated trunk coordinators if no matching associated device is found. The message will either get to the intended destination address or it will terminate at any number of trunk coordinators that are not serving any

associated trunk coordinators. Any trunk coordinator without a matching destination address and without any associated trunk coordinators shall discard the message.

8.7 Peer-to-peer topology

All devices shall operate in a peer-to-peer topology.

In a peer-to-peer topology, messages may be passed between two devices where neither is acting as a network coordinator. This topology operates independently from the other network types. An example is a device operating with a secondary device (e.g. device to handheld). One of the two devices initiates the communication by sending a point-to-point NDB message with the destination address set to the peer that it wishes to communicate with. The command waiting option shall be used with the destination MAC address inserted in the location as specified in the NDB message description (see 8.9.1).

The point-to-point NDB message is used to communicate with a known device and avoids the network association process. It is expected that the radio with the destination MAC address should respond with a data request MAC command as specified in ISO/IEC/IEEE 8802-15-4. The data request MAC as specified in ISO/IEC/IEEE 8802-15-4 command shall use 64-bit long addressing for both source and destination addresses.

Upon receipt of the data request command, the device that initiated the NDB message shall send the intended message knowing that the destination device is ready to receive a message. The two devices shall continue sending messages to each other until finished.

This type of communications is depicted in Figure 26.

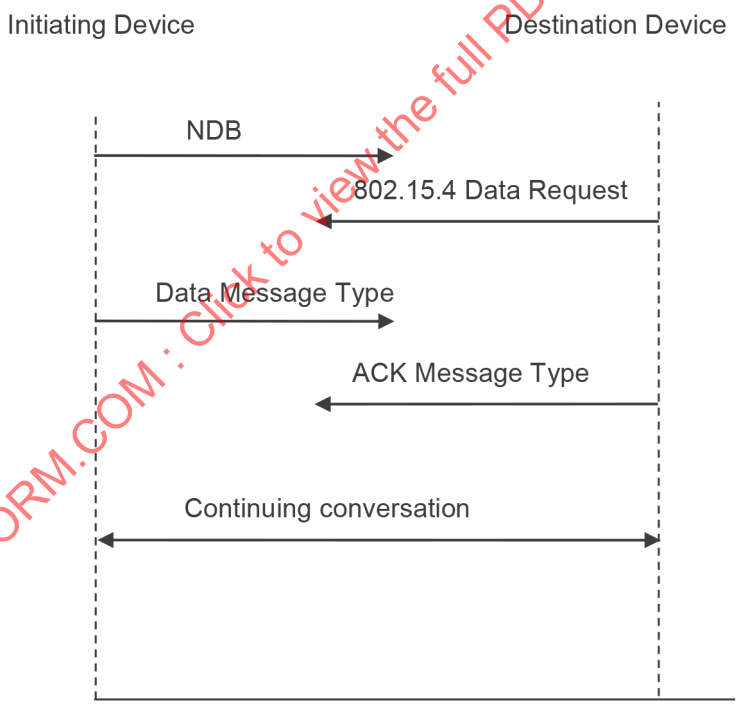


Figure 26 — Network access for peer-to-peer topology

8.8 Mesh topology

A mesh network is a method using cooperative FFDs to communicate between a network coordinator and a non-associated device. The primary reason to establish a mesh network is for a device to report a message or for a coordinator to query a non-associated device.

A cooperative FFD is configured to relay messages, device to device. The message may terminate at either the SCC or the targeted end-point device. A device is not required to be a cooperative device. The mesh network shall be established using beacon request command frames as specified in ISO/IEC/IEEE 8802-15-4 and beacon frames as specified in ISO/IEC/IEEE 8802-15-4.

Cooperative FFDs shall maintain an active receiver for a minimum of 5 s to ensure message reliability between communicating devices after receiving a beacon request.

8.8.1 Establishing a mesh network

8.8.1.1 General

Either a device or a coordinator may initiate a mesh network. A remote device establishes a mesh network by initiating the path discovery process as described in 8.8.1.2. The coordinator establishes a mesh network by broadcasting a mesh request command to the target device to initiate the path discovery process as described in 8.8.1.3.

A failed mesh networking attempt may not result in restarting the initiation process immediately. An optimal approach would be to restart the initiation process after there is evidence that there has been a network configuration change, e.g. detected tag motion, increased receiver signal strength, or detection of a new tag ID in the mesh architecture.

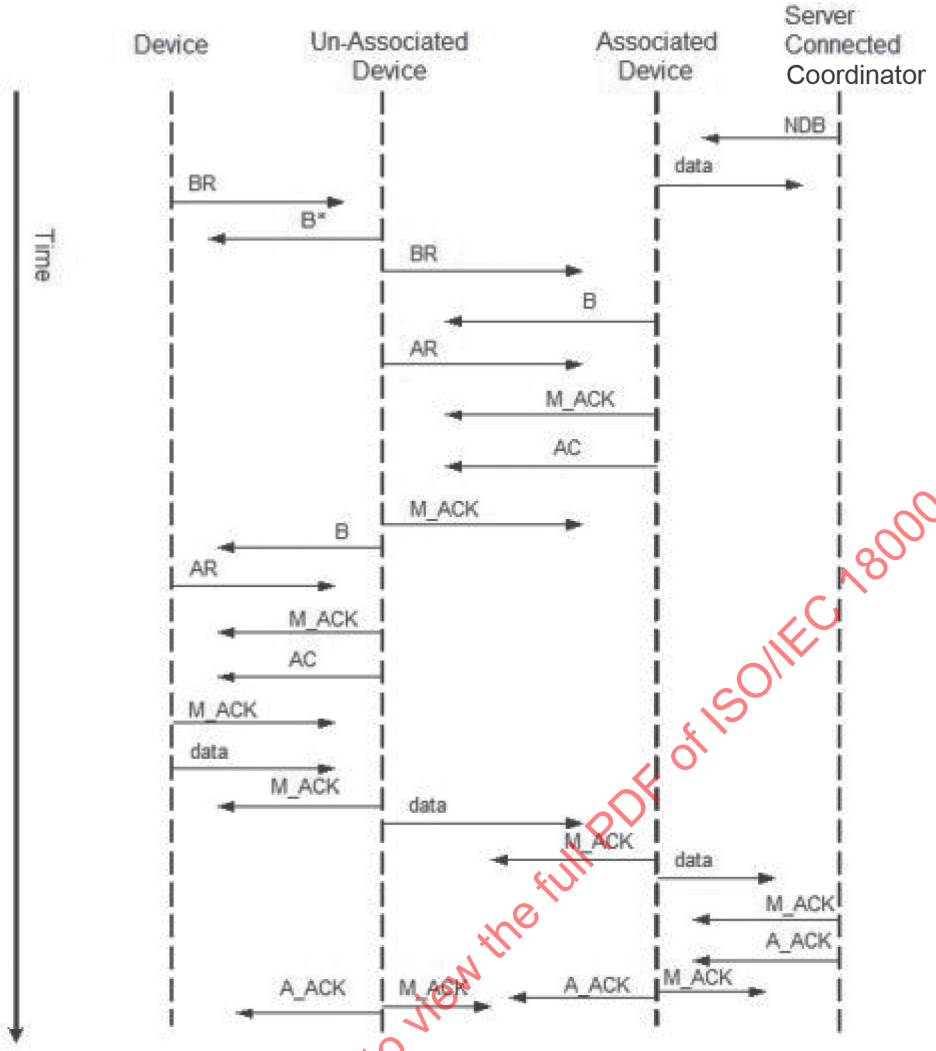
8.8.1.2 Mesh network initiated by remote device

A mesh network may be initiated by a non-associated device that needs to report a message.

To send a message, the device shall initiate a mesh network by periodically transmitting an ISO/IEC/IEEE 8802-15-4 beacon request command (see Figure 27). When any cooperative FFD receives a beacon request, it shall respond by transmitting its own beacon. If the cooperative FFD is associated, then the association-permit bit in the super-frame specification field of the beacon frame shall indicate that associations are permitted. If the cooperative FFD is not associated, the association-permit bit shall indicate that associations are not permitted.

Non-associated cooperative FFDs shall attempt to associate by issuing a beacon request to other devices that may be in range. To prevent excessive transmission attempts, a timeout of 5 s after receiving a beacon request shall be maintained.

An associated cooperative FFD that receives a beacon request shall respond by broadcasting a beacon with the association-permit bit enabled. All cooperative FFDs that receive that beacon shall attempt to associate. If successful, the associated cooperative FFD transmits a beacon indicating that it will accept associations. This process continues until the initiating device is able to associate. Once the initiating device is associated, it transmits data to the cooperative FFD coordinator. The data are sent using a mesh data message type. The end-point ISO/IEC/IEEE 8802-15-4 MAC address in the mesh data message is set to the initiating remote device's MAC address. The cooperative FFD coordinator subsequently forwards the message to its coordinator until it reaches the network coordinator. As the message is retransmitted, each FFD substitutes the ISO/IEC/IEEE 8802-15-4 MAC header information for the communications between itself and its coordinator. The coordinator receives the mesh data for an associated device and knows the originator's source address from the embedded originator ISO/IEC/IEEE 8802-15-4 MAC address field within the mesh data message. Every cooperative FFD associated within the mesh that receives a message for the initiating device shall forward that message.



Key

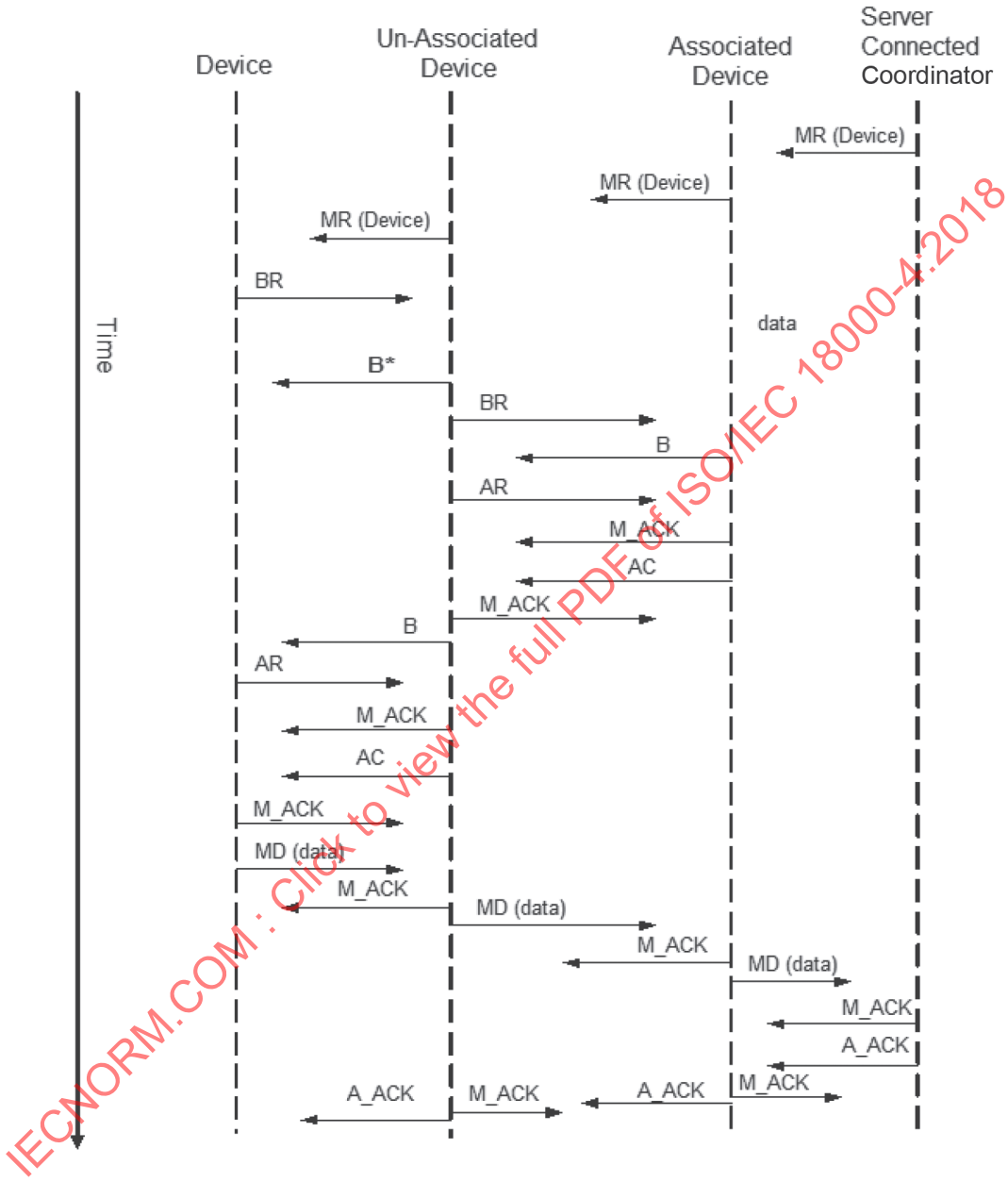
NDB	Network Discovery Beacon
B	802.15.4 Beacon (Association Bit Set)
B*	802.15.4 Beacon (Association Bit Not-Set)
AR	802.15.4 Association Request
AC	802.15.4 Association Command
M_ACK	802.15.4 Acknowledgement
A_ACK	Application Acknowledgement

Figure 27 — Mesh network initiated by remote device

8.8.1.3 Mesh network initiated by coordinator

Figure 28 depicts the process a coordinator shall use to establish a mesh network with an unassociated device. To communicate with the remote device, the coordinator shall broadcast a mesh request command to the device's MAC address. All cooperative FFDs shall rebroadcast the mesh request command. ISO/IEC/IEEE 8802-15-4 MAC layer acknowledgement frames shall not be transmitted in response to this command. Once the targeted device receives the forwarded command, it will initiate the path discovery method described in the previous subclause. Once the path discovery is complete, the targeted device transmits an acknowledgement message type to inform the coordinator that the mesh has been established.

The coordinator shall use the mesh data message type to transmit data to a remote device in a mesh network. The coordinator shall insert the targeted remote device's ISO/IEC/IEEE 8802-15-4 MAC address in the end-point ISO/IEC/IEEE 8802-15-4 MAC address section of the mesh data message. Each cooperative FFD established during the path discovery step shall forward this type of message to their associated device(s).



Key

MR	Mesh Request Command
B	802.15.4 Beacon (Association Bit Set)
B*	802.15.4 Beacon (Association Bit Not-Set)
AR	802.15.4 Association Request
AC	802.15.4 Association Command
M_ACK	802.15.4 Acknowledgement
A_ACK	Application Acknowledgement
MD	Mesh Data Packet (source address appended)

Figure 28 — Mesh network initiated by coordinator

8.9 Message types

This document defines message types that ensure interoperability between a diverse set of devices and manufacturers. Within message type categories, devices may send user data that are routed using this document. Although messages are not limited by the use of this document, these messages shall meet the format for message types described herein if they are to join the network and be accurately delivered.

The message header implemented by this document shall meet the message header specified in ISO/IEC/IEEE 8802-15-4. This clause and its subclauses specify message types and formats to implement: 8.9.1: network discovery beacon (NDB), 8.9.2: network status message (NSM), 8.9.3: acknowledgement message, 8.9.4: command message, 8.9.5: data message, 8.9.6: mesh request, and 8.9.7: mesh data.

The first two octets of the payload shall incorporate the correct device type and message type fields and follow the specified format identified in Table 118 and Table 119. The data message is variable depending on the message type. See Figure 29 and the subclauses of 8.9 for details.

The message types ACK, NSM, and NDB are provided for network administration and maintenance and therefore the payload data definition of each of these message types is defined in this document.

The command message format and data message format are provided for manufacturer defined payload data to be sent and received between the remote device and a data collection end-point.

To ensure data integrity, the payload of all packets shall be terminated with a two octet CRC. The two octet CRC shall be calculated over the entire payload using the 16-bit CCITT algorithm with an initial value of 0xffff.

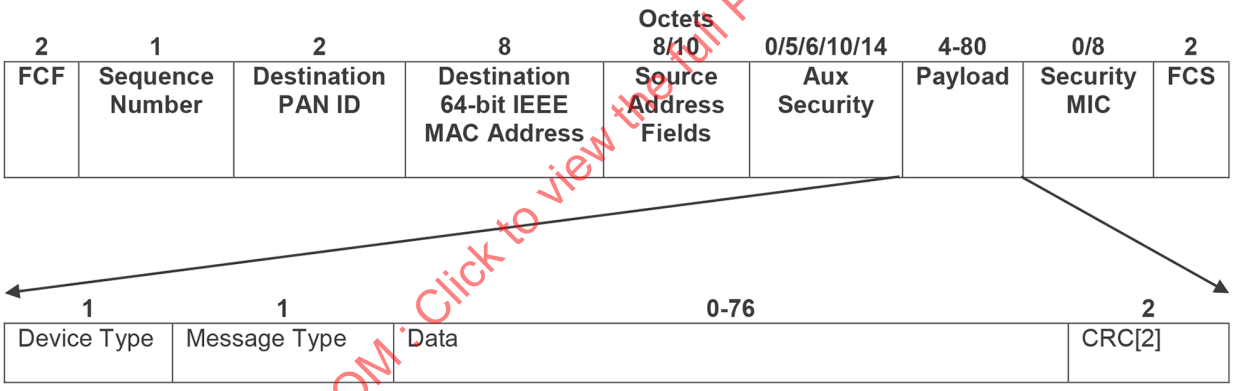


Figure 29 — General description of data packet showing payload section

Table 118 summarizes a list of reserved device types specific to this network. The device type shall be included in the header information of all transmissions.

Table 118 — Defined device types

Device type	Device type description
0x00	Unspecified
0x01-0x3F	FFD
0x40-0x7F	RFD
0x80-0xFF	Reserved (non-commercial)

The device type field is a single octet in every message payload that identifies the device type to a receiving radio.

Table 119 — Message type octet format

Message type	Message type description	Mandatory
0x00	Command (implements command pending NDB) (see 8.9.4)	
0x01	Data (status, tracking, stored data, etc.) (see 8.9.5)	Yes
0x02	ACK status (or NAK) (see 8.9.2.2.1.3)	Yes
0x7D	Mesh request (see 8.9.6)	
0x7E	Mesh data (see 8.9.7)	
0x7F	Network status message (see 8.9.2)	Yes
0x80 – 0xFE	Reserved	
0xFF	Network discovery beacon (see 8.9.1)	For FFD devices only

The data packet will set the Frame Type subfield in the FCF byte to 1 for Data.

8.9.1 Network discovery beacon (NDB)

The NDB shall be broadcast by a network coordinator to announce a network is available. This network utilizes a network discovery method that may respond within 1 s and support receiver duty cycles <2,0 %.

See Figure 30, Figure 32, Table 120 and Table 121 for details.

NDB payloads provide

— alternate radio channels for network traffic management,

— broadcast and response interval information for power management purposes, and

— optional data, e.g. date and time, name and location of network coordinator, GPS location, differential GPS, and encrypted data for proprietary use.

NDB timing

— NDBs are emitted periodically to announce network availability.

— NDBs are emitted at a high rate for a 1 s period to allow remote devices to support deep duty cycles.

— NDBs are emitted on ISO/IEC/IEEE 8802-15-4 channels 15, 17, 21 and 23.

— NDB payloads may redirect devices to an alternate channel based on network load.

The peer-to-peer NDB data frame description as per Figure 31.

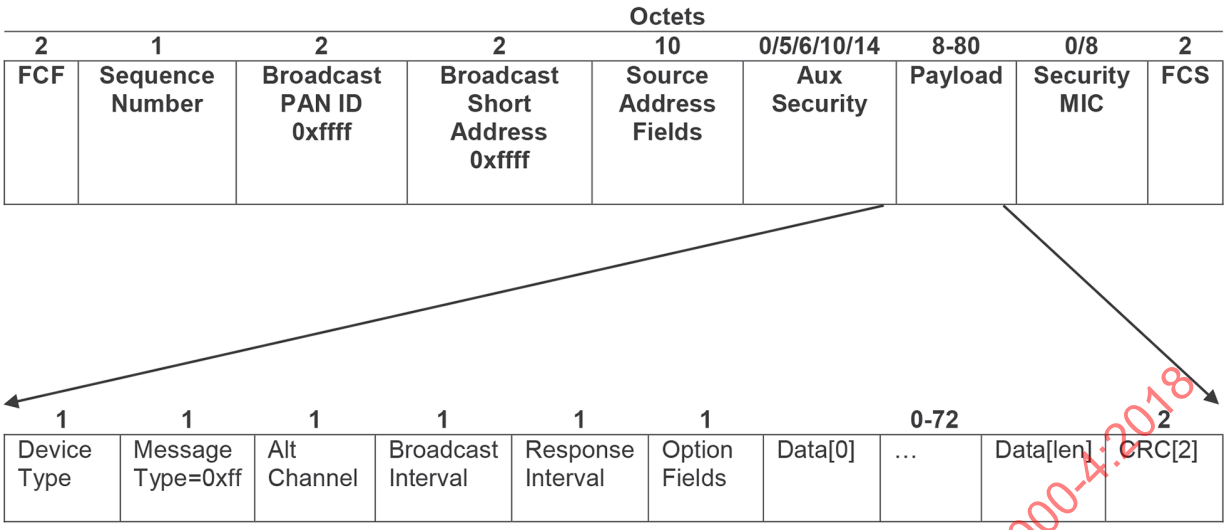


Figure 30 — Broadcast NDB data frame description

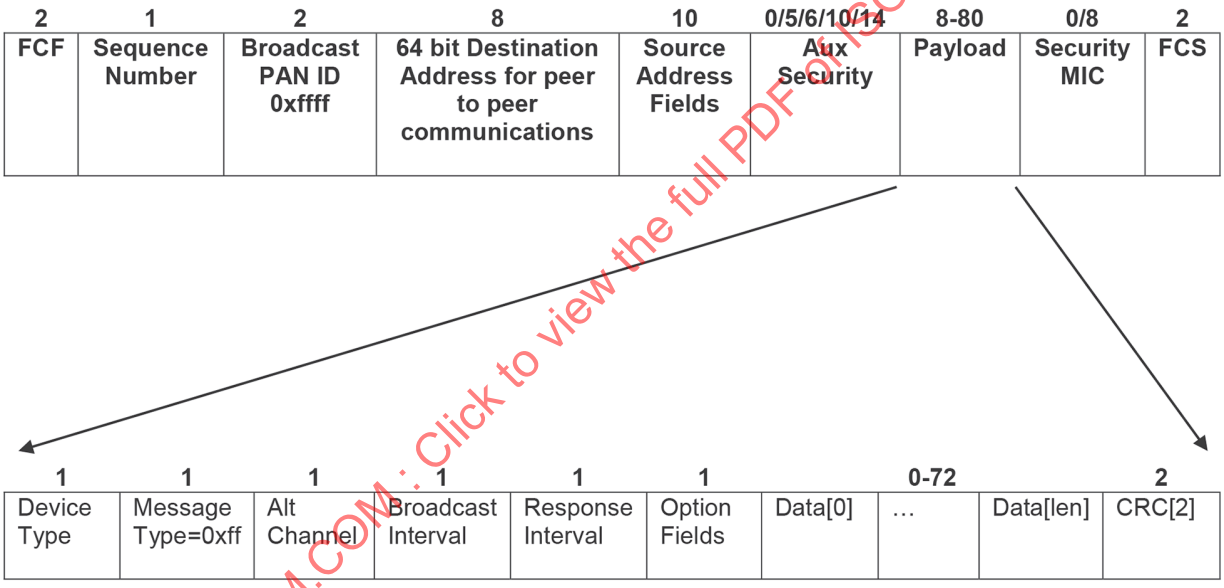


Figure 31 — Peer-to-peer NDB data frame description

Table 120 — NDB data frame payload description

Byte name	Byte definition
Device type	Identifies device type of NDB sender (see Table 118)
Message type	0xff indicates this packet is a network discovery packet
Alt channel	Identifies alternate channel for network association (other than hailing channel)
Broadcast interval	Integer value in seconds between each broadcast period (or 1 s burst of NDB frames)
Response interval	Integer value of NDB burst (see Figure 38) that a tag receives before it shall communicate status to the network coordinator
Option fields	See Table 122

Table 120 *(continued)*

Byte name	Byte definition
Data[0]	Option payload data length including the 2 byte CRC
Data[1]-[71]	Option payload data content
CRC[0][1]	Two octet CRC CCITT (0xffff)

Table 121 — Bit fields of the alternate channel octet

Bit 7	6	5	4-0
Alternate channel use mandatory	Network coordinator is accepting associations	RFU	Channel value

8.9.1.1 Alternate channel

8.9.1.1.1 General

The bit fields of the alternate channel octet are defined in the following subclauses and shown in Table 121.

8.9.1.1.2 Bit 7 most significant bit (MSB)

If channel is different from current channel, use alternate channel for association requests and data transmissions.

— 1: Indicates alternate channel is mandatory for association requests and data.

— 0: Current channel receives association requests and data. Alternate channel is optional.

8.9.1.1.3 Bit 6

— 1: Indicates the network coordinator is accepting associations.

— 0: Indicates the network coordinator is not accepting associations.

8.9.1.1.4 Bit 5

— Not used. Shall be set to 0.

8.9.1.1.5 Bits 0 to 4

— Integer value (lower 5 bits) representing alternate ISO/IEC/IEEE 8802-15-4 channel.

— Valid channel values: 11 to 26.

8.9.1.1.6 An example of the alternate channel octet

An octet with the value of 0x51 (0101 0001b) specifies the following:

— Network coordinator specifies alternate channel 17.

— Alternate channel is not mandatory.

— Network coordinator is accepting associations.

8.9.1.2 Broadcast interval

The broadcast interval octet defines the interval in seconds between NDB frames.

It has the following range of values:

— 0 = continuous NDB transmissions;

— 1 to 254: 1 s to 254 s between NDB frame transmissions;

— 255: no NDB retransmissions.

The typical values are:

— long range network coordinator (800 m to 1 000 m): 60 = 60 s;

— short range network coordinator (50 m to 100 m): 15 = 15 s.

8.9.1.3 Response interval

The response interval octet defines the interval in NDB bursts that a tag should provide unsolicited status to remain connected. This reduces power consumed by the tag by responding with data at a network specified interval and randomizes and distributes data traffic based on timing of network association.

The range of values is:

— 0 = never respond without request;

— 1 to 255: respond every (1 to 255) NDB bursts.

8.9.1.4 Option fields and data

The option field description is shown in Table 122. The subsequent subclauses provide the specifications for each payload type.

Table 122 — Option field description

Payload type	Data description
0x00	No NDB payload data
0x81	Network coordinator date and time stamp
0x82	Network coordinator identification – ASCII string
0x84	Network coordinator location
0x88	Network coordinator date and time + network coordinator ID + network coordinator location
0x90	Reserved
0xff	Commands pending
All others	Reserved for future use

8.9.1.4.1 0x81 Network coordinator date and time stamp (7 octets)

This has the format MM/DD/YY Wd HH:MM:SS (GMT) where:

— MM: month 1 to 12;

— DD: day 1 to 31;

— YY: year 0 to 99;

— Wd: weekday 0 to 6;

— HH: 0 to 23;

— MM: 0 to 59;

— SS: 0 to 59.

EXAMPLE

12/4/9 5 12:30:20 is represented as option code 0x81, length of 7, and 7 data octets:

0x81 0x09 0x0C 0x04 0x09 0x05 0x0C 0x1E 0x14 <CRC[0]> <CRC[1]>

8.9.1.4.2 0x82 Network coordinator identification (20 octets)

User definable: (pad with ASCII blank space: 0x20).

EXAMPLE

"GEO US LA East Gate" represented in ASCII:

0x82 0x16 0x47 0x45 0x4f 0x20 0x55 0x52 0x20 0x4c 0x41 0x20 0x45 0x61 0x73 0x74 0x20 0x47 0x61 0x74 0x65 0x20 <CRC[0]> <CRC[1]>

8.9.1.4.3 0x84 Network coordinator location (12 octets)

Single precision IEEE floating point representation.

Latitude (degree), longitude (degree), altitude (meters).

EXAMPLE

37,2744621276855 –121,985816955566 –24,6905689239502

0x84 0x0E 0x74 0x7F 0x15 0x42 0xBD 0xF8 0xF3 0xC2 0x49 0x86 0xC5 0xC1 <CRC[0]> <CRC[1]>

8.9.1.4.4 0x88 Network coordinator date and time stamp, network coordinator identification, Network coordinator location (39 octets)

Combined network coordinator date and time stamp, network coordinator identification, network coordinator location information.

EXAMPLE

12/4/9 5 12:30:20

"GEO US LA East Gate"

37,2744621276855 –121,985816955566 –24,6905689239502

0x88 0x29 0x0C 0x04 0x09 0x05 0x0C 0x1E 0x14 0x47 0x45 0x4f 0x20 0x55 0x52 0x20 0x4c 0x41 0x20 0x45 0x61 0x73 0x74 0x20 0x47 0x61 0x74 0x65 0x20 0x74 0x7F 0x15 0x42 0xBD 0xF8 0xF3 0xC2 0x49 0x86 0xC5 0xC1 <CRC[0]> <CRC[1]>

8.9.1.4.5 0xff Commands pending (multiple of 8 octets)

Data contain one or more ISO/IEC/IEEE 8802-15-4 64-bit MAC addresses.

Number of addresses indicated by Data[0] which is the data length.

Indicated device may send an ISO/IEC/IEEE 8802-15-4 data request MAC command to retrieve pending commands.

EXAMPLE

Data pending for MAC addresses 63.64.65.66.01.02.03.04 and 70.71.72.73.01.02.03.04

0xFF 0x12 0x63 0x64 0x65 0x66 0x01 0x02 0x03 0x04 0x70 0x71 0x72 0x73 0x01 0x02 0x03 0x04 <CRC[0]> <CRC[1]>

8.9.1.4.6 0x00 No NDB payload present

Set the option field to 0x00 to indicate no NDB payload present.

8.9.1.5 NDB CRC (cyclical redundancy check)

The NDB CRC strengthens ISO/IEC/IEEE 8802-15-4 error detection.

8.9.2 Network status message (NSM)

The network status message (NSM) is a packet transmitted by a remote device requesting network access. The NSM is sent to the destination PAN and MAC address received in the NDB packet.

The NSM provides information to negotiate and route data on the network.

The NSM provides information on power saving parameters and network coordinator acknowledgement requests.

The NSM shall be forwarded to the Device Server indicated in the NSM and in the manner indicated by the Device Server Connection Method in the NSM.

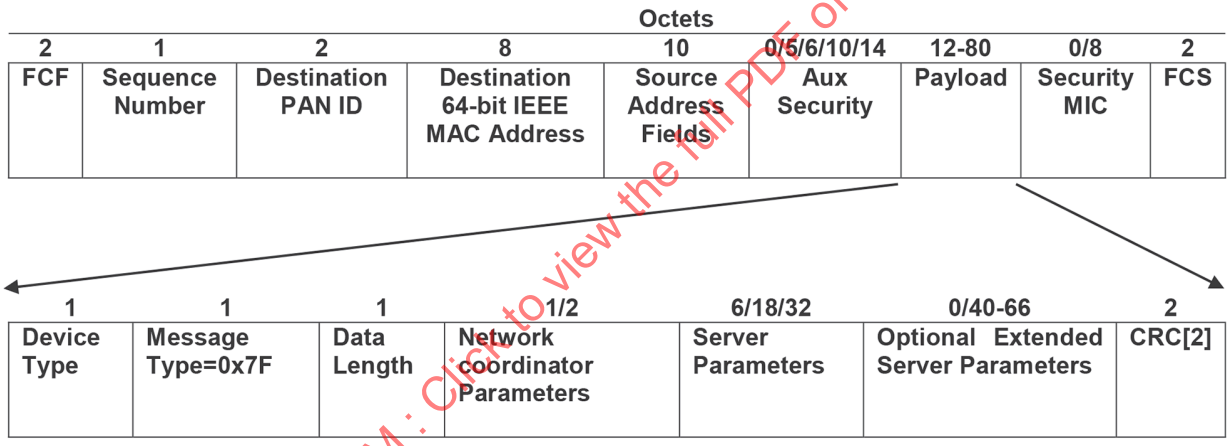


Figure 32 — Network status message packet description

The NDB packet will set the Frame Type subfield in the FCF byte to 1 for Data.

The Network status message payload description is specified in Table 123.

Table 123 — Network status message payload description

Name	Description
Device type	Identifies the sender device type as defined in this document
Message type	Identifies the message type as defined in this document. 0x7F for this message.
Data length	Total number of octets to follow up to and including the 2 octet CRC
Network coordinator parameters	Parameters affecting network coordinator to device communications

Table 123 *(continued)*

Name	Description
Server parameters	Parameters for routing data between joining device and device's server
Optional extended server parameters	User defined data between joining device and device's server
CRC[2]	CRC CCITT (0xffff)

8.9.2.1 Network coordinator parameters

The joining device provides these parameters to the network coordinator to improve power management and data transmission speed while connected to the network. These parameters may be unique for each joining device.

The network coordinator parameters consist of a one octet option field and an optional sleep value octet.

8.9.2.1.1 Option field

The one octet option field is organized as a series of bit flags as shown in Table 124.

Table 124 — One octet option field

Bit 7 MSB	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 LSB
RFU	RFU	RFU	RFU	RFU	Trunk coordinator configuration	Network coordinator acknowledgement request	Sleep value included

8.9.2.1.1.1 Sleep value included (bit 0)

0: Sleep value not included as next octet.

1: Sleep value is included in the next octet.

8.9.2.1.1.2 Network coordinator acknowledgement request (bit 1)

This parameter requests that the network coordinator reply with an acknowledge data packet with every data message received.

This method is recommended to accelerate data transmission rather than wait for an application acknowledgement from the remote server. When requested, the network coordinator shall send an acknowledgement message indicating successful receipt of the device's packet.

0: Remote device does not request ACK response message.

1: Remote device requests an ACK message response for each data packet sent to the network coordinator.

See 8.9.2.2.1.3 for information on the acknowledgement message format.

8.9.2.1.1.3 Trunk coordinator configuration (bit 2)

0: Remote device is not configured as a trunk coordinator.

1: Remote device is configured as a trunk coordinator.

8.9.2.1.2 Sleep value (1 octet)

If provided in the NSM, the network coordinator shall use the sleep value parameter to predict sleep intervals when sending commands.

Number of seconds remote device will leave its receiver on before going to power savings mode. Used for commanding.

— 0: Commands pending NDB/data request exchange shall always be used to command.

— 255: Receiver always active.

— All others: receiver will be active for sleep value seconds after each ISO/IEC/IEEE 8802-15-4 radio exchange. Network coordinator may command at will if it is within sleep value seconds.

8.9.2.2 Server parameters

The server parameters fields in the NSM payload are located after the network coordinator parameters. These parameters provide data for the SCC server to route device data messages and manage the network.

These parameters are provided by the joining device to allow the network coordinator to route data to the joining device's server.

8.9.2.2.1 Server options (1 octet)

Table 125 shows the overview of the server options.

Table 125 — Server options

Bit 7 MSB	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 LSB
Link key request	Device server TCP/IP	Device server UDP	Device server Email	RFU	RFU	RFU	IP address Address specification

8.9.2.2.1.1 IP address specification (bit 0)

0: IPv4 address follows.

1: IPv6 address follows.

8.9.2.2.1.2 Device Server Connection Method (bits 4 to 6)

0x10: Email.

0x20: UDP.

0x40: TCP/IP.

8.9.2.2.1.3 Link key request (bit 7)

This bit is an indication to the device server that the device is requesting a link layer encryption key for use between the coordinator and the device.

If the Device Server Connection Method is TCP/IP, this value may be set to 1. This value shall be set to 0 for all other connection methods.

See 8.11 for more information.

All others reserved for future use.

8.9.2.2.2 Server address (6, 18, or 32 octets)

This field is the joining device's server address for forwarding data to the device's server and may be different from the network coordinator's server IP address.

8.9.2.2.2.1 Email Device Server Connection Method

When specified Device Server Connection Method is Email, this field is the Email recipient (32 octets interpreted as ASCII) – fill unused octets with 0x00.

8.9.2.2.2.2 UDP or TCP/IP Device Server Connection Method

When specified Device Server Connection Method is either UDP or TCP/IP, this field is the IP address followed by the port number.

IPv4 address followed by server port number (6 octets).

192.168.1.200/6200 represented in 6 octets as C0 A8 01 C8 18 38. C8 in binary form is 11001000.

IPv6 address followed by server port number (18 octets).

See IPv4 example for binary representation.

8.9.2.3 Optional extended server parameters

The optional extended server parameters field in the NSM payload is located after the server parameters field. The maximum length of this field may vary from 40 octets to 66 octets. The variable maximum octet length is calculated from a maximum payload length of 80 octets minus the length of all preceding fields in the NSM. These parameters may be used to pass manufacturer specific information from the device to the device server.

8.9.2.3.1 NSM Key Exchange

When the link key request bit is set in the NSM server parameters octet, the device server may send the NSM back to the coordinator. The NSM is the same message that was received with the optional extended server parameters field overwritten with the security level identifier in the first octet followed by the 16 octet key if appropriate. The data length and CRC octets of the original NSM message shall be recalculated and the recalculated length and CRC values replace the original values before being sent to the coordinator.

Table 126 indicates the key requirement for each security level identifier as defined in ISO/IEC/IEEE 8802-15-4:2018, Table 9-6.

Table 126 — Security levels and key requirement

Security level identifier	Security attribute	Key required
0x00	None	No
0x01	MIC-32	No
0x02	MIC-64	No
0x03	MIC-128	No
0x04	ENC	Yes
0x05	ENC-MIC-32	Yes
0x06	ENC-MIC-64	Yes
0x07	ENC-MIC-128	Yes

8.9.3 Acknowledgement message

The acknowledgement message is transmitted in an ISO/IEC/IEEE 8802-15-4 data frame. This acknowledgement should not be confused with the ISO/IEC/IEEE 8802-15-4 acknowledgement frame.

The acknowledgement message shall be sent from a device to confirm receipt of a command message.

If the device indicates that a request for data acknowledgement is required in its NSM, then an acknowledgement message shall be sent from the connected data server to a remote device upon receipt of all data messages.

The acknowledgement message may also be used to issue a no-acknowledgement reply (NAK). A no-acknowledge (NAK) may indicate a specific error type. The sender may determine the appropriate response based on the received NAK.

If link encryption is used, the acknowledgement message shall be encrypted using parameters specified in the NSM message.

8.9.3.1 Acknowledgement message format

The application ACK/NAK packet format as per Figure 33.

The application ACK/NAK packet payload format is specified in Table 127.

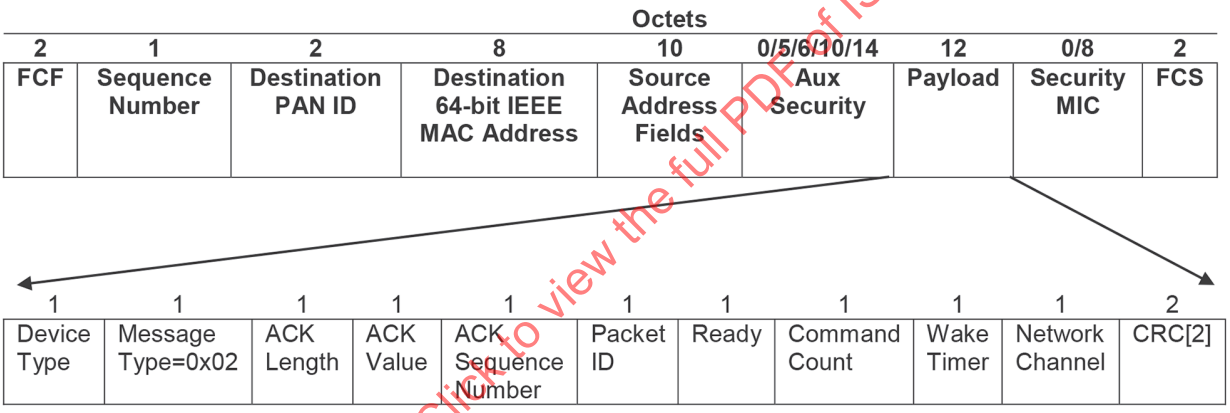


Figure 33 — Application ACK/NAK packet format

Table 127 — Application ACK/NAK packet payload format

Device type	Identifies device type of ACK sender
Message type	0x02 indicates this is an acknowledgement packet
ACK length	Length of remaining data in ACK packet
ACK value	0x02: successful ACK 0x00: NAK/error
ACK sequence number	ISO/IEC/IEEE 8802-15-4 frame sequence number that is being ACK'd from recipient (provided by recipient)
Packet ID	Packet ID that is ACK'd Or if NAK, the error type See 8.9.3.1.2
Ready	Ready to receive more data if 0; otherwise, delay ready seconds or until an identical ACK with Ready = 0 is received
Command count	Incrementing counter indicating sequence of processed packets

Table 127 *(continued)*

Wake timer	Seconds that device keeps its receiver continuously on after transmitting this ACK Used for commanding; network coordinator can send another command if transmission is within this time window. Otherwise, network coordinator may have to use commands pending field of NDB.
Network channel	Current network channel that the sending device utilizes for data transfers
CRC[2]	CRC CCITT (0xffff)

8.9.3.1.1 ACK sequence number

The ACK sequence number octet is the ISO/IEC/IEEE 8802-15-4 sequence number of the data frame that the sender is acknowledging.

EXAMPLE Device A sends a data frame to device B. The data frame from device A contains the ISO/IEC/IEEE 8802-15-4 sequence number 0x53. The acknowledgement message from device B to device A will contain the value 0x53 for the ACK sequence number.

This value may be used by device A to confirm that the acknowledgement message from device B is for a particular data packet that device A previously sent.

8.9.3.1.2 Packet ID

8.9.3.1.2.1 Successful ACK

For an ACK value indicating a successful acknowledge, this octet, along with the ACK sequence number, may be used by the receiver to reliably confirm acknowledgement for a previously sent packet.

Acknowledgement for a command message

If the acknowledgement message is in response to a command message, this value shall be the octet command[0] from the command message.

Acknowledgement for a data message

If the acknowledgement message is in response to a data message, this value shall be the octet data[0] from the data message.

8.9.3.1.2.2 NAK/error

For an ACK value indicating a failed acknowledgement, this octet shall be set to one of the predefined indicators.

This indicator shall be used to improve reliability and reduce undesired radio traffic.

CRC error (0x01)

Indicates a CRC error for the received frame.

The device that sent the original message may immediately resend the message.

Encryption error (0x03)

Indicates that the receiver cannot access an encryption key for a received message that is encrypted or that the receiver will not accept a message that was sent unencrypted.

The device that sent the original message shall not attempt to resend the original message.

MIC error (0x05)

Indicates a mismatch between the received message integrity code and the calculated message integrity code.

The device that sent the original message may immediately resend the message.

If a second MIC error is received, the message shall not be resent.

Command error (0x07)

Indicates a bad command was received.

The device that sent the original message shall not attempt to resend the original message.

ISO/IEC/IEEE 8802-15-4 FCS error (0x09)

Indicates that there was an ISO/IEC/IEEE 8802-15-4 frame check sequence error in the received message.

The device that sent the original message may immediately resend the message.

8.9.3.1.3 Ready

A non-zero value indicates that the sender of the acknowledgement message cannot receive and process any further messages. The receiver of the acknowledgement message should wait the number of seconds indicated in this octet before sending any additional messages.

A subsequent acknowledgement message with a ready value of zero shall be sent when the sender of the original acknowledgement message is able to receive additional messages.

8.9.3.1.4 Command count

A one octet value that may be used by the device to indicate the number of received commands.

8.9.3.1.5 Wake timer

The same value as the sleep value in the NSM. This value may be used to update or change the original value sent in the NSM.

8.9.3.1.6 Network channel

A device may be associated with a network coordinator that has specified an alternate channel in its NDB. This value indicates the present alternate channel that the device uses for data transfers.

8.9.3.2 Proprietary acknowledgements

Users may implement proprietary ACK messages using the message type = 0x01 (data) and adding payload data specific to their requirements. Users who operate their devices using this method are responsible for the application level functions on both the joining device and SCC so that data flow control may be maintained with minimum retransmissions.

8.9.4 Command message

Message type 0x00 is intended for commanding remote devices.

The command message is transmitted in an ISO/IEC/IEEE 8802-15-4 data frame.

8.9.4.1 Commanding a device operating in a sleep duty cycle

If the remote device is operating in a sleep duty cycle, the network coordinator shall queue the command and send the device address with the NDB. When the destination device detects a command is pending,

the remote device sends a data request MAC command as specified in ISO/IEC/IEEE 8802-15-4:2018, 7.5.5 to retrieve the command.

The remote device shall send an acknowledge message in response to a command message.

For any additional commands, the network coordinator shall use the wake timer indicated in the acknowledgement message to determine if it can continue sending commands or if the device address shall be added in the NDB for each command.

8.9.4.2 Commanding a device operating with its receiver always active

If the device has associated and set the receiver-on-when-idle bit in the capability information field of the ISO/IEC/IEEE 8802-15-4 association request command, a network coordinator may send commands immediately.

8.9.4.3 Commanding in a mesh network

In a mesh network, this message type shall be interpreted as a command for the receiving device. A frame with this message type shall not be forwarded in a mesh network.

8.9.4.4 Command message format

The command packet format as per Figure 34.

The command packet payload format is specified in Table 128.

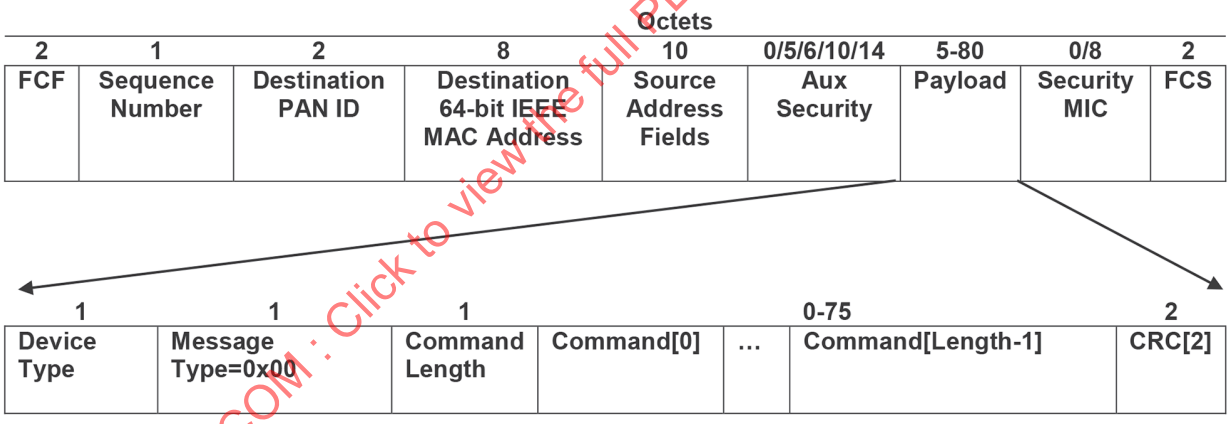


Figure 34 — Command packet format

Table 128 — Command packet payload format

Device type	Identifies device type of command sender
Message type	0x00 indicates this is a command packet
Command length	Length of command data
Command[0] -> Command [length-1]	Application specific command
CRC[2]	CRC CCITT (0xffff)

8.9.5 Data message

Message type 0x01 data packets are sent when remote devices are communicating with network coordinators that have receivers enabled for the duration of the expected communication.

The data packet length is included in the data payload because the ISO/IEC/IEEE 8802-15-4 header information may be stripped by the SCC prior to sending to the network server.

8.9.5.1 Data packets options

The content of data packets are defined by the user and are specific to device types and application. Examples of data that may be transmitted using data packets are the following:

— health and status data engineering data (battery status, timers, etc.);

— logged or historical network coordinator data;

— time tagged stored data and real time data during transit-reads.

8.9.5.2 Commercial proprietary data protection

Commercial proprietary data may be protected with encryption on a per trip basis. Data packets are generally sent to a known destination address assuming the receiving radio receiver is currently on.

8.9.5.3 Data message format

The data message format is shown in Figure 35.

The data packet payload format is specified in Table 129.

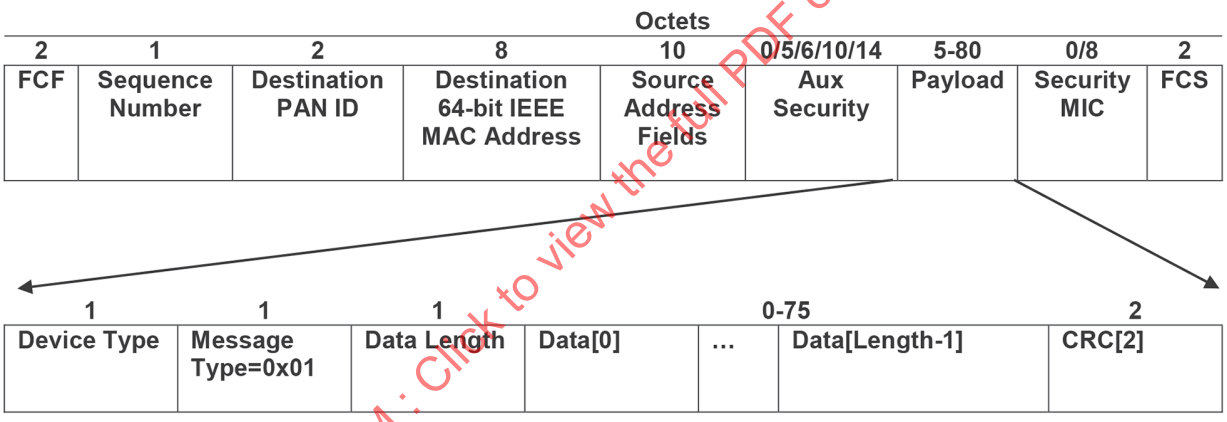


Figure 35 — Data message format

Table 129 — Data packet payload format

Device type	Identifies device type of data sender
Message type	0x01 indicates this is a data packet
Data length	Length of data content
Data[0] -> Data[len-1]	Application specific data
CRC[2]	CRC CCITT (0xffff)

8.9.6 Mesh request

Message type 0x7D data packets are sent when a coordinator initiates a communication network path with a non-associated device.

8.9.6.1 Mesh request message format

The mesh request packet format as per in Figure 36.

The data packet payload format is specified in Table 130.

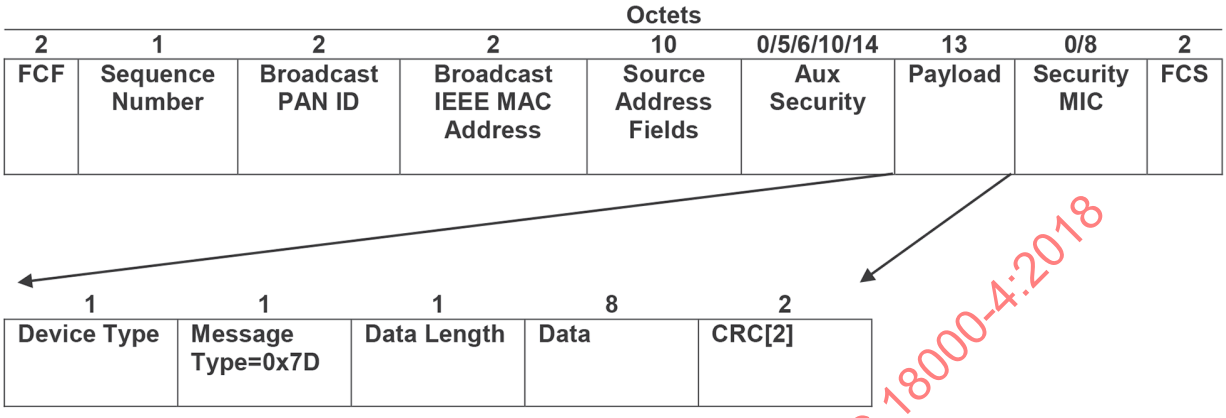


Figure 36 — Mesh request packet format

Table 130 — Data packet payload format

Device type	Identifies device type of data sender
Message type	0x7D indicates this is a mesh request packet
Data length	Length of data content
Data[0] -> Data[7]	MAC address as specified in ISO/IEC/IEEE 8802-15-4
CRC[2]	CRC CCITT (0xffff)

8.9.7 Mesh data

Message type 0x7E packets are used when sending data to a coordinator in a mesh network. To provide the source address of the sender, mesh data packets insert the source MAC address of the original sender in the payload section of the ISO/IEC/IEEE 8802-15-4 packet.

8.9.7.1 Mesh data message format

The mesh data message format as per Figure 37.

The data packet payload format is specified in Table 131.

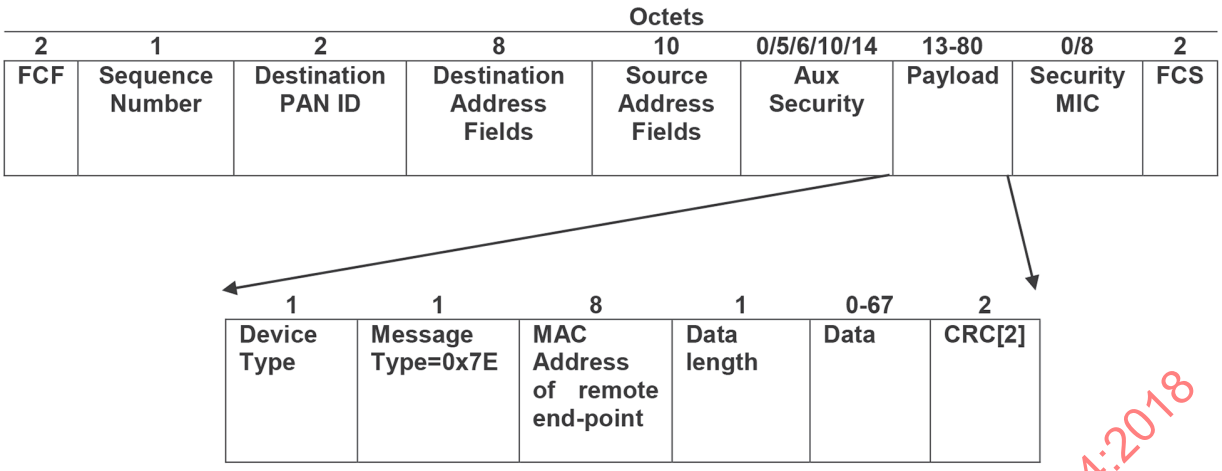


Figure 37 — Mesh data message format

Table 131 — Data packet payload format

Device type	Identifies device type of data sender
Message type	0x7E indicates this is a mesh data packet
MAC address of remote end-point	ISO/IEC/IEEE 8802-15-4 MAC address of the end-point remote device of the mesh, not the coordinator
Data length	Number of data bytes to follow
Data[0] -> Data[len-1]	User defined data
CRC[2]	CRC CCITT (0xffff)

8.10 Network discovery

This document identifies two methods for devices to join a network. Both methods broadcast network availability to remote devices by transmitting NDBs on a periodic basis. How often network discovery beacons are transmitted depends on the duration of time a device may be in range of the coordinator. The farther the communication range of a coordinator, the less often an NDB needs to be transmitted or if a device is traveling past a coordinator, the period between NDBs may need to be short to ensure adequate opportunities to read data from the device. The specific application dictates which network discovery method is used and how often the NDB packets are broadcast.

Any device attempting to operate as a network coordinator shall emit radio NDBs in accordance with Table 120.

The NDB timing indicated in this document provides rapid (1 s) response times and <2 % receiver duty cycles.

8.10.1 Methods of network discovery

8.10.1.1 Minimum requirements

A network coordinator shall support at least one of the two network connection types.

8.10.1.2 Temporary network

A temporary network is a network where a device may send only one data message that reports status to the joining device's server. Once the device reports status, additional transmissions by the device are not required.

8.10.1.3 Associated networks

Associated networks maintain device communication for a period of time. Once a device has negotiated with a coordinator to become a member of the network, the device is considered associated. Associated networks are used for data uploads, link encryption, tag authentication, and network access control. To remain associated with a network, devices shall periodically provide data messages to the coordinator per the response interval defined in Table 120.

8.10.2 Transmitting network discovery beacons

The NBD timing shall be implemented as shown in Figure 38. NDB timing is critical for power management. A joining device shall enable its receiver for a minimum of 20 ms every second to detect an available network coordinator.

NDBs shall be transmitted as a 1 s burst of NDB messages, followed by a time period of no NDB transmissions. In the quiescent period, command messages, data messages, acknowledgement messages, and other ISO/IEC/IEEE 8802-15-4 MAC frames may be issued.

The period between NDB transmissions is determined by expected dwell times of joining devices at the network coordinator's location.

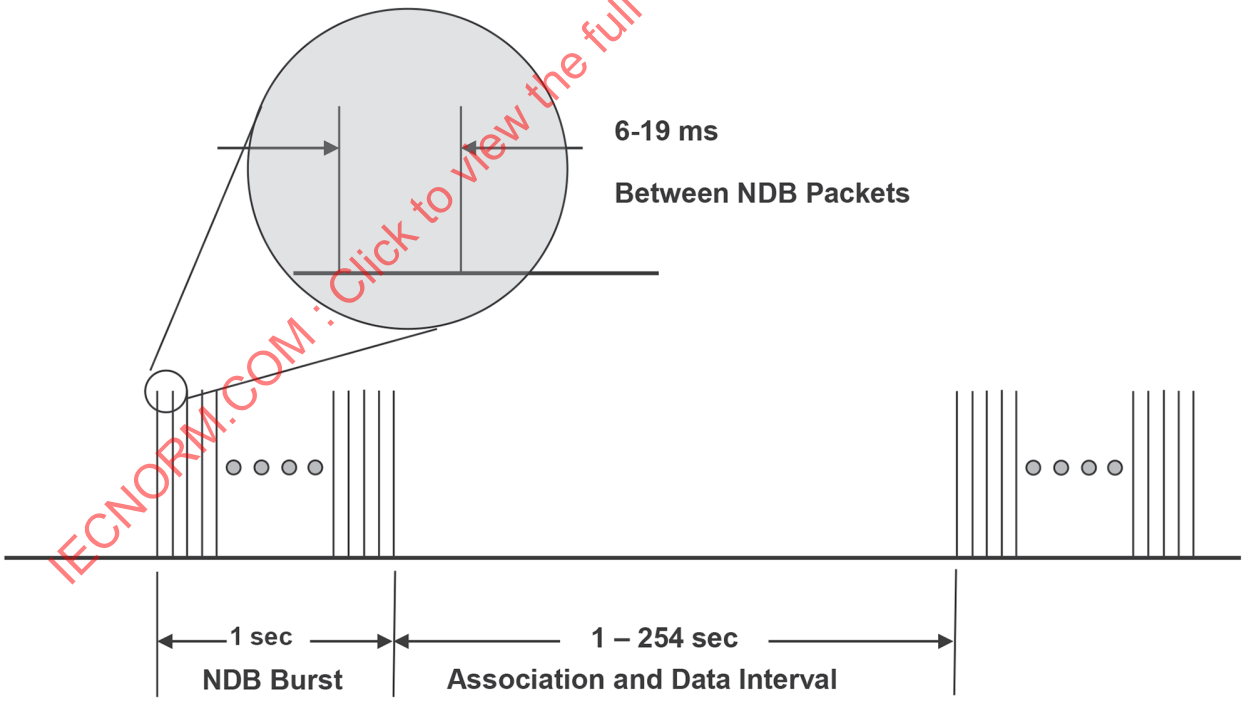


Figure 38 — NDB timing

8.10.3 Connectionless network

A connectionless network is a temporary network used to obtain status from remote devices. Examples include: installation at gates, on a crane, or at a weigh station. For these applications, the device being read is not expected to remain in the proximity of the network coordinator for long durations of time. This connection is not maintained and the device does not transmit significant amounts of data.

Embedded in the NDB is a response interval that a device may use to ignore repeated NDBs if the device remains near a network coordinator. Figure 39 illustrates the message exchange that takes place in this type of network.

8.10.3.1 Connectionless NDB message

Devices according to this document shall use NDB message packets described in 8.9.

8.10.3.2 NDB device response for NDBs

Joining devices shall respond to network coordinator NDB packets with a network status message (NSM). The network coordinators shall use parameters in the NSM to authenticate and route data to the appropriate server specified by the device.

8.10.3.3 NSM encryption

The NSM may provide an encrypted payload for the joining device server to authenticate if the NSM is from a valid device. The network coordinator is not required to decrypt the NSM payload.

The joining device's server shall respond with an acknowledgement to the device via the coordinator for the NSM.

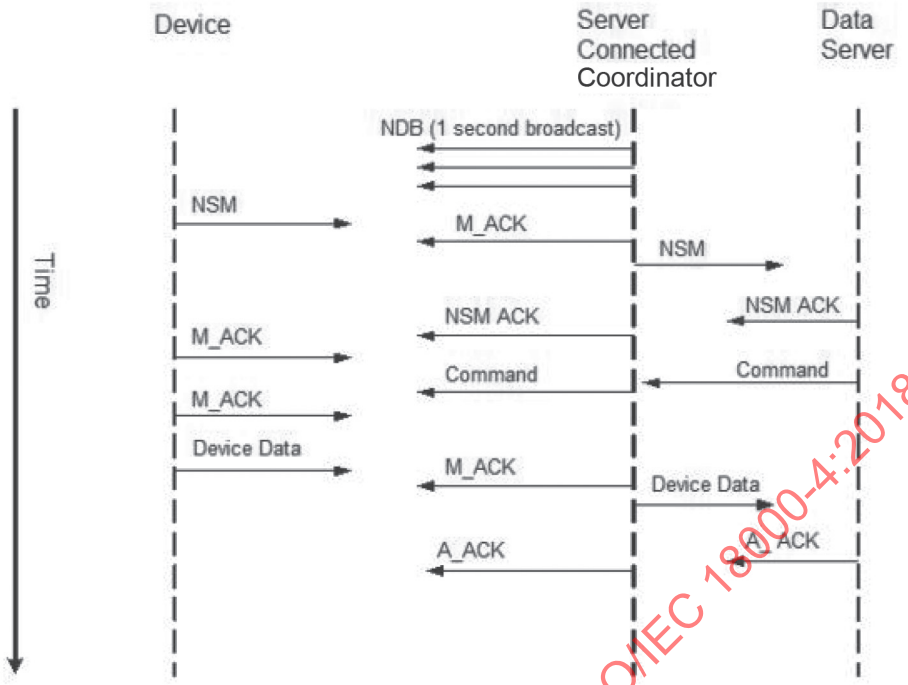
8.10.3.4 Data exchange

After the NSM parameters are exchanged and a route is established to the joining devices server, the joining device may send data to the network coordinator to be forwarded to the joining device's server.

A device shall send at least one packet every response interval value in the NDB message.

8.10.3.5 Commanding

While a device is transmitting data via the connectionless network, the devices server may issue commands to the device with that device's wake cycle indicated in the NSM.



Key

NDB	Network Discovery Beacon
NSM	Network Status Message
NSM_ACK	Acknowledgement of NSM
Command	Any directive to Device
M_ACK	802.15.4 Acknowledgement
A_ACK	Application Acknowledgement

Figure 39 — Network discovery — Connectionless network message flow

8.10.4 Associated network connection (ANC)

The associated network is intended for use in an area where devices will dwell for any period of time. This type of network connection is used for uploading significant amounts of data from a device or to establish trusted-networks that may use link layer encryption. The associated network connection provides a means for indicating a device is in a secured location. See Figure 40 for details.

A network coordinator may perform device commanding in an efficient manner in this type of network. If the device's receiver is active while it is idle, the device may indicate this fact in the appropriate field in the association request command. A network coordinator may command this device in a peer-to-peer ad-hoc fashion, skipping the commands pending step described in the NDB packet definition clause.

If the device requires an encrypted communication link, the network status message (NSM) provides a field indicating the device requests an encrypted link. The network coordinator will request a link encryption key from the device's server to be used by the network coordinator to communicate with the device. See Figure 32 for a diagram of the NSM exchange.

A device may choose to not participate in the association process. In this case, the device will treat this network coordinator as a connectionless network.

8.10.4.1 Associated network NDB message

This document uses NDB message packets to announce network availability for sustained network connections (see 8.9.2).

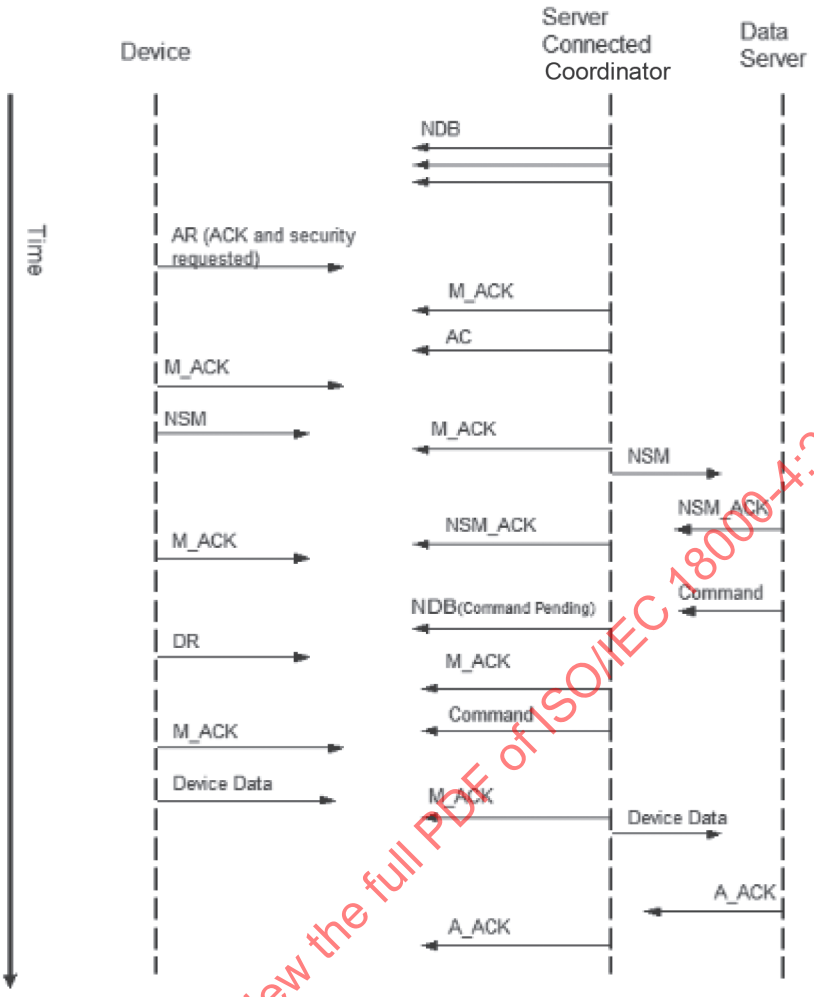
8.10.4.2 ISO/IEC/IEEE 8802-15-4 association request and association response

Joining devices shall use association request command frames as specified in ISO/IEC/IEEE 8802-15-4 to communicate that an associated network connection is desired. If a network coordinator is accepting connections, the network coordinator shall use association response commands as specified in ISO/IEC/IEEE 8802-15-4 frames to notify the remote device that it has joined the network. The joining device shall not begin transmitting additional data while waiting for the association response from the network coordinator.

Both the joining device and the network coordinator may reject either association command frame type if the network or device operators have specific security or authentication requirements.

8.10.4.3 Link encryption requests

If the joining device requests link encryption in the NSM and the device server provides a link key to the network coordinator, then the network coordinator shall communicate with the device using the encryption methods specified in 8.11.



Key

NDB	Network Discovery Beacon
B	802.15.4 Beacon (Association Bit Set)
B*	802.15.4 Beacon (Association Bit Not-Set)
AR	802.15.4 Association Request
AC	802.15.4 Association Command
DR	802.15.4 Data Request Command
M_ACK	Acknowledgement
A_ACK	Application Acknowledgements

Figure 40 — Network discovery — Associated network

8.11 Link encryption methods

Link encryption is optional and each remote tag may have a unique key. The network shall be capable of operating in both unencrypted and encrypted modes simultaneously.

Link encryption shall be implemented as specified in ISO/IEC/IEEE 8802-15-4:2018, Clause 9. Link encryption is not supported in a connectionless network.

In an associated network, link encryption shall be implemented using an NSM Key Exchange and only if the Device Server Connection Method is specified as TCP/IP.

At a minimum, two of the eight ISO/IEC/IEEE 8802-15-4 security levels shall be supported.

The two mandatory levels are: security level identifier 0x00, which indicates no encryption and no authentication code, and security level identifier 0x06, which indicates ENC-MIC-64 (see ISO/IEC/IEEE 8802-15-4:2018, Table 9-6).

The ENC-MIC-64 security level shall be CCM using AES-128. Both CCM and AES-128 are well documented in NIST Special Publication 800-38C and FIPS Publication 197 and exportable and in most cases supported in hardware.

The network status message shall request that the remote server provide the network coordinator with a link key. Only ISO/IEC/IEEE 8802-15-4 key identifier mode 0x00 (see ISO/IEC/IEEE 8802-15-4:2018, Table 9-7) shall be supported. The network coordinator shall use this specified link key when communicating with the particular remote tag until that remote tag has left the network.

9 MODE 4: Configurable data rate active RFID system

9.1 General

This mode describes the air interface for active RFID systems operating in the 2,45 GHz bands for item management applications. It provides a technical specification for active RFID tags and interrogators. The active RFID system defined by this model provides the functions of long range objects identification and environmental sense. This mode is intended to realize the low cost device and low power consumption, long range identification, fast and reliable tags access.

Tag access is the process in which tags establish communication links with interrogators according to certain communication protocols.

Some of the functional capabilities of this mode are as follows:

a) Support for O-QPSK modulation with 250 kbps or DBPSK modulation with variable information rate for higher reliable transmission.

b) Spread spectrum is adopted for reliable transmission at 2,45 GHz band.

c) The high efficient anti-collision algorithm is provided for fast tag access.

d) Support for directly tag response in the single or few tags application.

e) Support for internal and external wake-up mode to maximize the tag power use.

f) Support for three protocol operation modes: inventory mode, monitor mode and information release mode. In the inventory mode, the active tags are fast identified by the anti-collision algorithm; in the monitor mode, the sensor data on the tag side are periodically transmitted to the interrogators; and in the information release mode, the interrogators only broadcast data to tags in RF fields.

g) Support for file system on tag side to provide data organization and security memory.

h) Flexible protocol operation to decrease the communication delay and the power consumption.

i) Data security and authentication.

9.2 Cryptographic suite indicators

A tag may support one or more cryptographic suites. The authenticate commands include a CSI field that specifies a single cryptographic suite. CSI is an 8-bit field with bit values defined below.

a) Four most-significant bits: Cryptographic suite assigning authority, as follows:

— 0000_2 – 0011_2: ISO/IEC 29167, (all parts);

— 0100_2 – 1100_2: RFU;

— 1101_2: Tag manufacturer;

— 1110_2: GS1;

— 1111_2: RFU.

b) Four least-significant bits: One of 16 cryptographic suites that the assigning authority may assign.

EXAMPLE CSI=00000000_2 is the first and CSI=00000001_2 is the second suite that ISO/IEC 29167, (all parts) may assign.

9.3 Physical layer

9.3.1 Operating frequency

The frequency range of RFID systems defined in this document is between 2 400,00 MHz and 2 483,50 MHz. There are 16 channels in this band, and their serial numbers are from 0 to 15. The bandwidth of each channel is 5 MHz. The default channel is Channel 0 and centre frequency is 2 405,00 MHz, and the frequency tolerance is $\pm 20 \times 10^{-6}$ (20 ppm) maximum. The centre operating frequency of each channel is shown in Table 132.

Table 132 — Channel centre operating frequency

Channel serial number	Centre operating frequency	Channel serial number	Centre operating frequency
0	2 405,00 MHz	8	2 445,00 MHz
1	2 410,00 MHz	9	2 450,00 MHz
2	2 415,00 MHz	10	2 455,00 MHz
3	2 420,00 MHz	11	2 460,00 MHz
4	2 425,00 MHz	12	2 465,00 MHz
5	2 430,00 MHz	13	2 470,00 MHz
6	2 435,00 MHz	14	2 475,00 MHz
7	2 440,00 MHz	15	2 480,00 MHz

9.3.2 Emission spectrum density mask

Emission spectrum density mask should meet the requirement in Table 133.

Table 133 — Emission spectrum density template

Modulation	Frequency	Relative value	Absolute value
O-QPSK	$\|f\text{-}f_c\|$>3,5 MHz	<-20 dB/100 kHz	<-30 dBm/100 kHz
DBPSK	$\|f\text{-}f_c\|$>3,5 MHz	<-20 dB/100 kHz	<-20 dBm/100 kHz

Relative value and absolute value should be measured under the condition of 100 kHz resolution bandwidth. Reference level should be the maximum value in the range of carrier frequency ±1 MHz.

9.3.3 Modulation and spectrum spread

9.3.3.1 O-QPSK

Tags and interrogators that comply with this document should support O-QPSK modulation. In the O-QPSK modulation, the 16 dimensional quasi-orthogonal modulation is used to map every four bits of information to a meta-data symbol firstly, then each data symbol is mapped to one of the 16 quasi-orthogonal spreading sequences, and each chip in the spreading sequence is modulated to the carrier

by O-QPSK. The functional block diagram of modulation and spectrum spreading process is shown in Figure 41.

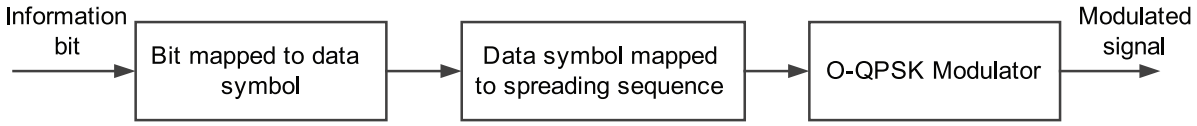


Figure 41 — O-QPSK modulation and spectrum spreading process

In the O-QPSK modulation, the length of the spreading sequence is 32 bits. The relationship among the four information bits (b_0, b_1, b_2, b_3), the 16 meta-data symbols and the quasi-orthogonal spreading sequence is shown in Table 134. During each symbol period, the LSB c_0 is transmitted firstly while the MSB c_{31} is the last.

Table 134 — Information bit to spreading sequence mapping

Information bit (b_0, b_1, b_2, b_3)	Data symbol (Decimal)	Spreading sequence (c_0, c_1, ..., c_{30}, c_{31})
0 0 0 0	0	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1 0 0 0	1	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
0 1 0 0	2	0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
1 1 0 0	3	0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
0 0 1 0	4	0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1
1 0 1 0	5	0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0
0 1 1 0	6	1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1
1 1 1 0	7	1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1
0 0 0 1	8	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
1 0 0 1	9	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1
0 1 0 1	10	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1
1 1 0 1	11	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
0 0 1 1	12	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0
1 0 1 1	13	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1
0 1 1 1	14	1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0
1 1 1 1	15	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0

The spreading spectrum sequence corresponding to each data symbol is modulated to the carrier by O-QPSK and pulse shaping. Even-indexed chips are modulated to the in-phase (I) carrier and odd-indexed chips are modulated to the quadrature-phase (Q) carrier. Because each data symbol is represented by 32 chips, the chip rate is 32 times as much as the symbol rate. In order to form offset between I-phase chips and Q-phase chips, the Q-phase chips should be delayed by T_c with respect to the I-phase chips (see Figure 42), where T_c is the inverse of chip rate.

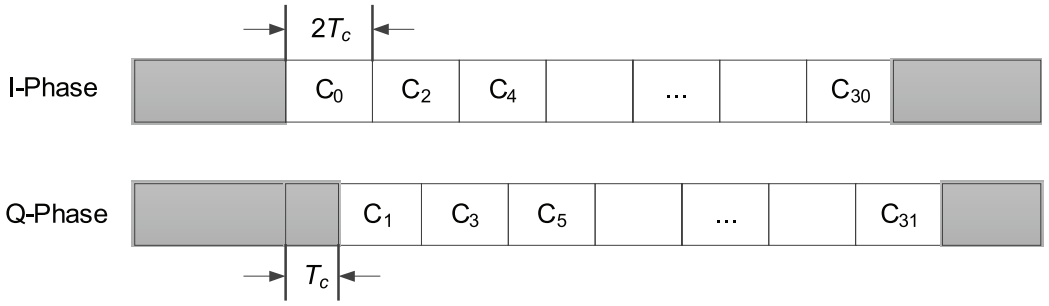


Figure 42 — O-QPSK modulation chips offset

9.3.3.2 Pulse shaping using O-QPSK modulator

The pulse shaping function of each chip is shown in Formula (1):

$$p(t)=\begin{cases}\sin\dfrac{\pi t}{2T_c}, 0\le t\le 2T_c\\ 0, \text{others}\end{cases} \quad (1)$$

where

$p(t)$ is the pulse amplitude, V;

t is the time, s;

T_c is the pulse period, s.

9.3.3.3 DBPSK

Tags and interrogators that comply with this document can optionally support DBPSK modulation. In the DBPSK modulation, each information bit should be differentially encoded firstly. Each encoded information bit '0' or '1' is mapped to a spreading sequence respectively and then each chip in the spreading sequence is modulated to the carrier by BPSK. The functional block diagram of modulation and spreading process is shown in Figure 43.

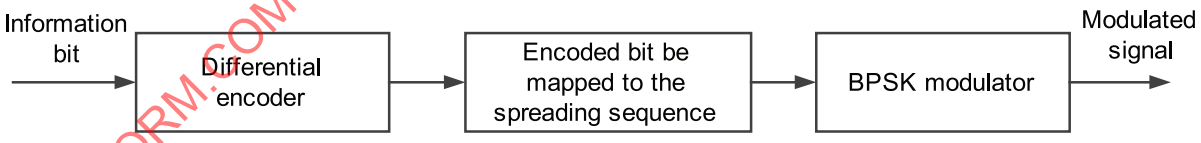


Figure 43 — DBPSK modulation and spreading functions

Differential encoding rules are shown in Formula (2):

$$E_n = R_n \oplus E_{n-1} \quad (2)$$

where

E_n is the differentially encoded bit;

R_n is the information bit to be encoded;

E_{n-1} is the previous differentially encoded bit.

For each transmission, R_1 is the first information bit to be encoded, and E_0 is '0'.

The rules of differential encoding and decoding are shown in Formula (3):

$$R_n = E_n \oplus E_{n-1} \tag{3}$$

where

R_n is the decoded information bit;

E_n is the differentially encoded bit;

E_{n-1} is the previous differentially encoded bit.

For each transmission, E_1 is the first differential encoding bit to be decoded, and E_0 is '0'.

In the DBPSK modulation, four types of spreading sequence lengths (8-bit, 32-bit, 64-bit and 128-bit) can be optionally supported. During each information bit period, the LSB c_0 is transmitted firstly and the MSB c_{31} is the last. Spreading sequences corresponding to each information rate are defined by users. They should be quasi-orthogonal pseudo-random sequences.

EXAMPLE When the length of spreading sequences is 32 bits, a typical spreading sequence is shown in Table 135. Differentially encoded information bits '0' and '1' correspond to 32-bit spreading sequence, respectively.

Table 135 — Differentially encoded information bit to spreading sequence mapping

Differential encoding bit	Spreading sequence (c_0, c_1, ..., c_{30}, c_{31})
0	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0

9.3.3.4 Pulse shaping using DBPSK modulation

The pulse shaping function of raising cosine which is adopted by chip is shown in Formula (4):

$$p(t) = \begin{cases} \dfrac{\sin(\pi t / T_c)}{\pi t / T_c} \times \dfrac{\cos(\pi t / T_c)}{\left(1 - 4t^2 / T_c^2\right)}, t \neq 0 \\ 1, t = 0 \end{cases} \tag{4}$$

where

$p(t)$ is the pulse amplitude, V;

t is the time, s;

T_c is the pulse period, s.

9.3.3.5 Information rate

The chip rate, the length of spreading sequence and the information rate in the O-QPSK and DBPSK modulations are shown in Table 136 and Table 137, respectively. In the DBPSK modulation, the user can choose different lengths of spreading sequence according to Table 137 to get the corresponding information rate. The tolerance of chip rate is ±20 × 10^{-6} (20 ppm) maximum and the tolerance of bit rate is the same as ±20 × 10^{-6} (20 ppm) maximum.

Table 136 — Comparison between length of spreading sequence and information rate by O-QPSK

Chip rate Mcps	Length of spreading sequence chip	Information rate kbps
2	32	250

Table 137 — Comparison between length of spreading sequence and information rate by DBPSK

Chip rate Mcps	Length of spreading sequence chip	Information rate kbps
2	8	250
2	32	62,5
2	64	31,25
2	128	15,625

9.3.4 TX/RX switch time

The RX-to-TX turnaround time shall be less than or equal to 192 µs, and the TX-to-RX turnaround time shall be less than or equal to the RX-to-TX turnaround time.

9.3.5 EVM

In the condition of measuring 1 000 chips, EVM in O-QPSK and DBPSK modulation should be no more than 35 %.

9.4 Data link layer

9.4.1 General

Data frames are used to transmit data between interrogators and tags. Data frame is composed of preamble, synchronous code, data length, frame option, message data and CRC, as shown in Table 138. Message data are composed of several data items. The details are given in 9.4.6. The transmission of data frames is in the order of preamble, synchronous code, data length, frame option, message data and CRC. For each data item composed by many bytes, the most significant byte should be transmitted firstly. And for each byte, the least significant bit should be transmitted firstly.

Table 138 — Data frame structure

Preamble	Synchronous code	Data length	Frame option	Message data	CRC
4 bytes	1 byte	1 byte	1 byte	≤124 bytes	2 bytes

9.4.2 Preamble

Preambles of the forward and reverse data frame are 32-bit binary numbers, and each bit is '0'.

9.4.3 Synchronous code

Synchronous codes of the forward and reverse data frame are 8-bit binary numbers. Details are shown in Table 139.

Table 139 — Synchronous code

Information bit	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7
Value	1	1	1	0	0	1	0	1

9.4.4 Data length

Data length refers to the total length from frames option to CRC, and its unit is byte. It occupies one byte. b_0 to b_6 represent length and b_7 is reserved.

9.4.5 Frame option

Frame option includes modulation scheme, information rate, frame direction, frame type and so on. Frame option occupies one byte, and its definitions are shown in Table 140.

Table 140 — Frame option definition

b_0~b_2	b_3	b_4	b_5~b_7
000_b: O-QPSK/250 kbps 100_b: DBPSK/250 kbps 010_b: DBPSK/62,5 kbps 110_b: DBPSK/31,25 kbps 001_b: DBPSK/15,625 kbps Others: Reserved	0_b: Interrogators to tags 1_b: Tags to interrogators	0_b: Broadcast 1_b: Point to point	Reserved

b_0~b_2 represent modulation scheme and the corresponding information rate. By default, a tag shall initially work in the O-QPSK modulation after it was woken up. Interrogators set the modulation scheme and the information rate that are used in subsequent communication between interrogators and tags in the frame option. After a tag receives the interrogator frame option, it will communicate with the interrogator in subsequence frame in the modulation scheme and information rate that are indicated in the frame option. The tag keeps the existing modulation scheme and information rate until it receives the next interrogator data frame. And it records the modulation scheme and information used in the current frame option.

b_3 represents the frame direction. If the value of b_3 is 0_b, it indicates that the data frame direction is from interrogators to tags (forward). And if the value of b_3 is 1_b, it indicates that the data frame direction is from tags to interrogators (reverse).

b_4 represents the broadcast frame or point to point frame. If the value of b_4 is 0_b, it indicates that the data frame is broadcast frame. If the value of b_4 is 1_b, the data frame is point to point frame.

b_5~b_7 are reserved, and the default value is 000_b.

9.4.6 Message data

9.4.6.1 Interrogator to tag

Data frames that an interrogator sends to a tag include broadcast frame mode and point to point frame mode. The broadcast frame is used when an interrogator sends data to all tags simultaneously, while the point to point frame is used when an interrogator sends data to a specified tag. After receiving the broadcast frame, all tags should process the frame data. After receiving the point to point frame, tags should compare TID in the data frame with that of their own. If they are consistent, the point to point frame data will be processed. Otherwise, there will be no response.

The message data format of broadcast frames from interrogators to tags is shown in Table 141.

Table 141 — Message data format of broadcast frame from interrogators to tags

RID	Command code	Command parameter
3 bytes	1 byte	≤120 bytes

The message data format of point to point frames from interrogators to tags is shown in Table 142.

Table 142 — Message data format of point to point frame from interrogators to tags

TID	RID	Command code	Command parameter
8 bytes	3 bytes	1 byte	≤112 bytes

Data items in Table 141 and Table 142 are defined as follows:

a) TID: It is the unique 8 bytes binary number for each tag which is written by manufacturers that cannot be changed by users. TID is used to uniquely identify one tag in the communication process. For more information about the TID format, refer to 9.4.6.4.

b) RID: It is a 3 bytes binary number which is used to uniquely identify one interrogator in the communication process. The composition of RID is defined by users, and both the broadcast frame and the point to point frame contain it.

c) Command code: It is used to identify different commands, and its length is one byte. For command codes list, see Table 158.

d) Command parameter: Each command parameter is defined in 9.7.

The broadcast frame does not contain the TID data item, but the point to point frame contains it.

9.4.6.2 Tag to interrogator

9.4.6.2.1 Overview

Data frames from tags to interrogators include the long frame mode and the short frame mode.

Tags should respond to interrogator commands, but there are two exceptions as follows:

a) The interrogator commands that tags do not need to respond.

b) The CRC that tag has received does not match the result that is calculated according to the received data.

9.4.6.2.2 Short frame from tags to interrogators

The format of short frame from tags to interrogators is shown in Table 143. It is only used when tags respond to the access command sent by interrogators. For more information about the access command, see 9.7.4.

Table 143 — Message data format of short frame from tags to interrogators

Response data
1 byte

9.4.6.2.3 Long frame from tags to interrogators

The format of long frame from tags to interrogators is shown in Table 144. It is used when tags respond to interrogator commands except the access command.

Table 144 — Message data format of long frame from tags to interrogators

Tag state word	RID	TID	Command code	Response data
1 byte	3 bytes	8 bytes	1 byte	≤111 bytes

Data items in Table 144 are defined as follows:

a) Tag state word: It shows the basic state information of tags, including battery power, without sensors, initialized state and so on. Details are defined in Table 145, in which the initialization refers

to processes of establishing embedded operating system and file system, writing configuration information of tag and so on.

b) RID: It is a 3 bytes binary number which is used to uniquely identify one interrogator in the communication process. The composition of RID is defined by users.

c) TID: It is the unique 8 bytes binary number for each tag which is written by manufacturers that cannot be changed by users. TID is used to uniquely identify one tag in the communication process. For more information about the TID format, refer to 9.4.6.4.

d) Command code: It is the command code from interrogators to tags.

e) Response data: It is the response data that tags respond to valid commands from interrogators. Some response data consist of the execution state data in two bytes, and for the execution explanation, see 9.4.6.3.

Table 145 — Tag state word definition

b_0~b_1	b_2	b_3	b_4~b_7
Battery power	Sensor	Tag initialization	Reserved
00_b: 75 %~100 % (Full power) 10_b: 50 %~75 % 01_b: 10 %~50 % 11_b: <10 % (Low power)	0_b: With 1_b: Without	0_b: Uninitialized 1_b: Initialized	—

9.4.6.3 Execution state

When the tag is executing broadcast command or point to point command, if successfully executed, it means that the tag has received a valid command from interrogators and successfully processed. But if it fails, it means that the command is not valid or the tag encounters an error while processing the command.

When the tag is executing a point to point command, if successfully executed, the response data include execution state, and its execution code is 0000_h. If it fails, the response data of the tag only contain execution state and its code is the error code which is shown in Table 146. Error code contains error category code and error sub-code. Error category code indicates the error category. Each error sub-code further indicates the meaning of error. Error category code and error sub-code are defined in Table 147.

When the tag is executing a broadcast command, if successfully executed, it will return a response. The response data include execution state, and its execution code is 0000_h. In unslotted operation, if it fails, the tag does not return any error information. In time-slot operation, the tag will only respond to the interrogator command whose code is correct, and the tag will return the error category code and the error sub-code according to time-slot. The error category code and the error sub-code are defined in Table 147.

Table 146 — Error code format

Error category code	Error sub-code
1 byte	1 byte

Table 147 — Definition of error category code and error sub-code

Error category code	Meaning	Error sub-code	Meaning
01_h	Invalid command code	—	A tag received the information packet which contains an undefined command code.
02_h	Invalid command parameter	01_h	Parameter is invalid.
		02_h	Bytes in the command parameter are less than expected
		03_h	Bytes in the command parameter are more than expected.
		Others	Reserved
03_h	Unsupported optional command code	—	A tag receives the optional command which is defined in the standard, but does not support the optional command.
04_h	Command execution failure	—	Unexpected error occurred while a tag executes a command.
05_h	Identification failure	—	Entity authentication failure between interrogators and tags.
06_h	Files access failure	00_h	Operated file does not exist.
		01_h	Do not meet control condition of file access.
		02_h	Files being set as invalid.
		03_h	Access record does not exist.
		04_h	Access field does not exist.
		05_h	The last record already.
		06_h	Beyond the boundary.
		Others	Reserved
07_h	Monitoring mode failure	00_h	Unknown
		01_h	Data of boundary violation.
		02_h	Configuration for monitoring period failing.
		Others	Reserved
09_h	Killing operation failure	00_h	Unknown
		01_h	Killing password does not match.
		02_h	Killing password of tags is zero.
		Others	Reserved
$0A_h$	Modification of system passwords	00_h	Unknown
		00_h	Password of administrator does not match.
		01_h	Invalid password index or mode.
		02_h	New password is invalid.
		03_h	Administrator password does not match.
		Others	Reserved
08_h,$0B_h$-FF_h	Reserved	—	—

9.4.6.4 TID format

The tag identifier consists of the allocation class, the tag manufacturer code and the tag serial number, and the allocation class is 00100000_b (20_h).

The tag manufacturer code is an 8-bit binary number.

The tag serial number is a 48-bit binary number and is assigned by the tag manufacturer.

The TID format is shown in Table 148.

Table 148 — TID format

Allocation class	Tag manufacture code	Tag serial number
8 bits	8 bits	48 bits

9.4.7 CRC

The CRC-16 check mode is adopted in this document, the checkout result occupies two bytes, and the checkout content consists of two data items (the frame options and the message data). CRC-16 is calculated by the polynomial $x^{16} + x^{12} + x^5 + 1$, and its initial value is 0000_h.

9.5 Tag memory structure

9.5.1 Overview

Tag memory includes the security section and the user section. Security section is used to store tag security data. The storage location and storage mode can be defined by the user. User section is used to store user data, which is organized and managed by file operation system.

9.5.2 Data organization of security section

Security section is used to store tag security data. It consists of the administrator password, the killing password and the ready/sleep password, and they are shown in Figure 44.

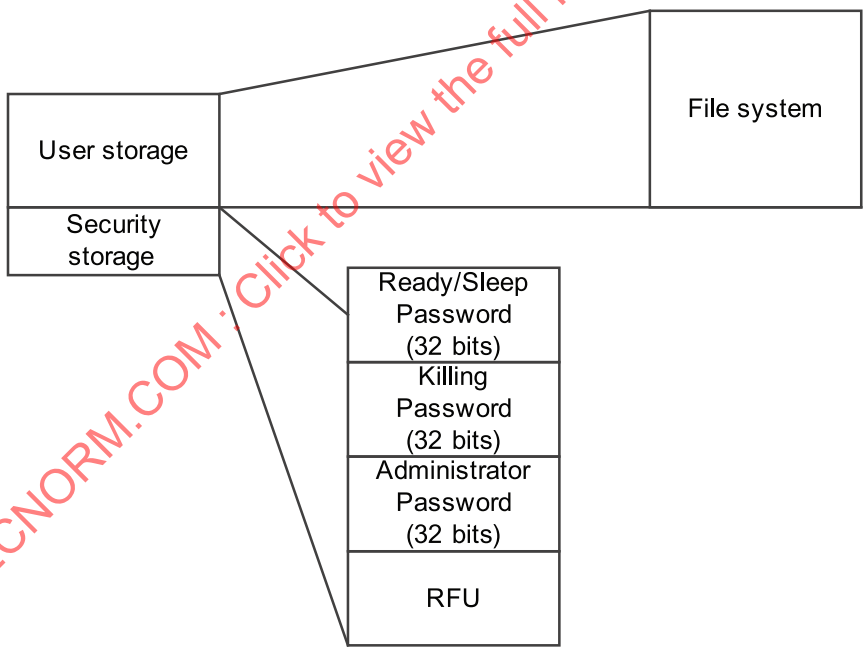


Figure 44 — Memory logic mapping

The administrator password, the killing password and the ready/sleep password should be 32-bit binary numbers and MSB is stored in low address memory. Those passwords should be written in the initialization stage. During the operation of tags, those passwords can only be read and used by internal programs of tag and cannot be accessed by any external commands.

The killing password and the ready/sleep password can be modified by the interrogator after the completion of the authentication through the administrator password, but they should not be read.

9.5.3 Data organization of user section

9.5.3.1 File system structure

The file system contains two files:

a) Dedicated file (DF): DF is used to support files folder. DF is the parent file of EF.

b) Elementary file (EF): EF is used to store data. EF cannot be the parent file of other files.

The data in user section of tag are composed by DF and EF through the structured classification. The DF at the root of file system is called MF. The MF is mandatory for a tag, and a tag is only one MF. MF can include the subordinate DF with the maximum nesting of two layers. The upper file is the parent file of this layer file. EF is included in the MF or other DFs. Empty file is a special kind of DF. It is included in the MF and can be only identified and selected but it cannot include subordinate files. File system structure of a tag is shown in Figure 45.

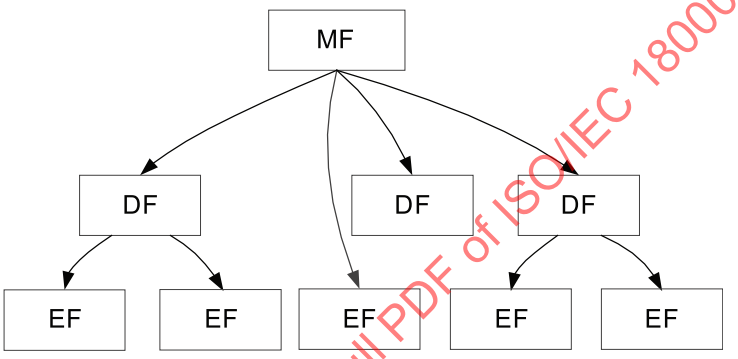


Figure 45 — Structures for file system of a tag

9.5.3.2 EF data structure

The EF includes transparent structures or record structure:

a) Transparent structure: EF can be considered as a sequence data unit.

b) Record structure: EF can be considered as a recording that can be identified by sequence individually. The record structure only supports fixed-length and be organized by linear order.

The EF data structure is shown in Figure 46.

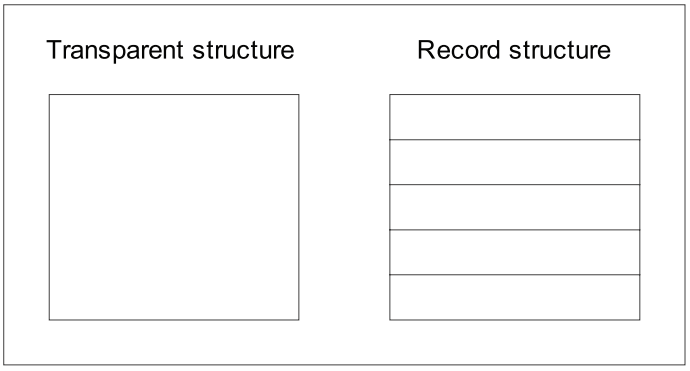


Figure 46 — EF data structures

The DF and EF contain not only the user data but also the file attributes information. The file attributes information is shown in Table 149.

Table 149 — File attributes information

Name	Address	Description
File identifier	Byte 1~2	The unique identifier addressed a file within a tag.
File type and valid information	Byte 3	File type: Bit 0 to 3 — DF: 0000_b; — Transparent EF: 0001_b; — Record EF: 0011_b. Valid information: Bit 4 — File valid: 0_b; — File invalid: 1_b. Others: Reserved
File maximum available length	Byte 4~5	DF: 0001_h Transparent EF: Length value of file content (not excluding the file attributes) in bytes Record EF: Length value of each record in bytes
File maximum allowable number of records	Byte 6~7	DF: The number of files under the DF Transparent EF: 0001_h Record EF: The maximum allowable number of records
File actual length	Byte 8~9	DF: The number of files under the DF Transparent EF: Length value of file content (not excluding the file attributes) in bytes Record EF: Actual number of records (not excluding the file attributes)
File security control attribute	Byte 10	Security control conditions of file access (see Tables 151, 152 and 153

9.5.3.3 File selection method

Any file can be indexed by the file identifier. The range of the file identifier code is from $3F00_h$ to $FFFF_h$. MF is identified by the value '$3F00_h$'. All EFs and DFs should have different file identifiers in the same DF.

The file selection includes ambiguous selection and unambiguous selection. Ambiguous selection is that the file is default selected, not by the file command. Unambiguous selection is that the file is selected by the file command.

MF is ambiguous selection. It will be selected when the tag transfers into the ready state. Other files including empty files should be selected unambiguously by their identifiers.

9.5.3.4 File system operation command

The file system operation command contains the select file, file password verification, file rehabilitate, file invalidate, read transparent file, update transparent file, read record file, update record file and search record commands.

Definitions of file system operation commands for file types are shown in Table 150.

Table 150 — File system operation commands for file types

File system operation commands	File type
Select file	DF, EF
File password verification	DF, EF
File invalidate	DF, EF

Table 150 *(continued)*

File system operation commands	File type
File validate	DF, EF
List file	DF
Read transparent file	Transparent EF
Update transparent file	
Read record file	Record EF
Update record file	
Search record	

Each file operation is composed of a set of the command-response sequence. The interrogator sends the file operation commands, including the operation command code and the data to be processed. The tag operates the commands and returns a response, including the command code, execution state and the response data.

9.5.3.5 File system operation process

The file system operation process (tag to interrogator) is as follows:

a) Select file: The interrogator selects a file, and then the tag returns the file attribute information including the file security control information. According to the file security control information, the interrogator decides whether to go to step b) or c).

b) Entity authentication: The interrogator judges the tag's current entity authentication state; if entity authentication is not completed, an entity authentication procedure should be processed between the interrogator and the tag, then go to step c); otherwise, go to step d) directly.

c) File password verification: The interrogator judges whether the current file operations have completed the password verification. If password verification is not completed, operation file verification password should be provided by the interrogator; if successful, then go to step d). If password verification is completed, go to d) directly.

d) File operation: The interrogator operates the file.

The file operation process is shown in Figure 47.

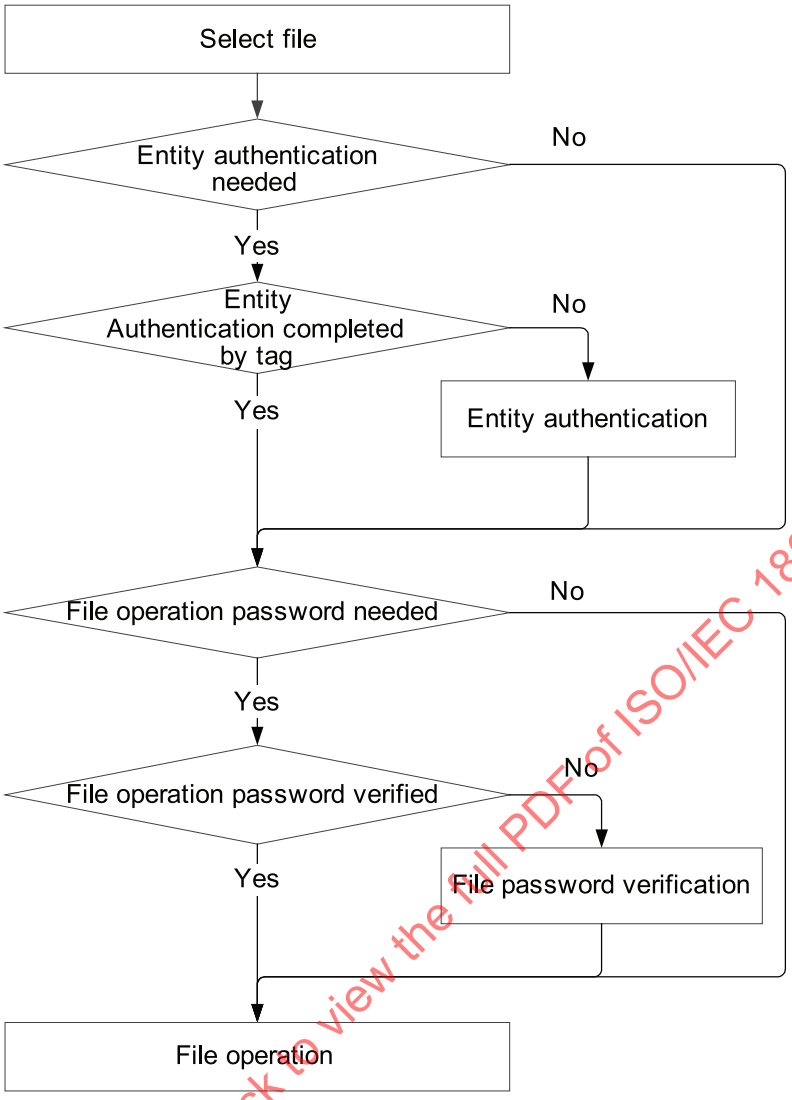


Figure 47 — File operation process

9.5.3.6 File system access control

Each file operation (interrogator to tag) should meet the conditions of the access control of file. The access control conditions of tag file should be satisfied before executing an operation. The condition of the access control is shown in Table 151.

Table 151 — Access control conditions of tag file

Access control condition	Description
ALW	Uncontrolled, unconditional implementation
PWD	Password needed
ATH	Authentication needed
NEV	Disallowed under any conditions

Whether it is a transparent file or record file, access control permissions and file operations are binding. If the access control permissions have been obtained, the same operations followed can be inherited. However, when changing the type of file operation in the follow-up file, the permissions of the operation should be re-verified.

The permissions of entity authentication obtained are subject to the entire tag, which are valid throughout the access period of the tag. The other permissions of access control conditions should only be limited to the specific file operations. When the file operated is changed, the permissions for the file will be invalid.

Users should define access control attributes for each DF or EF, respectively. The information should be only written in the initialization stage. During the operation of tags, the information can be read but not modified through the select file command by users.

The security control attributes of the DF and the EF are shown in Table 152 and Table 153, respectively.

Table 152 — DF security control attributes

b_0~b_1	b_2~b_3	b_4~b_5	b_6~b_7
Access control condition of select file	Access control condition of list file	Reserved	Access control condition of validate or invalidate file
00_b: ALW Others: Reserved	00_b: ALW 10_b: ATH 01_b: PWD 11_b: NEV	—	00_b: ALW 10_b: ATH 01_b: PWD 11_b: NEV

Table 153 — EF security control attributes

b_0~b_1	b_2~b_3	b_4~b_5	b_6~b_7
Access control condition of select file	Access control condition of read/query file	Access control condition of update file	Access control condition of validate or invalidate file
00_b: ALW Others: Reserved	00_b: ALW 10_b: ATH 01_b: PWD 11_b: NEV	00_b: ALW 10_b: ATH 01_b: PWD 11_b: NEV	00_b: ALW 10_b: ATH 01_b: PWD 11_b: NEV

9.5.4 File definitions of user section

The tag file system should be generated in the initialization stage. The tag shall contain at least MF, TID file, UII file, tag configuration file, empty file and query result file, and the file identifiers are defined in Table 154. The file directory structure is shown in Figure 48. Definitions of above files are shown in Table 154.

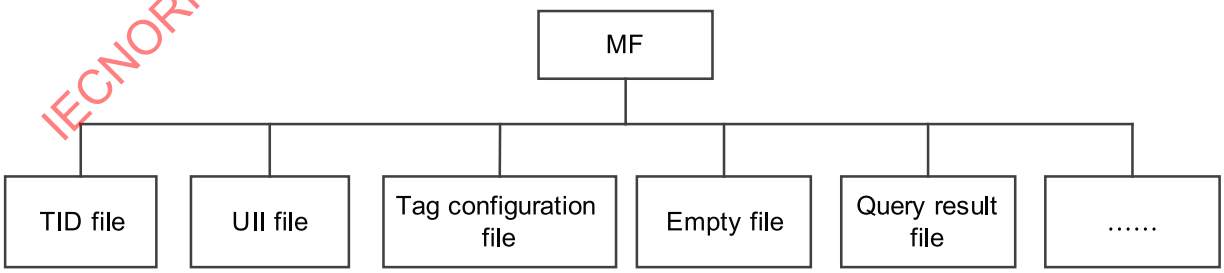


Figure 48 — File directory structure

Table 154 — File definitions of user area

File name	File identifier	File type	Explaining
MF	$3F00_h$	DF	See Figure 49
TID file	$A101_h$	Transparent EF	See Figure 50
UII file	$A102_h$	Transparent EF	See Figure 51
Tag configuration file	$A103_h$	Transparent EF	See Figure 52
Empty file	$A104_h$	DF	See Figure 53
Query result file	$F000_h$	Record EF	See Figure 54

File identifier: $3F00_h$	File type: DF	Mandatory/Optional: Mandatory
File actual length: --	Update frequency: --	
File security control attribute: ——Select operation: ALW ——List operation: ALW ——Read operation: N/A ——Update operation: N/A ——Valid or invalid operation: NEV		

Figure 49 — MF definition

File identifier: $A101_h$	File type: Transparent EF	Mandatory/Optional: Mandatory	
File actual length: 8 bytes	Update frequency: low		
File security control attribute: ——Select operation: ALW ——List operation: N/A ——Read operation: ALW ——Update operation: NEV ——Valid or invalid operation: NEV			
Memory allocation	Data content	Mandatory/Optional	Data length
Byte 1~8	TID	Mandatory	8 bytes

Figure 50 — Tag identifier file

File identifier: $A102_h$	File type: Transparent EF	Mandatory/Optional: Mandatory	
File actual length: User defined	Update frequency: low		
File security control attribute: ——Select operation: User defined ——List operation: N/A ——Read operation: User defined ——Update operation: User defined ——Valid or invalid operation: User defined			
Memory allocation	Data content	Mandatory/Optional	Data length
User defined	UII	Mandatory	32 bytes

Figure 51 — Unique item identifier file

File identifier: A103h	Mandatory/Optional: Mandatory	Mandatory/Optional: Mandatory
File actual length: 15 bytes	Update frequency: low	
File security control attribute: ——Select operation: ALW ——List operation: N/A ——Read operation: ALW ——Update operation: User defined ——Valid or invalid operation: NEV		
Memory allocation	Data content	Mandatory/Optional	Data length
Byte 1~4	Storage capacity	Mandatory	4 bytes
Byte 5	Maximum number of query elements	Mandatory	1 byte
Byte 6~7	Maximum number of query results of matches	Mandatory	2 bytes
Byte 8	Identifier of tag wakeup ability	Mandatory	1 byte
Byte 9	DBPSK modulation information rate	Mandatory	1 byte
Byte 10~11	Sense time	Mandatory	2 bytes
Byte 12~15	Sense time	Mandatory	4 bytes

Figure 52 — Tag configuration file

The wake-up ability in Figure 52 represents the wake-up methods supported by tags, and it occupies one byte. b_0~b_2 represent the support capability of tags for a certain wake-up method respectively, the value of 1_b means support the corresponding wake-up method while 0_b is contrary, b_3~b_7 are reserved. A tag supports at least one of the wake-up methods shown in Table 155.

In Figure 52, DBPSK modulation information rate occupies one byte. b_0~b_3 represent the support capability of a tag to a certain information rate respectively, the value of 1_b means support the corresponding information rate while th evlaue of 0_b is contrary, b_4~b_7 reserved, as shown in Table 156.

The sense time in Figure 52 represents the time that a tag stayed in sense state; the unit time is 0,1 ms. When the information rate is 250 kbps, 62,5 kbps, 31,25 kbps and 15,625 kbps, their minimum sense time is 2 ms, 8 ms, 16 ms and 32 ms, respectively. For tags which support the cycle sleep-wake-up, the sleep time represents the time that tags stayed in sleep state, and unit time is 1 ms. But if tags do not support the cycle sleep-wake-up, the sleep time makes no sense.

Table 155 — Tag wake-up ability

b_0	b_1	b_2	b_3~b_7
Cycle wake-up 0_b: not support 1_b: support	External 125 kHz wake-up 0_b: not support 1_b: support	External 2 kHz wake-up 0_b: not support 1_b: support	Reserved

Table 156 — DBPSK modulation information rate

b_0	b_1	b_2	b_3	b_4~b_7
DBPSK/15,625 kbps modulation 0_b: not support 1_b: support	DBPSK/31,25 kbps modulation 0_b: not support 1_b: support	DBPSK/62,5 kbps modulation 0_b: not support 1_b: support	DBPSK/250 kbps modulation 0_b: not support 1_b: support	Reserved

File identifier: A104$_h$	File type: DF	Mandatory/Optional: Mandatory
File actual length: 7 bytes	Update frequency: --	
File security control attribute: ——Select operation: ALW ——List operation: ALW ——Read operation: N/A ——Update operation: N/A ——Valid or invalid operation: NEV		

Figure 53 — Empty file

File identifier: A103$_h$	Mandatory/Optional: Mandatory	Mandatory/Optional: Mandatory
File actual length: User defined	Update frequency: high	
File security control attribute: ——Select operation: ALW ——List operation: N/A ——Read operation: ALW ——Update operation: NEV ——Valid or invalid operation: NEV		
Memory allocation	Data content	Mandatory/Optional	Data length
Byte 1~2	Record index1	Mandatory	2 bytes
Byte 3~4	Maximum number of query elements	Mandatory	2 bytes
⋮	⋮	⋮	⋮

Figure 54 — Query result file

9.6 Tag state transition

9.6.1 Overview

The tag states include sleep, sense, ready, operation and killing. And the operation state includes four sub-states: arbitration, collection, session and secure session.

9.6.2 Figure of tag state transition

The transition condition among the tag states is shown in Figure 55. Collection failure means that tags confirm collection failure when they receive collection failure command from interrogators in the collection period. Commands in the session period refer to all of the commands which can be transmitted from interrogators as the system is in the session period. It includes file access command, monitor command, security protocol command, updating system password command, killing command, and requesting random number command.

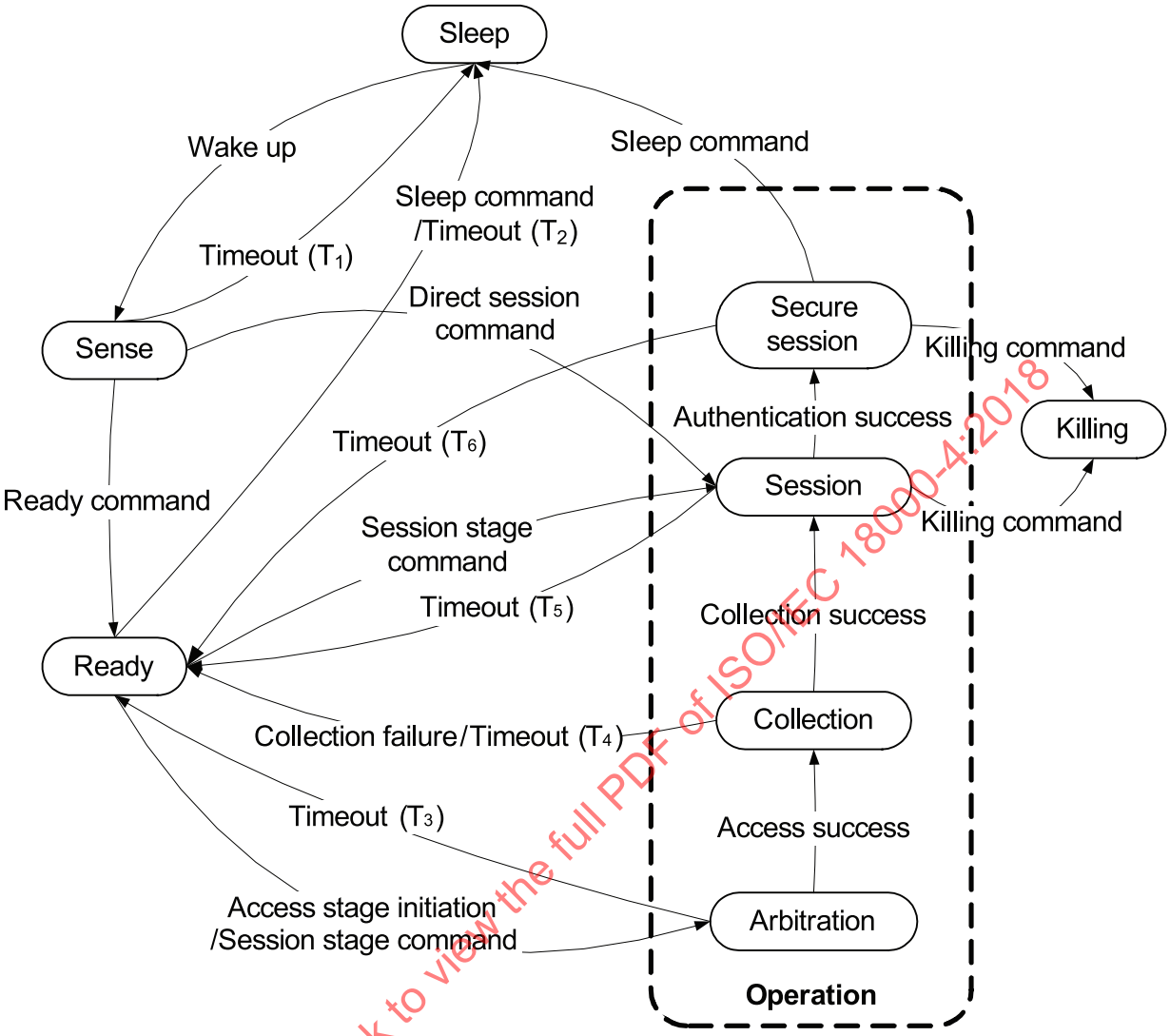


Figure 55 — Tag state transition

9.6.3 Sense state

In sense state, the tag listens to the reader command to respond the beginning of tag identification. After receiving the wake up signal, the tag in sleep state will transfer into sense state. The tag in sense state should turn on RX/TX circuit and wait for ready command or direct session command from the interrogator. If the tag receives ready command from the interrogator, it will transfer into ready state, and if the tag receives direct session command from the interrogator, it will transfer into session state. If the tag does not receive the ready command or direct session command within the specified time T_1, it will transfer into the sleep state. The value of T_1 is shown in Table 157.

Table 157 — Timeout value

Parameter	Explanation	Value
T_1	Tag in sense state will transfer into sleep state if ready command or direct session command is not received within the specified time T_1.	Sense time, shown in Figure 53
T_2	Tag in ready state will transfer into sleep state if it does not receive any interrogator commands within the specified time T_2.	30 s
T_3	Tag in arbitration state will transfer into ready state if the access command or the collection command is not received within the specified time T_3.	1 s
T_4	Tag in collection state will transfer into ready state if the collection command is not received within the specified time T_4.	30 s
T_5	Tag in session state will transfer into ready state if the interrogator command is not received within the specified time T_5.	5 s
T_6	Tag in session state will transfer into ready state if the interrogator command is not received within the specified time T_6.	30 s

9.6.4 Ready state

The ready state means the beginning of tag identification process, and the tag is ready for an anti-collision process. After receiving ready command, the tag in sense state will transfer into ready state. When the tag in ready state receives access frame initialization command, it will transfer into arbitration state. While the tag receives any of file access command, monitor command or security protocol command, the tag will transfer into session state. If the tag does not receive interrogator command within the specified time T_2, it will transfer into sleep state. The value of T_2 is shown in Table 157.

9.6.5 Operation state

9.6.5.1 Arbitration state

In arbitration state, a tag responds to the access commands so that multiple tags can be identified at the same time. The tag in ready state will transfer into arbitration state after receiving access frame initialization command. After receiving access success ACK command, the tag in arbitration state will transfer into the collection state; otherwise, it will remain in arbitration state. The tag will transfer into sleep state after receiving sleep command. And if access command or collection command is not received within the specified time T_3, the tag will transfer into ready state. The value of T_3 is shown in Table 157.

9.6.5.2 Collection state

In the collection state, the interrogator makes a fast tag information collection, such as TID, based on the arbitration results. The tag in arbitration state will transfer into collection state after receiving access success ACK command from the interrogator. After receiving collection success ACK command, the tag in collection state will transfer into session state. It will transfer into ready state after receiving the collection failure ACK command. The tag will transfer into sleep state after receiving sleep command. And if collection command or access command is not received within the specified time T_4, the tag will transfer into ready state. The value of T_4 is shown in Table 157.

9.6.5.3 Session state

In session state, the interrogator makes a point to point communication with individual tag by the known TID information. The tag in collection state will transfer into session state after receiving

collection success ACK command. The tag in sense state will transfer into session state after receiving file access command, monitor command or security protocol command. And the tag in sense state will transfer into session state after receiving direct session command. If entity identification is successful, the tag in session state will transfer into secure session state; otherwise, it will remain in session state. And if receiving killing command, tag memory will be erased and the tag will transfer into killing state. And if receiving sleep command, the tag will transfer into sleep state. And if interrogator commands are not received within the specified time T_5, the tag will transfer into ready state. The value of T_5 is shown in Table 157.

9.6.5.4 Secure session state

In secure session state, the interrogator makes a secure communication with individual tag after an authentication process. The tag in session state will transfer into secure session state after authentication succeeds by using authenticate command. The tag in secure session state can execute commands such as file access command, monitor command, SecureComm command, AuthComm command, KeyUpdate command, updating system password command and killing command. After a tag in secure session receives killing command, tag memory will be erased and it will transfer into killing state. If it receives sleep command, it will transfer into sleep state. And if interrogator commands are not received within the specified time T_6, it will transfer into ready state. The value of T_6 is shown in Table 157.

9.6.6 Sleep state

The sleep state is the low power state of the tag. The tag will transfer into sleep state from ready state or operation state when sleep command is received. If the tag in sense state has not received ready command or direct session command within the specified time T_1, it will transfer into sleep state. If the tag in ready state does not received the interrogator command within the specified time T_2, it will transfer into sleep state. After receiving the wake up signal, the tag in sleep state will transfer into sense state. The values of T_1 and T_2 are shown in Table 157.

9.6.7 Killing state

After the tag in session state or secure session state receives the killing command and verifies the killing password successfully, tag memory will be erased and it will transfer into killing state. The tag in killing state does not respond to any commands and always stays in killing state.

9.7 Interrogator commands and tag responses

9.7.1 Command types

Command types include mandatory commands, optional commands, special commands and custom commands.

a) Mandatory commands refer to the commands that tags and interrogators should support.

b) Optional commands refer to the commands that tags and interrogators can optionally support.

c) Special commands are used in the process of production which tags or interrogators manufacturers specifically define. All of them can be disabled permanently.

d) Custom commands refer to the commands that are defined specifically and used in specific RFID system by users.

9.7.2 Command codes list

According to the types of operation, interrogator commands include ready/sleep command, access command, collection command, file access command, monitor command and security protocol command and so on. The code, name, abbreviation, operation mode and command type of these

commands are shown in Table 158. The function and parameter of these commands and the tag's response after receiving the command will be described in the following items of this subclause. After executing interrogator command, the tag state transitions in each state are shown in Annex E.

Table 158 — Command code list

Operation type	Command code	Name	Abbreviation	Operation mode	Command type
Ready/sleep command	00_h	Reserved	—	—	—
	01_h	Ready	Ready	Broadcast/ point to point	Required
	02_h	Sleep all tags	SleepAll	Broadcast/ point to point	Required
	03_h	Sleep all tags except specific type	SleepOnes	Broadcast	Required
	04_h	Sleep all tags except specific one	SleepAllButOne	Broadcast	Required
	05_h	Direct session	DirectAccess	Broadcast/ point to point	Required
	06_h~$0F_h$	Reserved	—	—	—
Access command	10_h	Reserved	—	—	—
	11_h	Access frame initiation	AccessFrame Initiation	Broadcast	Required
	12_h	Access time-slot initiation	AccessTimeSlot Initiation	Broadcast	Required
	13_h	Access failure ACK	AccessFail	Broadcast	Required
	14_h	Access success ACK	AccessSuccess	Broadcast	Required
	15_h	Re-access	ReAccess	Broadcast	Required
	16_h~$1F_h$	Reserved	—	—	—
Collection command	20_h	Reserved	—	—	—
	21_h	Collection period initiation	CollectInitiation	Broadcast	Required
	22_h	Collection failure ACK	CollectFail	Broadcast	Required
	23_h	Collection success ACK	CollectSuccess	Broadcast	Required
	24_h~$2F_h$	Reserved	—	—	—

Table 158 *(continued)*

Operation type	Command code	Name	Abbreviation	Operation mode	Command type
File access command	30_h	Reserved	—	—	—
	31_h	Select file	SelectFile	Point to point	Required
	32_h	Verify file password	VerifyPWD	Point to point	Required
	33_h	File invalidate	Invalidate	Point to point	Required
	34_h	File validate	Validate	Point to point	Required
	35_h	List file	DirDF	Point to point	Required
	36_h	Read transparent file	ReadTransparent	Point to point	Required
	37_h	Update transparent file	UpdateTransparent	Point to point	Required
	38_h	Read record file	ReadRecord	Point to point	Required
	39_h	Update record file	UpdateRecord	Point to point	Required
	$3A_h$	Query record	QueryRecord	Point to point	Required
	$3B_h$~$4F_h$	Reserved	—	—	—
Monitor command	50_h	Reserved	—	—	—
	51_h	Monitor period configuration	MRConf	Point to point	Required
	52_h	Monitor period initiation	MRInitiation	Broadcast	Required
	53_h~$6F_h$	Reserved	—	—	—
Security protocol command	70_h~$7F_h$	Reserved	—	—	—
	80_h	Authenticate	Authenticate	Point to point	Optional
	81_h	Secure communication	SecureComm	Point to point	Optional
	82_h	Authenticate communication	AuthComm	Point to point	Optional
	83_h	Key update	KeyUpdate	Point to point	Optional
	84_h~$8F_h$	Reserved	—	—	—
Others	90_h	Reserved	—	—	—
	91_h	Killing	Kill	Point to point	Required
	92_h	Random number request	Challenge	Point to point	Required
	93_h	Update system password	UpdatePWD	Point to point	Required
	94_h	Data broadcast	DataBroard	Broadcast	Required
	95_h~$9F_h$	Reserved	—	—	—
—	$A0_h$~AF_h	Defined by the manufacturer	—	—	Special
—	$B0_h$~BF_h	Defined by the manufacturer	—	—	Customize
—	$C0_h$~FF_h	Reserved	—	—	—

If a tag receives a security protocol command specifying an unsupported CSI, an improperly formatted or not-executable message, or an improper cryptographic parameter, then the tag shall not execute the command and instead treat the command's parameters as unsupportive command (see Table 147).

9.7.3 Ready and sleep commands

9.7.3.1 Ready

The ready command is used to transfer tag into ready state by interrogators. The command can be used through the broadcast mode or point to point mode. The command format is shown in Table 159.

Table 159 — Ready command format

Command code	Command parameters	
	Channel number	Ready/sleep password
1 byte	1 byte	4 bytes

Data items in Table 159 are defined as follows:

a) Command code: 01_h.

b) Channel number: It is the tag operating channel number. Channel relationship is shown in Table 132. The lower four bits of channel sequence number are the channel number and the higher four bits are reserved.

c) Ready/sleep password: Shown in 9.5.2.

If the ready/sleep password that is stored in the tag is 00000000_h, this tag will not verify the password. And it transfers directly into ready state after switching channel. If the ready/sleep password that is stored in the tag is not 00000000_h, this tag needs to verify the password. If the password is consistent, this tag transfers into ready state after switching channel; otherwise, this tag does not switch its channel and stays in sense state. The time of tags used to switch channel should be no more than 1 ms.

Tags in sense state operate in default channel.

Tags in cycle sleep–sense operation mode will transfer into ready state after receiving the ready command when they are in sense state and stop the cycle sleep–sense operation mode. After tags transfer into sleep state, they will re-enter cycle sleep–sense operation mode.

Tags do not return response to this command.

9.7.3.2 Sleep all tags

The command of sleep all tags is used to sleep all tags within the operation distance range of interrogators. The command can be used in broadcast and point to point modes. The command format is shown in Table 160.

Table 160 — Command format of the sleep all tags

Command code	Command parameters
	Ready/sleep password
1 byte	4 bytes

Data items in Table 160 are defined as follows:

a) Command code: 02_h.

b) Ready/sleep password: Shown in 9.5.2.

If the ready/sleep password of the tag is 00000000_h, the tag will not verify the command and transfer into sleep state directly. If the ready/sleep password of the tag is not 00000000_h, the tag will verify the password. If verification is successful, the tag will transfer into sleep state. Otherwise, the tag will not execute the command and stay in the current state.

Tags do not return response to this command.

9.7.3.3 Sleep all tags except specific type

The sleep all tags except specific type command is used to sleep tags according to tag data. This command is broadcast command. The format is shown in Table 161.

Table 161 — Command format of the sleep all tags except specific type

Command code	Command parameters				
	Ready/sleep password	File identifier	Offset	Data length	Data content
1 byte	4 bytes	2 bytes	1 byte	1 byte	M byte

Data items in Table 161 are defined as follows:

a) Command code: 03_h.

b) Ready/sleep password: Shown in 9.5.2.

c) File identifier: The identifier of a tag file.

d) Offset: The beginning address of the data to be compared. Its unit is byte.

e) Data length: The amount of data bytes to be compared.

f) Data content: Data to be compared.

If data in the tag file and command parameters are inconsistent and the ready/sleep password is 00000000_h, the tag will not verify the password and transfer into the sleep state directly.

If data in the tag file and command parameters are inconsistent and the ready/sleep password is not 00000000_h, the tag will verify the password. If data in the tag file and command parameters are consistent, the tag will transfer into sleep state. Otherwise, the tag will not execute this command and stay in the current state.

Tags do not return response to this command.

9.7.3.4 Sleep all tags except specific one

The command of the sleep all tags except specific one will sleep tag according to TID. It is a broadcast command and its format is shown in Table 162.

Table 162 — Command format of the sleep all tags except specific one

Command code	Command parameters	
	TID	Ready/sleep password
1 byte	8 bytes	4 bytes

Data items in Table 162 are defined as follows:

a) Command code: 04_h.

b) TID: Refer to 9.4.6.4.

c) Ready/sleep password: Shown in 9.5.2.

After the tag receives this command, it will compare the TID in the command with its own. If they are consistent, the tag will not respond; otherwise, the tag will verify the ready/sleep password. If the ready/sleep password is 00000000_h, the tag will not verify the password and will transfer into the

sleep state directly. If the ready/sleep password is not 00000000_h, the tag will verify the password, and then if passwords are consistent, the tag will transfer into sleep state; otherwise, the tag will not execute the command and stay in the current state.

Tags do not return response to this command.

9.7.3.5 Direct session

9.7.3.5.1 Command format

Direct session command is used to transfer the tag from sense state to session state directly by the interrogator. It also specifies the response content and response time of the tag after it receives the command. The command can be used in the mode of broadcast or point to point. The command format is shown in Table 163.

Table 163 — Direct session command format

Command code	Command parameters				
	Channel number	Ready/sleep password	Response waiting time	File identifier	Sleep control
1 byte	1 byte	4 bytes	2 bytes	2 bytes	1 byte

Data items in Table 163 are defined as follows:

a) Command code: 05_h.

b) Channel number: It is the tag operating channel number. Channel relationship is shown in Table 132. The lower four bits of channel sequence number are the channel number and the higher four bits are reserved.

c) Ready/sleep password: Shown in 9.5.2.

d) Response waiting time: It is used to specify the waiting time of returning response after the tag receives this command. For more information on time unit, refer to Table 164.

e) File identifier: The identifier of a tag file.

f) Sleep control: It is used to specify whether the tag will transfer into sleep state directly after it responded to this command. 00_h means to sleep directly, and 01_h means to keep session.

Table 164 — Response waiting time

Information rate	Time unit
250 kbps	10 ms
62,5 kbps	25 ms
31,25 kbps	50 ms
15,625 kbps	100 ms

Tags in sense state operate in default channel.

If the storage ready/sleep password in the tag is 00000000_h, the tag will not verify the password. It will transfer into session state directly after switching channel. If the storage ready/sleep password in the tag is not 00000000_h, the tag will verify the password. And if the password is consistent, it will transfer into session state after switching channel. Otherwise, the tag does not switch its channel and stays in sense state.

The tag should switch to the channel which is specified by the channel number within 1 ms after receiving this command.

The tag in cycle sleep–sense operation mode will transfer into session state directly after receiving the direct session command when it is in sense state and stop the cycle sleep–sense operation mode. And after the tag transfers into the sleep state, it will re-enter cycle sleep–sense operation mode.

9.7.3.5.2 Response format

The tag will return responses to the interrogator after it receives the direct session command. The response format is shown in Table 165.

Table 165 — Tag response format

Command code	Response data			
	Execution state	File identifier	Data length	Data content
1 byte	2 bytes	2 bytes	1 byte	M bytes

Data items in Table 165 are defined as follows:

a) Command code: 05_h.

b) Execution state: Shown in 9.4.6.3.

c) File identifier: The identifier of a tag file.

d) Data length: The amount of data bytes that the tag returns.

e) Data content: The data that tag returns.

When the file identifier is specified as an empty file, the length of data that the tag returns is zero and the tag does not have the data content items. Otherwise, the tag will return all the data of the file and the maximum length of the data is 118 bytes.

9.7.4 Access commands

9.7.4.1 Access frame initiation

The command of access frame initiation is used for initiating a tag access process by the interrogator, and the command parameters include Q and N. The command format is shown in Table 166. For more information on tag access process, refer to 9.8.2.2.

Table 166 — Access frame initiation command format

Command code	Command parameters	
	Q	N
1 byte	1 byte	1 byte

Data items in Table 166 are defined as follows:

a) Command code: 11_h.

b) Q: Used to control the amount of tags in an access frame, and its value ranges from 0 to 15. In the tag access process, the interrogator can adjust the value of Q according to the tag collision status.

c) N: Represents the amount of time-slot in one access frame. It is generally fixed in the tag access process.

The tag selects the random number in the range of [0, 2^Q-1] after receiving the access frame initiation command. The tag with random number 0 randomly selects a time-slot in the range of [0, N-1]. During each time-slot, the tag accesses by the binary tree algorithm. The binary tree algorithm is further discussed in 9.9.3.

9.7.4.2 Access time-slot initiation

9.7.4.2.1 Command format

The command of the access time-slot initiation is used to initiate tag access time-slot. The tag accesses by the binary tree algorithm. The binary tree algorithm is further discussed in 9.9.3. The command format is shown in Table 167. Tag access process is further discussed in 9.8.2.2.

Table 167 — Access time-slot initiation command format

Command code	Command parameters
	SN_{slot}
1 byte	1 byte

Data items in Table 167 are defined as follows:

a) Command code: 12_h.

b) SN_{slot}: Access time-slot number.

9.7.4.2.2 Response format

After the tag receives the access time-slot initiation command, if the time-slot that the tag selected is SN_{slot}, the tag will set the counter to be 0 and start the random number generator. If the random generated number is 1, the counter adds 1. And if the random generated number is 0, the counter remains 0, and the tag returns a new RN8. The response format is shown in Table 168.

Table 168 — Access time-slot initiation command response format

Response data
1 byte

The response of the access time-slot initiation command is short frame. Its message data only include RN8.

9.7.4.3 Access failure ACK

9.7.4.3.1 Command format

The command of access failure ACK is used to notify the tag by the interrogator that it has failed to access in the current time-slot. If several tags resond to RN8 simultaneously, collision will occur to the interrogator. The interrogator receives the response, but cannot decode correctly. If collision occurs, the interrogator should send this command to tags. If the value of tag counter is 0, the tag returns RN8. The command format is shown in Table 169. Tag access process is further discussed in 9.8.2.2.

Table 169 — Access failure ACK command format

Command code
1 byte

The command code of access failure ACK is 13_h.

9.7.4.3.2 Response format

After the tag receives the access failure command, if the value of counter is not 0, the counter will add 1. And if the value of counter is FF_h, the counter remains FF_h. The tag whose counter value is 0 initiates random number generator. If the random generated number is 1, the counter adds 1. If the random

generated number is 0, the counter remains 0, and the tag sends a new RN8. The response format is shown in Table 170.

Table 170 — Access failure ACK command response format

Command code
1 byte

The response of the access failure ACK command is short frame, and its message data include only RN8.

9.7.4.4 Access success ACK

9.7.4.4.1 Command format

The command of access success ACK is used to feedback from interrogator to tag after the interrogator has successfully received the current time-slot or it has not received tag response. The command parameter contains RN8 received by the interrogator and $SN_{success}$. The command format is shown in Table 171. Tag access process is further discussed in 9.8.2.2.

Table 171 — Access success ACK command format

Command code	Command parameters	
	$SN_{success}$	RN8
1 byte	2 bytes	1 byte

Data items in Table 171 are defined as follows:

a) Command code: 14_h.

b) $SN_{success}$: The interrogator assigns a number to the tag that accessed successfully.

9.7.4.4.2 Response format

After the tag whose counter value is 0 receives the access success ACK command, if the RN8 in access success ACK command is equal to RN8 that the tag sends, it indicates that the tag accesses successfully and the tag transfers into collection state. Otherwise, it indicates that the tag accesses unsuccessfully, and the tag stays in arbitration state. The value of the tag counter remains 0, and the tag returns a new RN8. The response format is shown in Table 172.

Table 172 — Access success ACK command response format

Response data
1 byte

The response of the access success ACK command is short frame, and its message data only include RN8.

After the tag which does not send RN8 (counter is not 0) receives the access success ACK command, the value of counter subtracts 1. If the value of counter is 0, the tag will respond RN8.

9.7.4.5 Re-access

9.7.4.5.1 Command format

If the interrogator does not receive the response of the tag after the interrogator sends the access failure ACK command, the interrogator will send the re-access command in order to access the tag again. The command format is shown in Table 173. Tag access process is further discussed in 9.8.2.2.

Table 173 — Re-access command format

Code command
1 byte

Re-access command code is 15_h.

9.7.4.5.2 Response format

When the tag receives re-access command, if the value of counter is not 1, the value of counter will remain. If the value of counter is 1, the tag will initiate random number generator. For all tags where the random number generated is 1, the value of counter will remain. For all tags where the random number generated is 0, the value of counter will subtract 1. The tag whose counter value is 0 will return a new RN8 immediately. The tag response format is shown in Table 174.

Table 174 — Re-access command response format

Response data
1 byte

The response of the re-access command is short frame, and its message data only include RN8.

9.7.5 Collection commands

9.7.5.1 Collection period initiation

9.7.5.1.1 Command format

Collection period initiation command is used to initiate a collection period and a collection frame concurrently. The command format is shown in Table 175.

Table 175 — Collection period initiation command format

Command parameters	Command parameters							
	Time-slot width	RX/TX instruction	Initiation time-slot number	End of time-slot number	File identifier	Offset	Length	Delay width
1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes	2 bytes	1 byte	1 byte

Data items in Table 175 are defined as followings:

a) Command code: 21_h.

b) Time-slot width: The time distributed to a single tag for communication, and its unit is $8T_b$. T_b represents the time of an information bit period.

c) RX/TX instruction: The command is used to specify tag's timing of RX/TX in a single time-slot. RX/TX switch will execute automatically when receiving or transmitting is finished.

   1) 00_h: Transmit only. The tag only transmits in this time-slot.

   2) 01_h: Transmit-receive. The tag transmits first and then receives in this time-slot.

   3) Others: Reserved.

d) Initiation time-slot number: It is used to specify the tag that is the first response in a specific collection frame. The value is $SN_{success}$.

e) Ending time-slot number: Ending time-slot number is used to specify the tag that is the last response in a specific collection frame. The value is $SN_{success}$.

f) File identifier: The file identifier of the tag returning data in this time-slot. This file is transparent file.

g) Offset: Data initial address to be read in the file.

h) Length: The byte number to be read from offset.

i) Delay width: The initiating delay time of collection time-slot. Its unit is 10 μs. The tag can prepare response data in the time range of delay width.

After the tag receives the collection period initiation command, the tag initiates the collection time-slot after the time specified by delay width. The first time-slot number is equal to the initiation time-slot number after time-slot initiation and the following time-slot adds 1 sequentially.

9.7.5.1.2 Response format

After the tag receives the collection period initiation command, the tag whose $SN_{success}$ is between the initiation time-slot number and the ending time-slot number will respond in the specific time-slot. And the tag should respond immediately if the $SN_{success}$ is equal to the time-slot initiation number. Others will return in sequence. The tag returns the content of file specified by the file identifier in the corresponding time-slot. The tag response format is shown in Table 176.

Table 176 — Tag response format

Command code	Response data				
	Execution state	Time-slot number	File identifier	Data length	Data content
1 byte	2 bytes	2 bytes	2 bytes	1 byte	M bytes

Data items in Table 176 are defined as follows:

a) Command code: 21_h.

b) Execution state: Shown in 9.4.6.3.

c) Time-slot number: $SN_{success}$.

d) File identifier: File identifier of collection command.

e) Data length: The amount of data bytes that the tag returns.

f) Data content: The data returned by the tag.

If the file that is specified by the file identifier is an empty file, the length of data that tag returns is zero, and the tag does not have the data content items. If the file that is specified by the file identifier is not an empty file, the tag returns data according to the length specified in the collection period initiation command. If the length of file data exceeds the maximum allowed data frame length, tag will return data in the maximum allowed length. Otherwise, the tag will return data of the file real length. The command execution state is 0000_h in the above situation.

9.7.5.2 Collection failure ACK

9.7.5.2.1 Command format

The collection failure ACK is used to indicate the tag response failure in the specified time-slot in the interrogator broadcasting. The command format is shown in Table 177.

Table 177 — Command format of collection failure ACK

Command code	Command parameter		
	Time-slot number	Send instruction	File identifier
1 byte	2 bytes	1 byte	2 bytes

Data items in Table 177 are defined as follows:

a) Command code: 22_h.

b) Time-slot number: The time-slot number that the interrogator failed to receive. Its value is $SN_{success}$ of the tag.

c) Send instruction: It is used to specify whether to send response data or not after the tag receives the command. The specific definition is as follows:

   1) 00_h: Send;

   2) 01_h: Not send;

   3) Others: Reserved.

d) File identifier: It is used to specify the sending data file identifier after the tag receives the command.

9.7.5.2.2 Response format

The tag whose time-slot number is the same as the one of the command parameter decides whether to send data or not based on RX/TX instruction after it receives the collection failure ACK command. If RX/TX instruction is 00_h, the tag will send the data specified by the file identifier, and its response is the same as the response of the collection period initiation command. The format is shown in Table 176. If RX/TX instruction is 01_h, the tag will not send data and confirm collection failure in this time-slot. Then it will transfer into ready state and wait for the following access and collection operation.

9.7.5.3 Collection success ACK

The collection success ACK command is used to indicate that the interrogator has received the data sent by tag successfully in the collection period in the interrogator broadcasting. The command format is shown in Table 178.

Table 178 — Command format of the collection success ACK

Command code
1 byte

The command code of collection success ACK command is 23_h.

After the tag receives the collection success ACK command in the collection state, it will transfer into session state.

The tag will not return response to the collection success ACK command.

9.7.6 File access commands

9.7.6.1 Select file

9.7.6.1.1 Command format

The select file command is used to select a file unambiguously. The command format is shown in Table 179.

Table 179 — Select file command format

Command code	Command parameters
	File identifier
1 byte	2 bytes

Data items in Table 179 are defined as follows:

a) Command code: 31_h.

b) File identifier: The identifier of the file selected.

9.7.6.1.2 Response format

After receiving a command request, the tag will select the file according to the file identifier and return a command response. The command format is shown in Table 180.

Table 180 — Select file command response format

Command code	Response data						
	Execution state	File identifier	File type and valid information	File maximum available length	File maximum allowable number of records	File actual length	File security control attribute
1 byte	2 bytes	2 bytes	1 byte	2 bytes	2 bytes	2 bytes	1 byte

Data items in Table 180 are defined as follows:

a) Command code: 31_h.

b) Execution state: Shown in 9.4.6.3.

c) File identifier: The selected file identifier. It should be the same as the file identifier in request command.

d) File type: The selected file type. It is shown in Table 149.

e) Maximum allowed length: The maximum allowed length of the selected file to be stored. It is shown in Table 149.

f) Maximum allowed number of records: The maximum allowed number of records of the selected file to be stored. It is shown in Table 149.

g) Real length: The real length of the selected file. It is shown in Table 149.

h) Security control: The security control attributes of the selected file. It is shown in Table 149.

If the execution state in the response is successful, the response should include attribute information of the selected file; otherwise, return the execution state only.

9.7.6.2 File password verification

9.7.6.2.1 Command format

The file password verification command is used to verify the password for a password protected file. The command format is shown in Table 181.

Table 181 — File password verification command format

Command code	Command parameters	
	Operation type	Operation type
1 byte	1 byte	4 bytes

Data items in Table 181 are defined as follows:

a) Command code: 32_h.

b) Operation type: The types of passwords to be verified. The definitions are as follows:

1) 01_h: List file password.

2) 02_h: File valid or invalid password.

3) 03_h: Read transparent file password, record file password or query record password.

4) 04_h: Update transparent file password or update record file password.

5) Others: Reserved.

c) Password: File operation password corresponding to the operation type.

9.7.6.2.2 Response format

After receiving the file password verification command, tag will verify the file operation password and return execution state. The response format is shown in Table 182.

Table 182 — File password verification response

Command code	Response data
	Execution state
1 byte	2 bytes

Data items in Table 182 are defined as follows:

a) Command code: 32_h.

b) Execution state: Shown in 9.4.6.3.

9.7.6.3 Invalid file

9.7.6.3.1 Command format

The invalid file command is used to set the current selected file to be invalid. Data in the file will be erased and never be recovered. If the file is set to be invalid, the file can be listed and selected. But it will not respond to any read or write operation except the file invalidate command. The command format is shown in Table 183.

Table 183 — Invalid file command format

Command code
1 byte

Command code: 33_h.

If DF is invalid, all the sub-file of DF will be set to be invalid and be erased. If DF is invalid, the sub-file will not be listed.

9.7.6.3.2 Response format

After receiving the file invalid command, the tag will check whether the operation meets the access control conditions. If not, the tag will return an error code and stop the operation. Otherwise, the data in the file will be deleted and the file will be set to be invalid. The tag will return response. The response format is shown in Table 184.

Table 184 — Invalid file response format

Command code	Response data
	Execution state
1 byte	2 bytes

Data items in Table 184 are defined as follows:

a) Command code: 33_h.

b) Execution state: Shown in 9.4.6.3.

9.7.6.4 Valid file

9.7.6.4.1 Command format

The valid file command is used to set the current selected file to be valid. If DF is valid, this operation is only valid for the current DF and the sub-file state of DF does not change. The command format is shown in Table 185.

Table 185 — Valid file command format

Command code
1 byte

The command code of the valid file command is 34_h.

9.7.6.4.2 Response format

After receiving a valid file command, the tag will check whether the operation meets the access control conditions. If not, the tag will return an error code and stop the operation. Otherwise, the file will be set to be valid, and the tag will return response. The response format is shown in Table 186.

Table 186 — Valid file response format

Command code	Response data
	Execution state
1 byte	2 bytes

Data items in Table 186 are defined as follows:

a) Command code: 34_h.

b) Execution state: Shown in 9.4.6.3.

9.7.6.5 List file

9.7.6.5.1 Command format

The list file command is used to list the sub-files of DF, including invalid files, and return records sequentially. The command format is shown in Table 187.

Table 187 — List file command format

Command code	Command parameters	
	File offset	Mode
1 byte	1 byte	1 byte

Data items in Table 187 are defined as follows:

a) Command code: 35_h.

b) File offset: If in 04_h mode, the tag will read the file attribute information that is specified by the offset. Otherwise, the value is invalid.

c) Mode: The definitions are as follows:

1) 02_h: The next file of the current selected file.

2) 03_h: The previous file of the current selected file.

3) 04_h: Read the file that is specified by the offset.

4) Others: Reserved.

9.7.6.5.2 Response format

After receiving the list command request, the tag will check whether the operation meets the access control conditions. If not, the tag will return an error code and stop the operation. Otherwise, the tag will return the attribute information of the listed file and response. The response format is shown in Table 188.

Table 188 — List file command response format

Command code	Response data						
	Execution state	File identifier	File type and valid information	File maximum available length	File maximum allowable number of records	File actual length	File security control attribute
1 byte	2 bytes	2 bytes	1 byte	2 bytes	2 bytes	2 bytes	1 byte

Data items in Table 188 are defined as follows:

a) Command code: 35_h.

b) Execution state: Shown in 9.4.6.3.

c) File attribute information: Component of information is shown in Table 149.

9.7.6.6 Read transparent file

9.7.6.6.1 Command format

The read transparent file command is used to read data from the transparent EF after specifying the offset. The command format is shown in Table 189.

Table 189 — Read transparent file command format

Command code	Command parameters	
	Offset	Length
1 byte	2 bytes	1 byte

Data items in Table 189 are defined as follows:

a) Command code: 36_h.

b) File offset: Initial address of the data to be read in the file. The unit is byte.

c) Length: The amount of data bytes to be read beginning with the offset.

9.7.6.6.2 Response format

After receiving the read transparent file command, the tag will check whether the operation meets the access control conditions. If not, tag will return an error code and stop the operation. Otherwise, the tag will return specified length information after offset and response. The response format is shown in Table 190.

Table 190 — Read transparent file command response format

Command code	Command parameters	
	Execution state	Data
1 byte	2 bytes	M bytes

Data items in Table 190 are defined as follows:

a) Command code: 36_h.

b) Execution state: Shown in 9.4.6.3.

c) Data: File data returned.

9.7.6.7 Update transparent file

9.7.6.7.1 Command format

The update transparent file command is used to update data in the transparent EF after specifying the offset. The command format is shown in Table 191.

Table 191 — Update transparent file command format

Command code	Command parameters		
	Offset	Length	Data
1 byte	2 bytes	1 byte	M bytes

Data items in Table 191 are defined as follows:

a) Command code: 37_h.

b) File offset: Initial address of the data to be updated in the file. The unit is byte.

c) Length: The amount of data bytes to be read beginning with the offset.

d) Data: Data to be updated.

9.7.6.7.2 Response format

After receiving the update transparent file command, the tag will check whether the operation meets the access control conditions. If not, the tag will return an error code and stop the operation. Otherwise, the tag updates the data specified by the command and responds. Response format is shown in Table 192.

Table 192 — Update transparent file command response format

Command code	Response data
	Execution state
1 byte	2 bytes

Data items in Table 192 are defined as follows:

a) Command code: 37_h.

b) Execution state: Shown in 9.4.6.3.

9.7.6.8 Read record file

9.7.6.8.1 Command format

The read record file command is used to read a record in the recording EF. The command format is shown in Table 193.

Table 193 — Read record file command format

Command code	Command parameters			
	Record pointer	Mode	Offset	Length
1 byte	1 byte	1 byte	1 byte	1 byte

Data items in Table 193 are defined as follows:

a) Command code: 38_h.

b) Record pointer means a sequence number of the records in the record file. The value of record pointer is from 00_h, which means the first record in a record file. If in 04_h or 14_h mode, the tag will read the record specified by the pointer. Otherwise, the value is invalid.

c) Mode: Definitions are as follows:

1) 02_h: The next record of the file.

2) 03_h: The previous record of the file.

3) 04_h: The specified record of the file.

4) 12_h: The next record of the query result file.

5) 13_h: The previous record of the query result file.

6) 14_h: The specified record of the query result file.

7) Others: Reserved.

d) Offset: Initial address of the data to be read in the record. The unit is byte.

e) Length: The amount of data bytes to be read.

9.7.6.8.2 Response format

After receiving the read record file command, the tag will check whether the operation meets the access control conditions. If not, the tag will return an error code and stop the operation. Otherwise, the tag will return the corresponding record and response. The response format is shown in Table 194.

Table 194 — Read record file command response format

Command code	Command parameters	
	Execution state	Data
1 byte	2 bytes	M bytes

Data items in Table 194 are defined as follows:

a) Command code: 38_h.

b) Execution state: Shown in 9.4.6.3.

c) Data: Record data returned.

9.7.6.9 Update record file

9.7.6.9.1 Command format

The update record file command is used to update a record in the record EF. The command format is shown in Table 195.

Table 195 — Update record file command format

Command code	Command parameters				
	Record pointer	Mode	Offset	Length	Data
1 byte	2 bytes	1 byte	2 bytes	1 byte	M bytes

Data items in Table 195 are defined as follows:

a) Command code: 39_h.

b) Record pointer means a sequence number of the records in the record file. The value of record pointer is from 00_h, which means the first record in a record file. If in 04_h mode, the tag will read the record specified by the pointer. Otherwise, the value is invalid.

c) Mode: Definitions are as follows:

1) 02_h: The next record of the original file.

2) 03_h: The previous record of the original file.

3) 04_h: Update the specified record of the original file.

4) Others: Reserved.

d) Offset: The address of the data to be updated in the record.

e) Length: The amount of data bytes to be updated.

f) Data: Data to be updated.

9.7.6.9.2 Response format

After receiving the update record file command, the tag will check whether the operation meets the access control conditions. If not, the tag will return an error code and stop the operation. Otherwise, the tag will update and return response. The response format is shown in Table 196.

Table 196 — Update record file command response format

Command code	Response data
	Execution state
1 byte	2 bytes

Data items in Table 196 are defined as follows:

a) Command code: 39_h.

b) Execution state: Shown in 9.4.6.3.

9.7.6.10 Query record

9.7.6.10.1 Command format

The query record command is used to query the matching record in the recording EF. Users can read the query records in sequence by the read record command. The command format is shown in Table 197.

Table 197 — Query record command format

Command code	Command parameters
	Query condition
1 byte	M bytes

Data items in Table 197 are defined as follows:

a) Command code: $3A_h$.

b) Query condition: The query condition consists of query elements. It is a filtering condition shown in 9.7.6.10.2.

9.7.6.10.2 Query condition

Query condition consists of multiple query elements. Query element format is shown in Table 198.

Table 198 — Query element format

Logical operator	Logical operation number			
	Field offset	Relation operator	Data length	Data
1 byte	1 byte	1 byte	1 byte	M bytes

Data items in Table 198 are defined as follows:

a) Logical operator: The value can be one of 'C', 'A' and 'O'.

1) 'C': Ignore all previous query elements, delete the existing query results and start a new query.

2) 'A': The relationship between the current query element and the previous one is 'AND'.

3) ‘O’: The relationship between the current query element and the previous one is ‘OR’.

b) Field offset: The address of the queried field in the record. The unit is byte.

c) Relation operator: The value can be one of the operators: ‘>’, ‘<’, ‘=’, ‘!’, ‘G’ (>=) and ‘L’ (<=).

d) Data length: The amount of data bytes in the data field of the query elements.

e) Data: Data in the query elements.

A query condition consists of several query elements based on the following rules: the priorities of each query element are the same, and they will be calculated left-associatively. The coding of the logical operators and relation operators above shall comply with ISO/IEC 646.

EXAMPLE

The record EF file consists of ‘NAME’, ‘SEX’, ‘AGE’ and ‘CLASS’ fields. The query condition (‘NAME’ equals to ‘Tom’ and ‘AGE’ is greater than ‘12’) consists of the following query elements:

(C(0=Tom)A(2>12))

a) 0: The offset of the ‘NAME’ filed in the record.

b) 2: The offset of the ‘AGE’ filed in the record.

9.7.6.10.3 Response format

After receiving a query record command request, the tag will check whether the operation meets the access control conditions. If not, the tag will return an error code and stop the operation. Otherwise, the tag will execute the query operation and return the response. The response format is shown in Table 199.

Table 199 — Query record command response format

Command code	Response data	
	Execution state	Record number queried
1 byte	2 bytes	2 bytes

Data items in Table 199 are defined as follows:

a) Command code: $3A_h$.

b) Execution state: Shown in 9.4.6.3.

c) Record number queried: The record number meeting the querying condition.

Query results are stored in query result file of which the tag file identifier is $F000_h$. The tag will automatically select this file and not need to provide the operation permissions after query is completed. The record of this file has only one field with 2 byte length. It will store the record pointer which matches the query condition. The record pointer is sorted in ascending order. The interrogator can read the query results by sending read record file command directly.

9.7.7 Monitor commands

9.7.7.1 Monitor period configuration

9.7.7.1.1 Command format

The monitor period configuration command is used to configure relevant parameter by interrogator when the tag operates in the monitor mode. The command format is shown in Table 200.

Table 200 — Monitor period configuration command format

Command code	Command parameter								
	File identifier	Offset	Data length		Reporting period	Monitor period		Total transmission time-slot	RX/TX instruction
1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	1 byte

Date items in Table 200 are defined as follows:

a) Command code: 51_h.

b) File identifier: File identifier of the monitoring data to be reported.

c) Offset: The address of the monitor data to be reported. The unit is byte.

d) Data length: Monitor data length to be reported.

e) Time-slot number: Time-slot number allocated to the monitor data to be reported in the monitor period.

f) Reporting period: Reporting period of monitor data to be reported. MSB indicates the unit of time. Its unit is second if its value is 0_b, and its unit is minute if its value is 1_b. The reporting period shall be an integral multiple of the monitoring period.

g) Monitoring period: Monitoring period length. MSB indicates the unit of time. Its unit is second if its value is 0_b, and its unit is minute if the value is 1_b.

h) Time-slot width: Transmission time length of each monitor data to be reported. Its unit time width is $8T_b$, where T_b represents the time of an information bit period.

i) Total transmission time-slot: The amount of monitor data to be reported in a monitor period.

j) RX/TX instruction: The tag RX/TX timing in the time-slot. RX/TX switching will execute automatically when receiving or transmitting file is completed. The explanation of RX/TX timing is as follows:

   1) 00_h: Transmit only. The tag will send data in the time-slot only.

   2) 01_h: Transmit-receive. The tag will send the data first and then transfer into the receive state in the time-slot.

   3) Others: Reserved.

9.7.7.1.2 Response format

After the tag receives this command, it will configure the relevant monitor parameters based on the command parameters and return the execution state. The response format is shown in Table 201.

Table 201 — Monitoring period configuration command response format

Command code	Command code
	Execution state
1 byte	2 bytes

Data items in Table 201 are defined as follows:

a) Command code: 51_h.

b) Execution state: Shown in 9.4.6.3.

9.7.7.2 Monitor period initiation

9.7.7.2.1 Command format

The monitor period initiation command is used to synchronize all tags and start data transmission time-slot. The response format is shown in Table 202.

Table 202 — Monitoring period initiation command format

Command code	Command parameter		
	Initiation time-slot number	Ending of time-slot number	Delay width
1 byte	2 bytes	2 bytes	1 byte

Data items in Table 202 are defined as follows:

a) Command code: 52_h.

b) Initiation time-slot number: It is used to specify the time-slot number of the tag that responds firstly.

c) Ending time-slot number: It is used to specify the time-slot number of the tag that responds last.

d) Delay width: Delay time of monitoring time-slot initiation. Its unit time width is 10 μs. The tag can prepare response data in the time range of delay width.

After the tag receives the monitor period initiation command, the tag initiates the monitor time-slot after the time specified by delay width. The first time-slot number is equal to the initiation time-slot number after time-slot initiation and the following time-slot adds 1 sequentially.

9.7.7.2.2 Response format

After the tag receives the monitor period initiation command, the tag will report the specified contents of the file identifier in the specified time-slot. The tag response format is shown in Table 203.

Table 203 — Tag response format

Command code	Response data				
	Execution state	Time-slot number	File identifier	Data length	Data content
1 byte	2 bytes	2 bytes	2 bytes	1 byte	M bytes

Data items in Table 203 are defined as follows:

a) Command code: 52_h.

b) Execution state: Shown in 9.4.6.3. If the offset address in the file monitor configuration command reaches or exceeds the ending address of file, the tag will return the data overflow error.

c) Time-slot number: Time-slot number assigned to the current tag in monitor period configuration command.

d) File identifier: File identifier in monitor period configuration command.

e) Length: The amount of data bytes in data content field.

f) Content: Data returned by tag.

If the file that is specified by the file identifier is an empty file, the length of data that tag returns is zero, and the tag does not have the data content items. If the file that is specified by the file identifier is not an empty file, the tag returns data according to the length specified in the monitor period initiation command. If the length of file data exceeds the maximum allowed data frame length, the tag will return

data in the maximum allowed length. Otherwise, the tag will return data of the file real length. The command execution state is 0000_h in the above situation.

9.7.8 Security protocol commands

9.7.8.1 Authenticate command

9.7.8.1.1 Command format

Interrogators and tags shall implement the authenticate command as shown in Table 204.

Table 204 — Authenticate command

Command code	RFU	CSI	Length	Message	RN16	CRC-16
1 byte	1 byte	1 byte	EBV	Variable	2 bytes	2 bytes

Data items in Table 204 are defined as follows:

a) Command code: 80_h.

b) CSI: Selects the cryptographic suite that the tag and interrogator use for the authentication as well as for all subsequent communications.

c) Length: Length of message.

d) Message: Message includes parameters for the authentication.

An interrogator shall use authenticate commands to perform mutual authentication. The CSI specified in the authenticate command selects a particular cryptographic suite from among those supported by the tag. The number and encoding of the authenticate commands required to implement the authentication depend on the chosen cryptographic suite.

9.7.8.1.2 Response format

The fast response reply to an authenticate command is shown in Table 205.

Table 205 — Fast response reply to an authenticate command

Command code	Header	Length	Response	RN16	CRC-16
1 byte	1 bit	EBV	Variable	2 bytes	2 bytes

Data items in Table 205 are defined as follows:

a) Command code: 80_h.

b) Length: Length of response.

c) Response: The response for authenticate command.

9.7.8.2 Secure communication command

9.7.8.2.1 Command format

After a tag has cryptographically authenticated an interrogator, it may accept subsequent commands encapsulated in the SecureComm command shown in Table 206. The message field in a SecureComm may encapsulate an encrypted tag-supported command. Before encapsulating a command, an interrogator shall remove the preamble, handle, and CRC from the command. An interrogator does not include a MAC1 in the SecureComm.

Table 206 — SecureComm command

Command code	CSI	Length	Message	RN16	CRC-16
1 byte	1 byte	EBV	Variable	2 bytes	2 bytes

Data items in Table 206 are defined as follows:

a) Command code: 81_h.

b) CSI: Selects the cryptographic suite that the tag and interrogator use for the authentication as well as for all subsequent communications.

c) Length: Length of message.

d) Message: The encrypted message.

SecureComm allows an interrogator to read data from and write data to a tag while preventing an eavesdropper from decrypting the communications. It allows configuring a tag's secure features by writing to a memory location that configures these features. The authenticate communication is not supported.

9.7.8.2.2 Response format

The response reply to a SecureComm command is shown in Table 207.

Table 207 — Response reply to a SecureComm command

Command code	Header	Length	Response	RN16	CRC-16
1 byte	1 bit	EBV	Variable	2 bytes	2 bytes

Data items in Table 207 are defined as follows:

a) Command code: 81_h.

b) Length: Length of response.

c) Response: The encrypted response.

9.7.8.3 AuthComm command

9.7.8.3.1 Command format

After a tag has cryptographically authenticated an interrogator, it may accept subsequent commands encapsulated in the AuthComm command shown in Table 208.

Table 208 — AuthComm command

Command code	CSI	Length	Message	MAC1	RN16	CRC-16
1 byte	1 byte	EBV	Variable	16 bytes	2 bytes	2 bytes

Data items in Table 208 are defined as follows:

a) Command code: 82_h.

b) CSI: Selects the cryptographic suite that the tag and interrogator use for the authentication as well as for all subsequent communications.

c) Length: Length of message.

d) Message: Message includes parameters for authenticate communication.

e) MAC1: Message authentication code, computed over bits in the 'Message'.

The message field in an AuthComm shall encapsulate a tag-supported command. Before encapsulating a command, an interrogator shall remove the preamble, handle, and CRC from the command. An interrogator includes a MAC1 in the AuthComm.

9.7.8.3.2 Response format

The response reply to an AuthComm command is shown in Table 209.

Table 209 — Response reply to an AuthComm command

Command code	Header	Length	Response	MAC1	RN16	CRC-16
1 byte	1 bit	EBV	Variable	16 bytes	2 bytes	2 bytes

Data items in Table 209 are defined as follows:

a) Command code: 82_h.

b) Length: Length of response.

c) Response: The response for AuthComm command.

d) MAC1: Message authenticate code, computed over bits in the 'Message'.

9.7.8.4 KeyUpdate command

9.7.8.4.1 Command format

Interrogators and tags may implement the KeyUpdate command; if they do, they shall implement it as shown in Table 210.

Table 210 — KeyUpdate command

Command code	RFU	Key ID	Length	Message
1 byte	1 byte	5 bits	EBV	Variable

Data items in Table 210 are defined as follows:

a) Command code: 83_h.

b) Key ID: Specifies the key to be written or updated.

c) Length: Length of key.

d) Message: Is or contains the key. Message may contain other parameters (such as a MAC1) as specified by the cryptographic suite. Message may be encrypted.

KeyUpdate allows an interrogator to overwrite a key stored in a tag. A KeyUpdate command shall always be encapsulated in a SecureComm.

9.7.8.4.2 Response format

The fast response reply to a KeyUpdate command is shown in Table 211.

Table 211 — Response reply to a KeyUpdate command

Command code	Header	RN16	CRC-16
1 byte	1 bit	2 bytes	2 bytes

Data items in Table 211 are defined as follows:

a) Command code: 83_h.

9.7.9 Other commands

9.7.9.1 Killing command

9.7.9.1.1 Command format

The killing command is used to disable tag permanently. The command format is shown in Table 212.

Table 212 — Killing command format

Command code	Command parameters
	Killing password
1 byte	4 bytes

Data items in Table 212 are defined as follows:

a) Command code: 91_h.

b) Killing password: Its length is 4 bytes.

After receiving the killing command, if the killing password is 00000000_h, the tag will ignore the command and not execute the killing operation. Otherwise, the tag will verify the password, and then if the verification is successful, all tag data will be erased, and the tag responds successfully and transfers into killing state. If the verification is a failure, the killing operation will not be executed, and the tag will return the error code data. The tag in killing state will not respond to any command from the interrogator.

9.7.9.1.2 Response format

The response format of killing command is shown in Table 213.

Table 213 — Killing command response format

Command code	Response data
	Execution state
1 byte	2 bytes

Data items in Table 213 are defined as follows:

a) Command code: 91_h.

b) Execution state: Shown in 9.4.6.3.

The tag executes the killing operation in session state or secure session state.

9.7.9.2 Random number request

9.7.9.2.1 Command format

The random number request command is used to request random number from interrogators to tags. The command format is shown in Table 214.

Table 214 — Random number request command format

Command code	Command parameters
	Length of random number
1 byte	1 byte

Data items in Table 214 are defined as follows:

a) Command code: 92_h.

b) Length of random number:

1) 00_h: Reserved.

2) 01_h: Return an 8-bit random number.

3) 02_h: Return a 16-bit random number.

4) 03_h: Return a 32-bit random number.

5) 04_h: Return a 64-bit random number.

6) 05_h~FF_h: Reserved.

9.7.9.2.2 Response format

After receiving the random number request command, the tag will generate a random number with specified bit number and return response. The response format is shown in Table 215.

Table 215 — Random number request command response format

Command code	Response data	
	Execution state	Random number
1 byte	2 bytes	M bytes

Data items in Table 215 are defined as follows:

a) Command code: 92_h.

b) Executing state: Shown in 9.4.6.3.

c) Random number: Random number returned from tag.

9.7.9.3 Update system password command

9.7.9.3.1 Command format

The update system password command is used to update the killing password and the ready/sleep password. It can also update access password of the user section file. The command format of the update system password is shown in Table 216.

Table 216 — Update system password command format

Command code	Command parameters			
	Administrator password	Index of password to be updated	Mode	New password
1 byte	4 bytes	2 bytes	1 byte	4 bytes

Data items in Table 216 are defined as follows:

a) Command code: 93_h.

b) Administrator password: Its length is 4 bytes.

c) Index of password to be updated: Its length is 2 bytes. Defined as follows:

   1) 0001_h: Killing password.

   2) 0002_h: Ready/sleep password.

   3) $3F00_h$~$FFFF_h$: File identifier, the index of the file access password based on the mode parameter.

   4) Others: Reserved.

d) Mode: Its length is 1 byte. If the index of password to be updated is the file identifier, the data item is valid. Defined as follows:

   1) 01_h: Password of list files.

   2) 02_h: Valid or invalid password of files.

   3) 03_h: Password of reading transparent files, password of reading a record file or password of querying a record.

   4) 04_h: Password of updating a transparent or a record file.

   5) Others: Reserved.

e) New password: Its length is 4 bytes.

Update system password command can be executed in session state or secure session state.

9.7.9.3.2 Response format

When the tag in secure session state receives the update system password command, it will return the response according to the execution status. The response format is shown in Table 217.

Table 217 — Update system password command response format

Command code	Response data
	Execution state
1 byte	2 bytes

Data items in Table 217 are defined as follows:

a) Command code: 93_h.

b) Execution state: Shown in 9.4.6.3.

9.7.9.4 Data broadcast

9.7.9.4.1 Command format

The data broadcast command is used to send the broadcast data from interrogators to tags in RF fields in ready state. The command format is shown in Table 218.

Table 218 — Data broadcast command format

Command code	Command parameters	
	Ready password	Broadcast data
1 byte	4 bytes	M bytes

Data items in Table 218 are defined as follows:

a) Command code: 94_h.

b) Ready/sleep password: Shown in 9.5.2.

c) Broadcast data: The interrogator broadcast data.

If the ready/sleep password of tag is 00000000_h, the tag will receive the broadcast data without password verification. Otherwise, after receiving the data broadcast command and completing the password verification, the tag can receive the data.

The tag does not respond to this command.

After the tag receives the broadcast data, the tag process method is not specified in this document. It can be specified by application.

9.8 Protocol operation mode

9.8.1 Overview

In this document, the RFID system uses the interrogator speaking first operation mode. The interrogator will always send commands firstly in each communication process and the tag will respond to the interrogator commands. The tag will not send a message until it receives an interrogator command.

9.8.2 Operation process

9.8.2.1 Ready stage

The tag with the external wake-up mechanism will send the wake-up signal with certain duration to the sleeping tag by the interrogator or other external wake-up sources in order to wake them up. The tag with the internal cycle self wake-up mechanism will wake themselves up by the internal timer. And the waked up tag will transfer into sense state and intercept the interrogator command. The interrogator will send ready command for the tag transferring into the ready state and completing the ready process.

9.8.2.2 Access stage

The interrogator will execute a binary tree algorithm with data frame format to complete the tag access. There might be one or more access frames in one access stage. The interrogator will send the access frame initiation command to initiate an access frame. When the interrogator accesses a tag successfully, it will return an ACK message to the tag, and this ACK message consists of a successful access sequence number which is assigned by the interrogator. This sequence number is the collection time-slot number in the collection stage of the tag. The short frame data format is used in the access stage. The time-slot of access stage is shown in Figure 56.

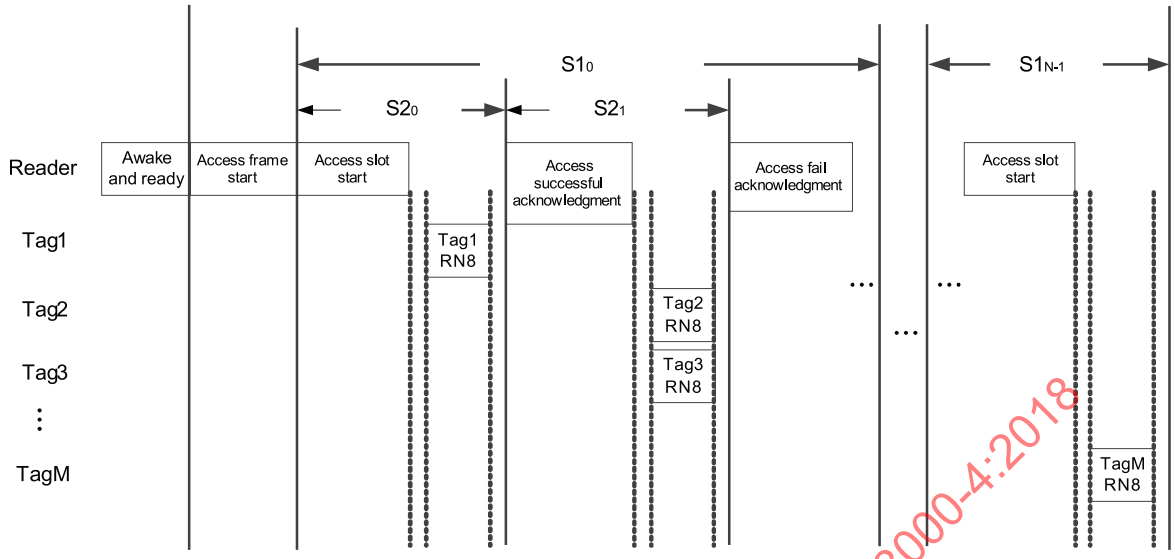


Figure 56 — Time-slot diagram in access stage

Time-slot in Figure 56 is described as follows:

a) An access frame is composed of N time-slots of S1. The interrogators select tags of the same access time-slot by a binary tree algorithm in each time-slot.

b) S2 represents an interactive time-slot between interrogators and tags. An interaction consists of sending an access time-slot initiation command, an access success ACK command or an access failure ACK command by interrogator, and RN8 returned by tags.

9.8.2.3 Collection stage

The interrogator will collect the data of accessed tags according to time-slot sequentially. After receiving the interrogator initiation command in collection stage, the tag will send the data requested by the interrogator according to the sequence number of time-slot obtained in the access stage. The time-slot in collection stage is shown in Figure 57.

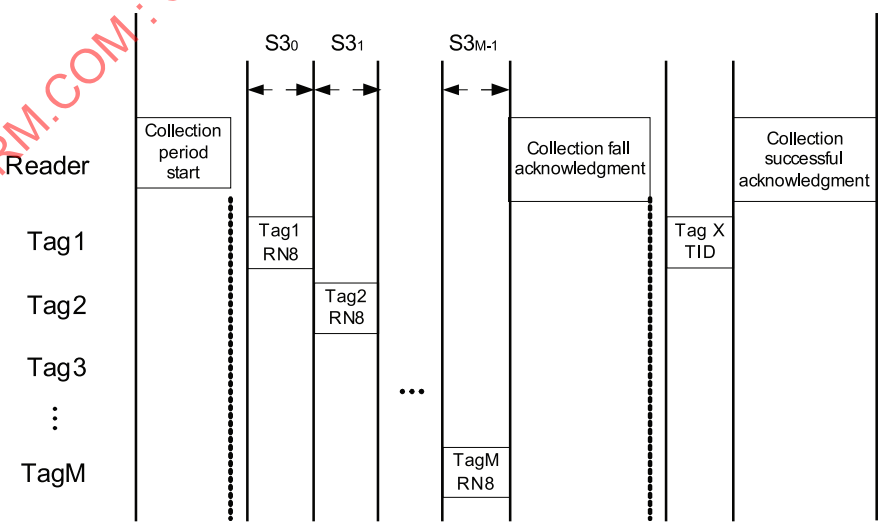


Figure 57 — Time-slot diagram in collection stage

S3 in Figure 57 represents the time-slot in collection stage. Before the initiation of collection stage, if there are M tags to be accessed successfully, and every tag is assigned an $SN_{success}$ by a certain order,