INTERNATIONAL STANDARD

ISO/IEC 18000-4

Third edition
2015-02-01

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 18000-4:2015(E)

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

Published in Switzerland

Contents

Page

Foreword

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

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

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

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

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

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

This third edition cancels and replaces the second edition (ISO/IEC 18000-4:2008), of which it constitutes a minor revision with the following changes:

— 5.1 has become Clause 5;

— 5.2 has become Clause 6;

— 5.3 has been Clause 7;

— Clause 8 has been introduced;

— Clause 6 has become Clause 9;

— Clause 1, Clause 2, Clause 3, Clause 4, Clause 5, and Clause 9 have been revised as necessary to also cover Clause 8.

ISO/IEC 18000 consists of the following parts, under the general title *Information technology — Radio frequency identification for item management*:

— *Part 1: Reference architecture and definition of parameters to be standardized*

— *Part 2: Parameters for air interface communications below 135 kHz*

— *Part 3: Parameters for air interface communications at 13,56 MHz*

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

— *Part 6: Parameters for air interface communications at 860 MHz to 960 MHz General*

— *Part 61: Parameters for air interface communications at 860 MHz to 960 MHz Type A*

- *Part 62: Parameters for air interface communications at 860 MHz to 960 MHz Type B*
- *Part 63: Parameters for air interface communications at 860 MHz to 960 MHz Type C*
- *Part 64: Parameters for air interface communications at 860 MHz to 960 MHz Type D*
- *Part 7: Parameters for active air interface communications at 433 MHz*

Introduction

This part of ISO/IEC 18000 is one of a series of International Standards and Technical Reports developed by ISO/IEC JTC 1/SC 31, WG 4 for the identification of items (item management) using radio frequency identification (RFID) technology.

This part of ISO/IEC 18000 defines three 2,45 GHz protocols. Each of the specific physical/data link configurations is defined in a separate sub-clause. 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 can involve the use of patents concerning radio-frequency identification technology given in all parts of the 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 can be obtained from the following companies.

Contact details

Patent Holder:

Legal Name iControl Inc

Contact for license application:

Name & Department George Cavage

Address 3235 Kifer Road, suite 260

Address 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 license application:

Name & Department Stacy Jones, Impinj, Inc.

Address 701 N 34th Street, Suite 300

Address 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

Patent Holder:

Legal Name Zebra Technologies Corporation

Contact for license application:

Name & Department James O'Hagan, Director of Patents & Technology

Address 475 Half Day Road, Suite 500

Address Lincolnshire, IL 60069, USA

Tel. +1 (847) 793-6798

Fax +1 (847) 955-4514

E-mail johagan@zebra.com

URL (optional)

Information technology — Radio frequency identification for item management —

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

1 Scope

This part of ISO/IEC 18000 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 part of ISO/IEC 18000 provides a common technical specification for RFID devices that can be used by ISO committees developing RFID application standards. This part of ISO/IEC 18000 is intended to allow for compatibility and to encourage inter-operability of products for the growing RFID market in the international marketplace. This part of ISO/IEC 18000 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 part of ISO/IEC 18000 further defines the communications protocol used in the air interface.

This part of ISO/IEC 18000 contains the following three 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, among others, the logistics and transportation industry; agnostic to any device, commercial or otherwise, requiring global availability.

The detailed technical differences between the modes are shown in the parameter tables.

2 Normative references

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

ISO/IEC 7816-6:—[1], *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/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 19762 (all parts):—[1], *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary*

[1] To be published

ISO/IEC/IEEE 8802-15-4:2010, *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)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762 (all parts):—[2] and the following apply.

3.1
associated
has successfully negotiated a di-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 International Standard

3.3
block cipher
cryptographic function that operates on strings of fixed size

3.4
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.

3.5
data server
device data termination point

3.6
device
any entity that meets the ISO/IEC/IEEE 8802-15-4 medium access control (MAC), physical interface to the wireless medium, and this protocol specification.

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

3.7
encryption
transformation of a message into a new representation so that privileged information is required to recover the original representation

3.8
full-function device
FFD
device capable of operating as a coordinator

3.9
group key
key that is known only to a set of devices

[2] To be published.

3.10
hailing channel
ISO/IEC/IEEE 8802-15-4 radio channel used to broadcast the NDB

3.11
key
privileged information that can 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

3.12
link key
secret key that is shared between precisely two devices

3.13
mesh networking
type of network where an FFD serves as a relay for other devices

3.14
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

3.15
network channel
primary ISO/IEC/IEEE 8802-15-4 radio channel between a coordinator and remote devices

3.16
packet
formatted, aggregated bits that are transmitted together in time across the physical medium

3.17
payload data
contents of a data message that is being transmitted

3.18
reduced-function device
RFD
device that is not capable of operating as a coordinator

3.19
server connected coordinator
SCC
network coordinator that terminates the wireless protocol described in this International Standard and is connected to control servers

3.20
tag
any device type associated with the device and capable of joining the network

4 Symbols and abbreviated terms

ACK	acknowledgement
CCITT	Comité Consultatif International Téléphonique et Télégraphique
Cht	Carrier high level tolerance
Clt	Carrier low level tolerance

CRC	cyclical redundancy check
CSMA	carrier sense multiple access
$f_{bitrate}$	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
M	Modulation
Ma	Modulation overshoot
MAC	medium access control
Mb	Modulation undershoot
MIC	message integrity code
MIN	Manufacturing Identification Number
Mlt	Modulation lower tolerance
Mut	Modulation upper tolerance
NAK	no-acknowledgement
NDB	Network Discovery Beacon
NSM	Network Status Message
QPSK	quad-phased shift keying
RTLS	real time locating system
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
TTL	tag talk last

5 General items on 2,45 GHz RFID protocols that support this part of ISO/IEC 18000

5.1 Protocols

Clause 5 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 and 8.

5.2 Frequency

This part of ISO/IEC 18000 is intended to address RFID devices operating in the 2 450 MHz Industrial, Scientific and Medical (ISM) frequency band.

5.2.1 Interface definitions

This part of ISO/IEC 18000 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 one meter.

5.2.1.1 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.

5.2.1.2 Minimum features

RFID systems defined by this part of ISO/IEC 18000 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.

5.2.1.3 Conformance

To claim conformance with this part of ISO/IEC 18000, an RFID system shall comply with one of the physical/data link implementations described in Clause 6, 7 and 8.

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 TTF (Tag Talks First) protocols. This part of ISO/IEC 18000 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 and in sub-clause 8.5.1 for mode 3.

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 part of ISO/IEC 18000.

— 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 minimise 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 is 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 the International Standard 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 is applied to the forward link (interrogator-to-tag) data.

— FM0 bit-wise encoding and CRC-16 packet-level protection is 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.

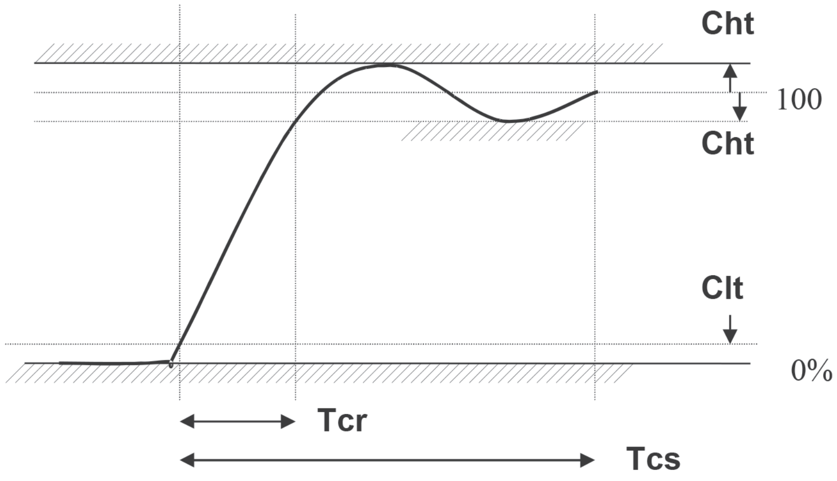


Figure 1 — Interrogator power-up waveform

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.

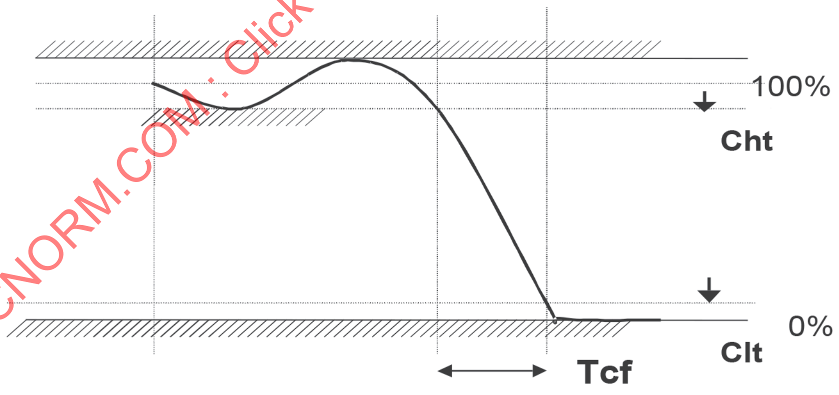


Figure 2 — Interrogator power-down waveform

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 mode (FHSS), the carrier rise and fall times shall conform to the characteristics specified in Figure 3 and Table 3.

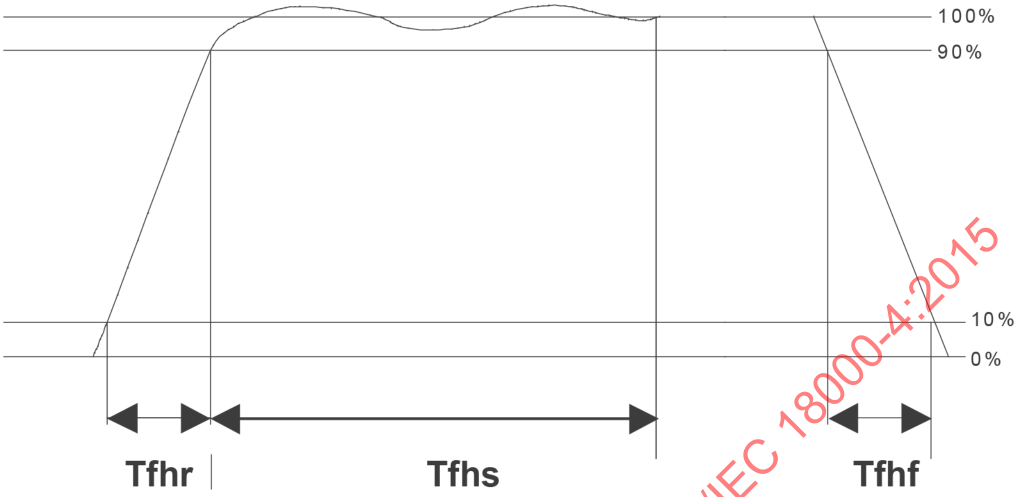


Figure 3 — FHSS carrier rise and fall characteristics

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.

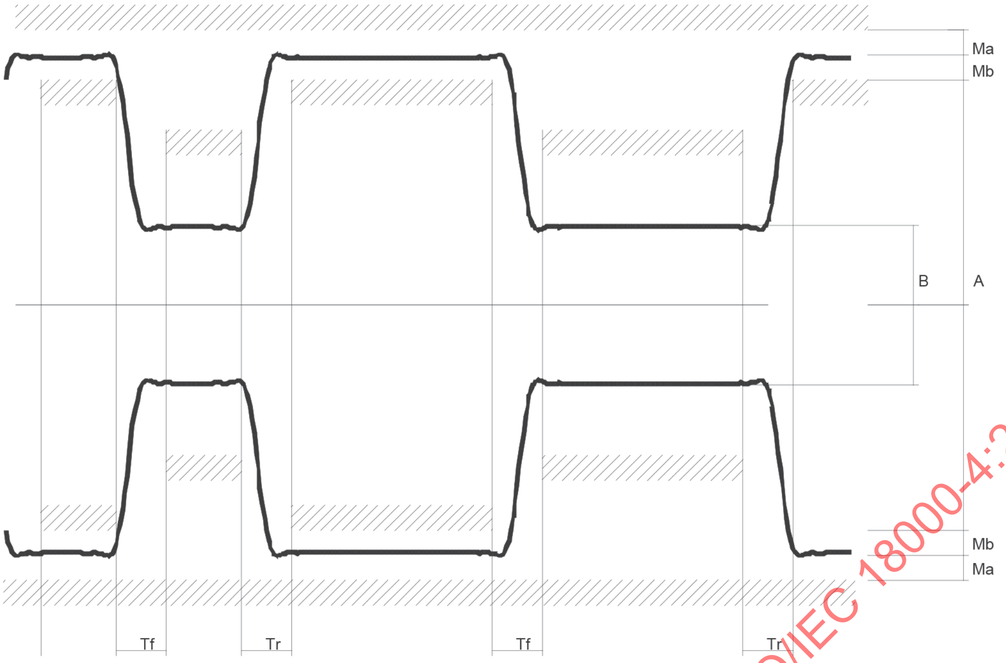


Figure 4 — Example of 40 kbit/s signal

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 is Manchester encoded as per Figure 5.

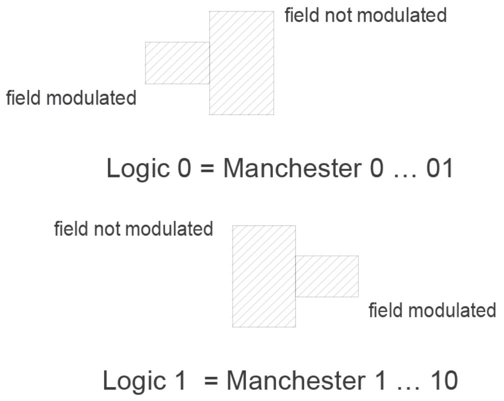


Figure 5 — Forward link bit coding

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 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 is 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 - 40 kbit/s	25 µs - 33 µs	± 15 %

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

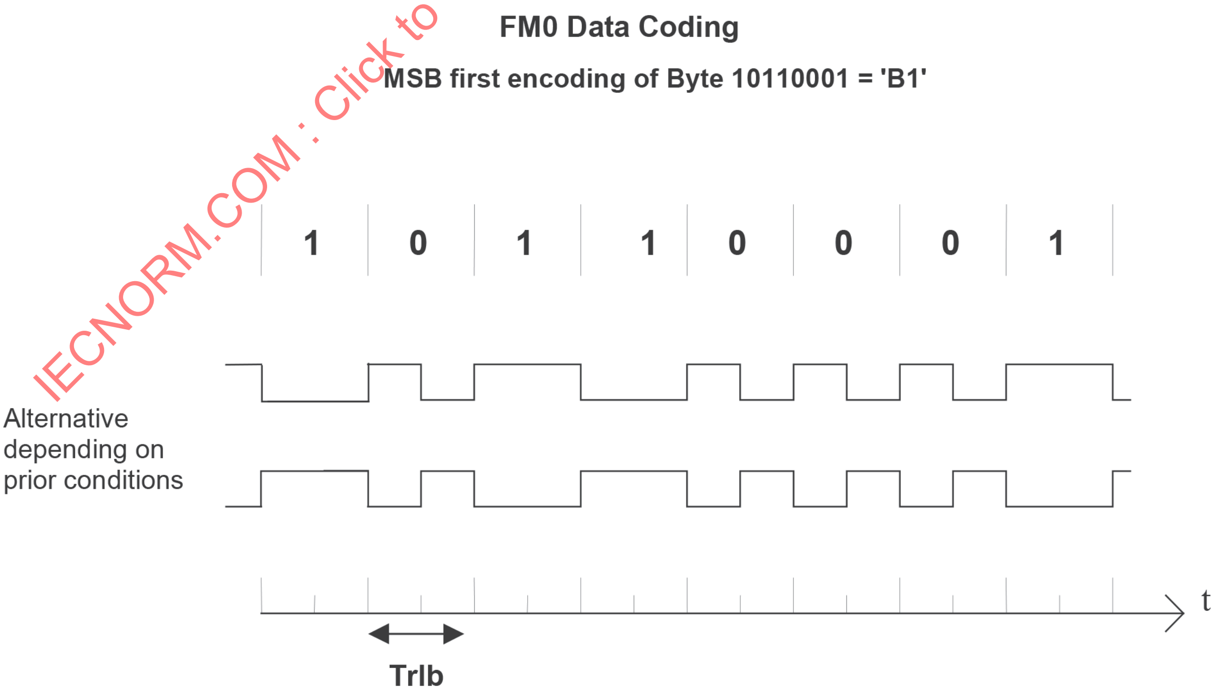

FM0 Data Coding
MSB first encoding of Byte 10110001 = 'B1'
1
0
1
1
0
0
0
1
Alternative depending on prior conditions
t
Trlb

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

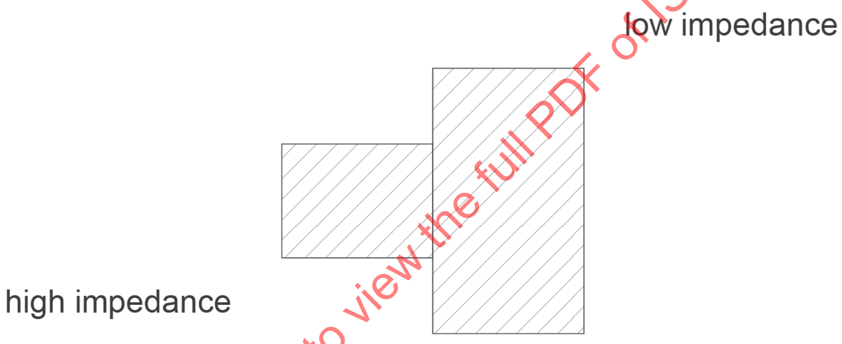
The return preamble is a sequence of backscatter modulation specified in Table 6.

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 bits 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 zeros.

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.

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 is encoded and presented in slightly different ways in the constituent fields. For interrogator-to-tag communication (forward link), data is 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 is 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 consist of the following fields:

- Preamble
- Delimiter
- Command
- Parameter and data files
- CRC

Table 8 — General command format

Preamble Detect	Preamble	Delimiter	Command	Parameter	Data	CRC

6.2.8.2 Preamble detect field

The preamble detect field consist 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 clauses 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

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 $16 * T_{rlb} - 0,75 * T_{flb}$. The duration of the quiet time is determined by the communication speed of the forward and return link.

6.2.9.3 CRC

See clauses 6.2.6 and Annex B.

6.2.10 WAIT

During the WAIT field, the interrogator provides steady power to the tag for 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 must 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.

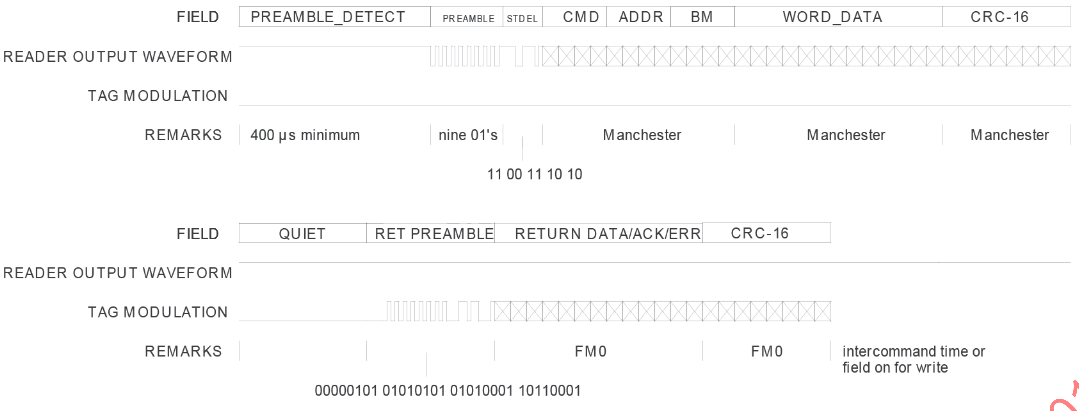


Figure 8 — Sample Command/Response Packets for (GROUP_SELECT) (40 kbit/s on forward and return link)

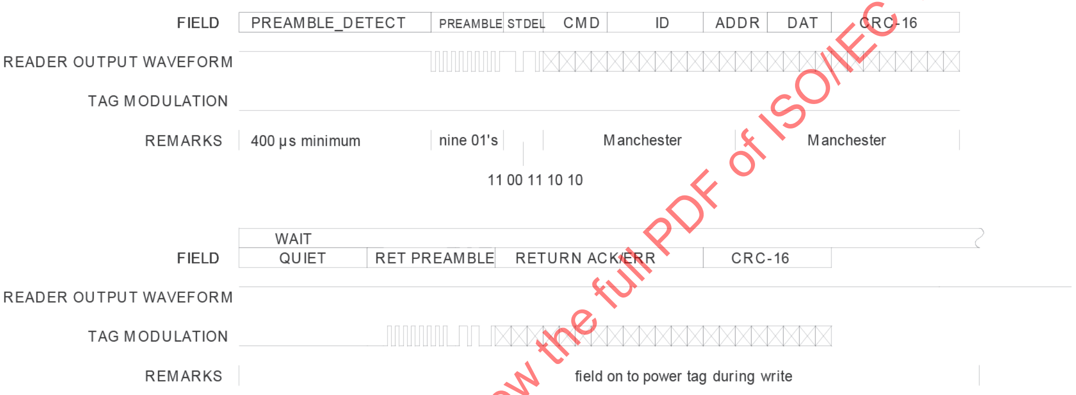


Figure 9 — Sample Command/Response Packets for WRITE (40 kbit/s on forward and return link)

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 initialise 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 Annex A, Clause A.2.

6.3.1.2 CRC

See clause 6.2.6 and Annex B.

6.3.1.3 FLAGS

The tag shall support a field of 8 flags. This field is called flags.

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 seconds in the temperature range 30 °C to 60 °C.

t_{DE_SB} shall be at least 4 seconds 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 organisation

The functional memory shall be organised 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.

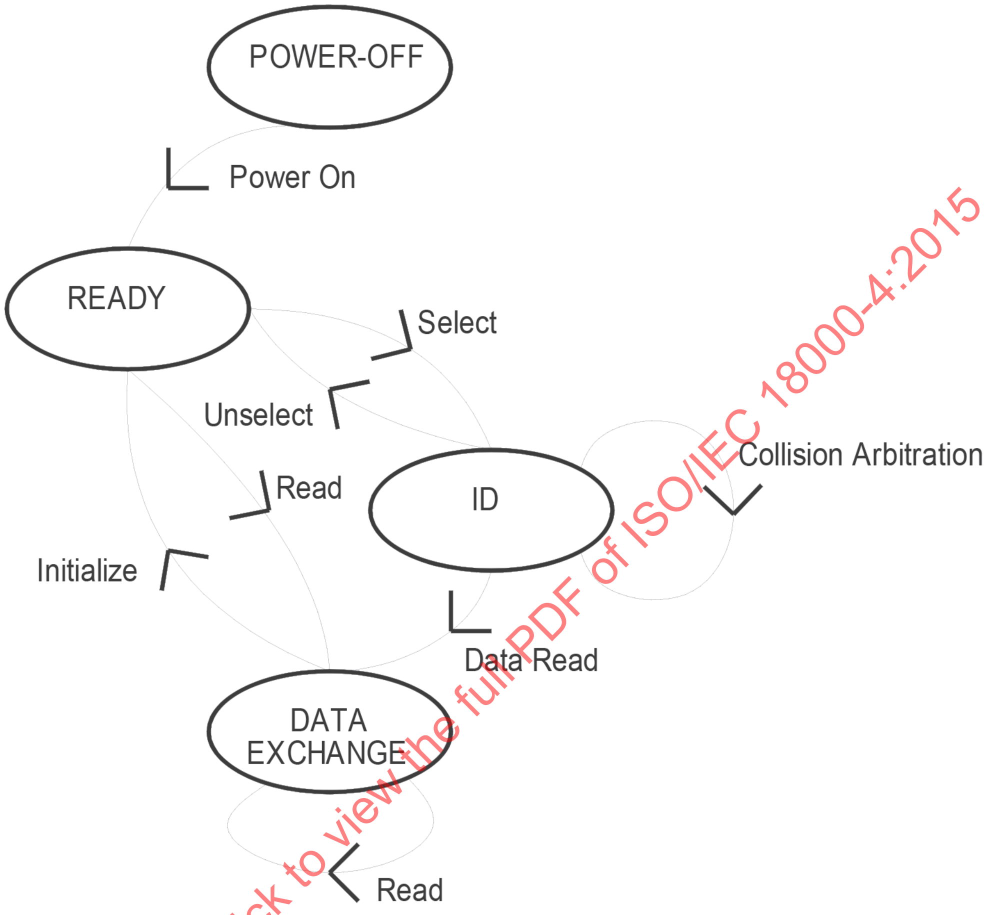
6.3.3 Block security status

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.

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 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
DATA_READ		X	X
READ_VERIFY	X	X	X
READ_VERIFY4BYTE	X	X	X
WRITE	X	X	X

Table 11 *(continued)*

COMMAND	States		
	READY	**ID**	**DATA EXCHANGE**
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
ID	GROUP_UNSELECT_NE	=	ID

Table 12 *(continued)*

Current State	Command	Condition	New state
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 notWRITE_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

Table 12 *(continued)*

Current State	Command	Condition	New state
DATA_EXCHANGE	READ		DATA_EXCHANGE
DATA_EXCHANGE	DATA_READ		DATA_EXCHANGE
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 are 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 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 zero shall keep COUNT at zero and shall send their UID again.

One of four possibilities now occurs:

d) If more than one tag transmits, the FAIL step 2 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 2. (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 5 or 6 repeats. If more than one tag transmits, step 2 repeats. (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 5 repeats. 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 2 repeats. (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 same IC version number shall behave the same.

6.3.6.1.1 Mandatory

The command codes range from ‘00’ to ‘0A’, ‘0C’, ‘15’,and ‘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 command 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 customised 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 the range from ‘E0’ to ‘FF’.

Proprietary Commands 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

Table 13 *(continued)*

Command code	Type	Command name	Parameters		
‘10’	Proprietary	IC manufacturer dependant			
‘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 dependant			
‘15’	Mandatory	RESEND	none	none	none
‘16’	Proprietary	IC manufacturer dependant			
‘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’-‘1F’	Mandatory	RFU			
‘20’-‘3F’	Mandatory	RFU			
‘40’ ‘9F’	Optional	RFU			
‘A0’ ‘DF’	Custom	IC Manufacturer dependent			
‘E0’ ‘FF’	Proprietary	IC Manufacturer dependent			

6.3.6.2.1 Command fields

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

Table 15 — Tag responses

Response code	Response name	Response size
‘00’	ACKNOWLEDGE	1 byte
	ACKNOWLEDGE_NOK	1byte
‘01’	ACKNOWLEDGE_OK	1byte
‘FE’	ERROR_NOK	1byte
‘FF’	ERROR	1byte
	ERROR_OK	1byte
Not applicable	WORD_DATA	8 bytes
Not applicable	BYTE_DATA	1byte
‘02’ ‘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 3 arguments (parameter and data):

ADDRESS

BYTE_MASK

WORD_DATA

and the tag shall make one of 4 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

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

and the argument of the command

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

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

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

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 2 arguments (parameter and data):

BYTE_MASK

BYTE_DATA

and the tag shall make of 2 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.

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
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 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 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 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.

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 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.

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 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.

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 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 stays 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.

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 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.

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 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.

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 to 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.

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 to 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.

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 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.

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 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.

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 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.

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 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.

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 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.

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 sent back it in the response.

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 an e.g. 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 sent back it in the response.

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 sent back it in the response.

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.

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 ID is not equal to UID or any other error the tag shall not send a reply. Further, the tag makes the byte of ADDESS lockable.

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

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 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, independently of the value of ID. Further, the tag makes the byte of ADDESS lockable.

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

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 ID is not equal to UID, WRITE_OK is not set, or any other error the tag shall not send a reply. Further, the tag makes the byte of ADDESS lockable.

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

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 ID is not equal to 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).

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, 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).

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, 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 tags 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 must be on a 4-byte page boundary.

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 that 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 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).

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 ID, 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 ADDESS 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).

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 unlocked, it shall program the data into the specified memory.

The tag shall not 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).

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 byte 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 any response.

Executing WRITE4BYTE a tags 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 must be on a 4-byte page boundary.

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 are 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 a 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 explicit 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 minimise 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.

Table 74 — Multiplication word

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

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

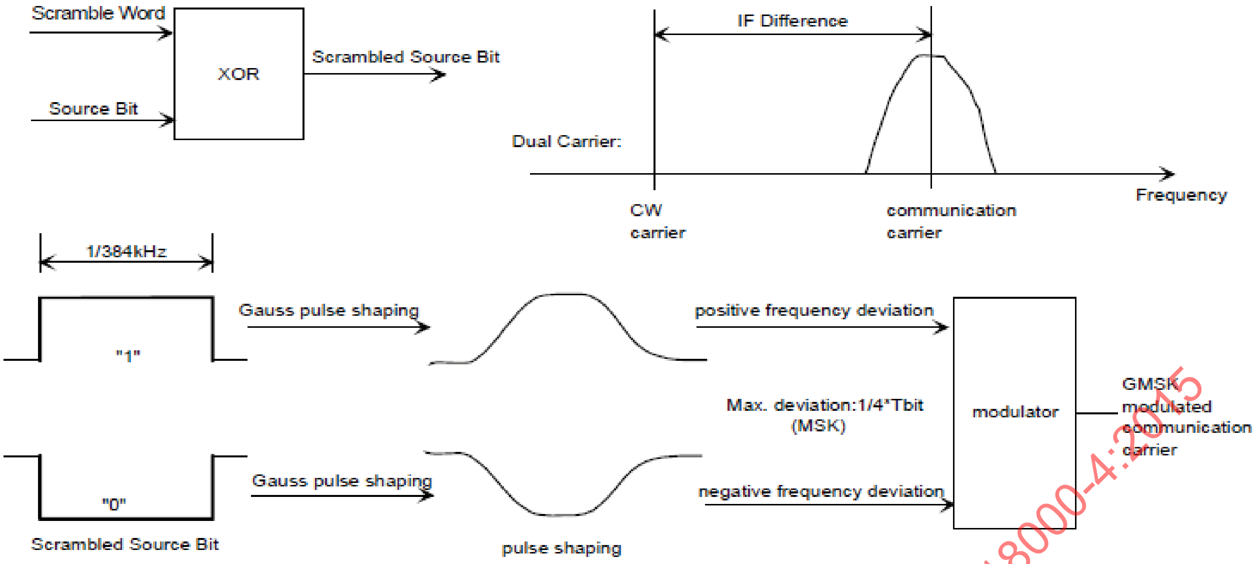


Figure 13 — Forward link modulation and coding

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 case of OOK the whole pre-processing like inverting, sub carrier modulation and differential encoding shall be applied.

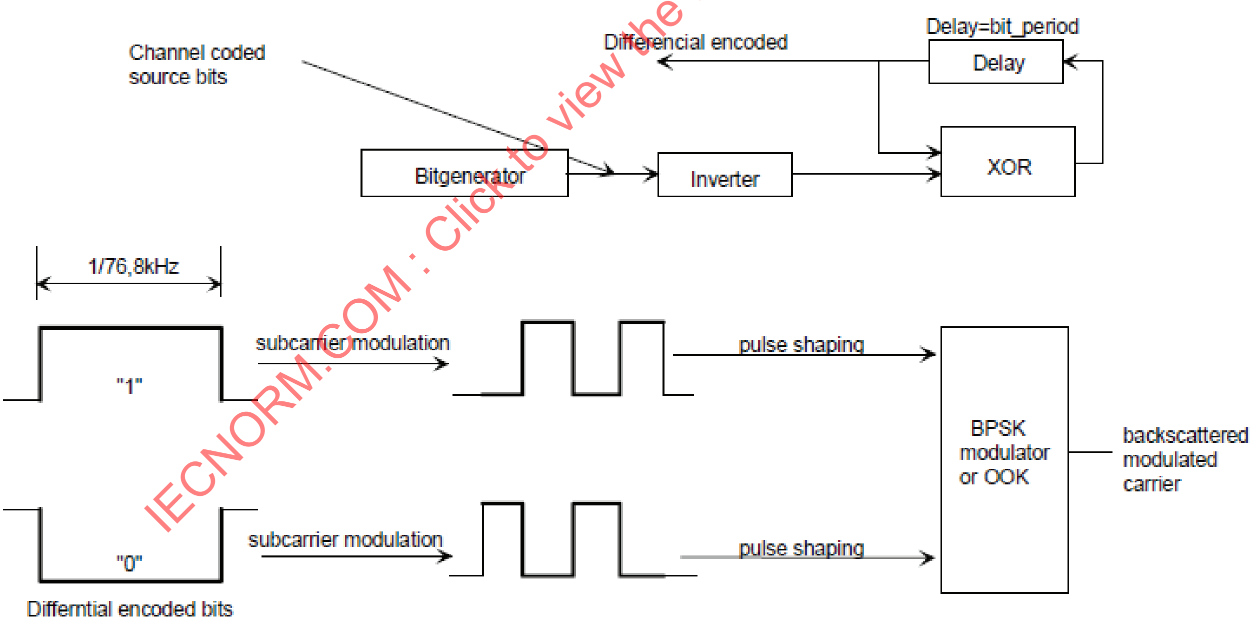


Figure 14 — Return link modulation and coding during notification

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.

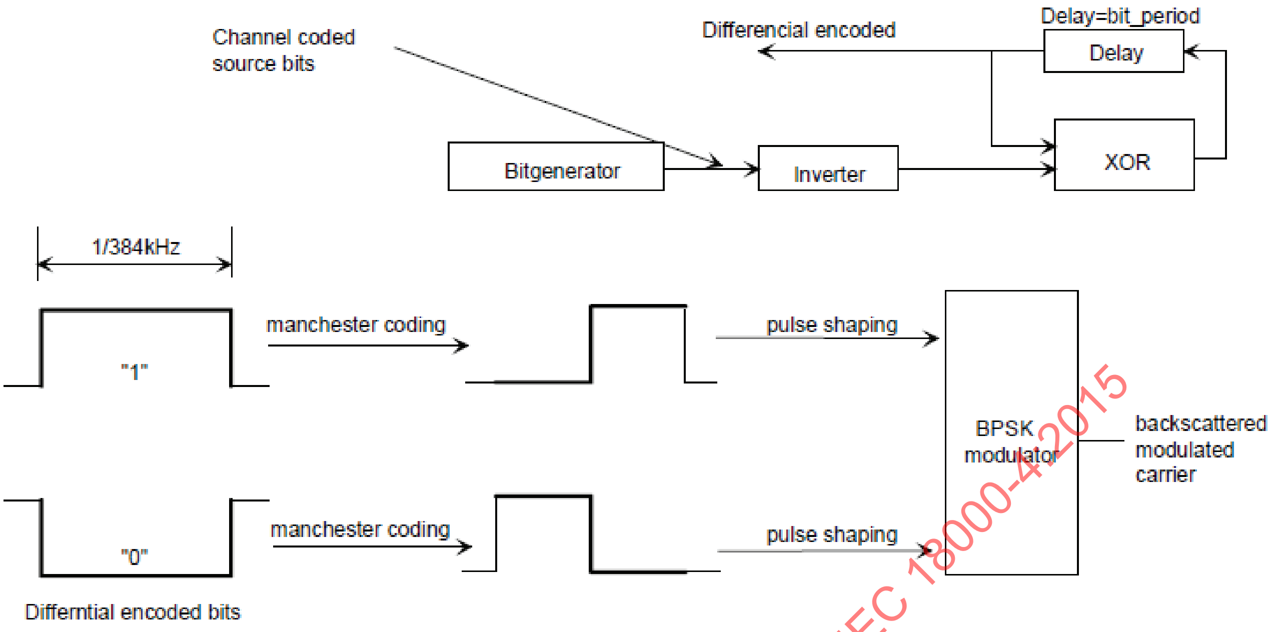


Figure 15 — Return link modulation and coding during communication

7.3 General system description

The system shall consist of an interrogator and at least one of 3 types of tags:

— A R/W-tag having read capability from and write capability to the tag.

— A R/O-tag behaving the same as a 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 synchronisation information and the tag data (including UserTagID, MfrTagID and MemoryID) to establish a communication. Depending on the content of the synchronisation 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 randomising the repetition time. Individual tags shall backscatter their sequence at an average repetition time but randomised (i.e. duty cycle of tag wake-up procedure and sleep time is random).

For the forward link of R/W tags two carriers are necessary, one modulated by GMSK (BT=0,5), 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 during 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, and may also be used to conform to regulatory requirements.

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 synchronising 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 synchronise to an arriving tag before the end of the notification mode. After completing the notification the interrogator turns back to the communication mode and continuous the existing transmission between the interrogator and tags. This allows the interrogator to organise all R/W-tags in the field in a time frame structure for subsequent service. All tag information from a 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 sub frames, each consisting of 14 time slots, are combined to a frame. The number of sub frames actually used is fixed at system installation, but may be changed on maintenance occasions. During communication between tag and interrogator, a sub frame 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 sub frames and each sub frame 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 sub frame is consequently (1/384 kHz)*200*14 = 7,29 ms.

The length of a frame is SW-configurable and ranges from one sub frame (approx. 7,3 ms) to 64 sub frames (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 sub frame is assigned permanently to a tag. It is not possible to assign a sub frame 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.

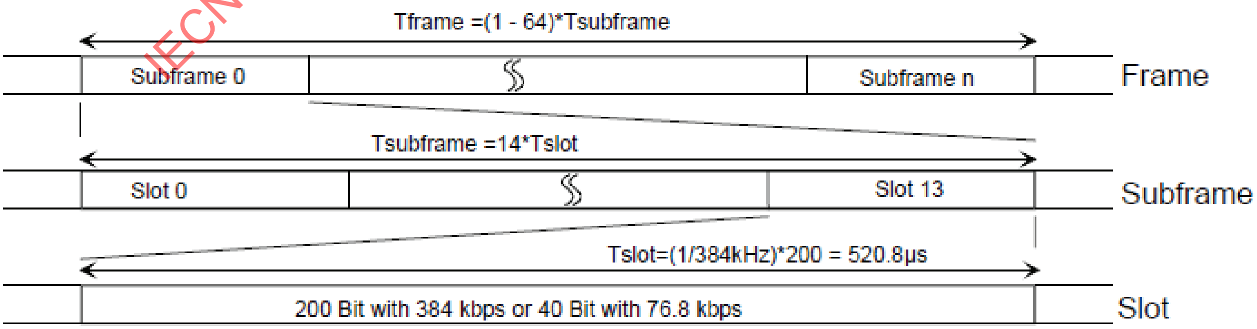

Tframe =(1 - 64)*Tsubframe
Subframe 0
Subframe n
Frame
Tsubframe =14*Tslot
Slot 0
Slot 13
Subframe
Tslot=(1/384kHz)*200 = 520.8µs
200 Bit with 384 kbps or 40 Bit with 76.8 kbps
Slot

Figure 16 — Frame structure

7.4.2 Logical channels

Definition: Logical channels are the assignments between sub frames 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 a R/W-tag (see Figure 17), or all the information shall be read out from the tag if it is a 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 are 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 sub frame assigned to and reserved for the communication channel.

Notification shall be cancelled if:

- The retrieved TagID is black listed (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 sub frames 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 tag reads the first command and shall be terminated when 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)

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

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 sub frame.

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

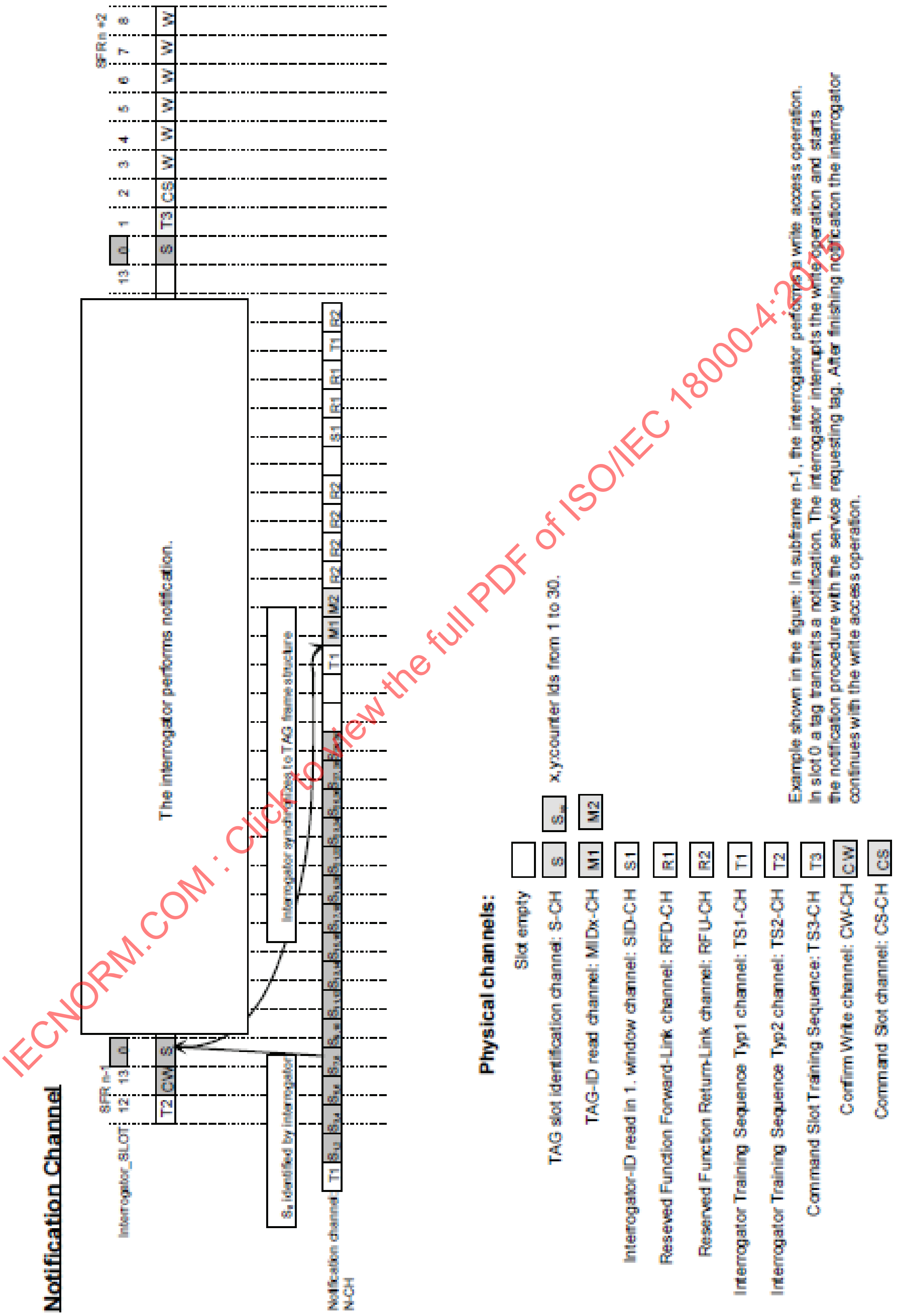
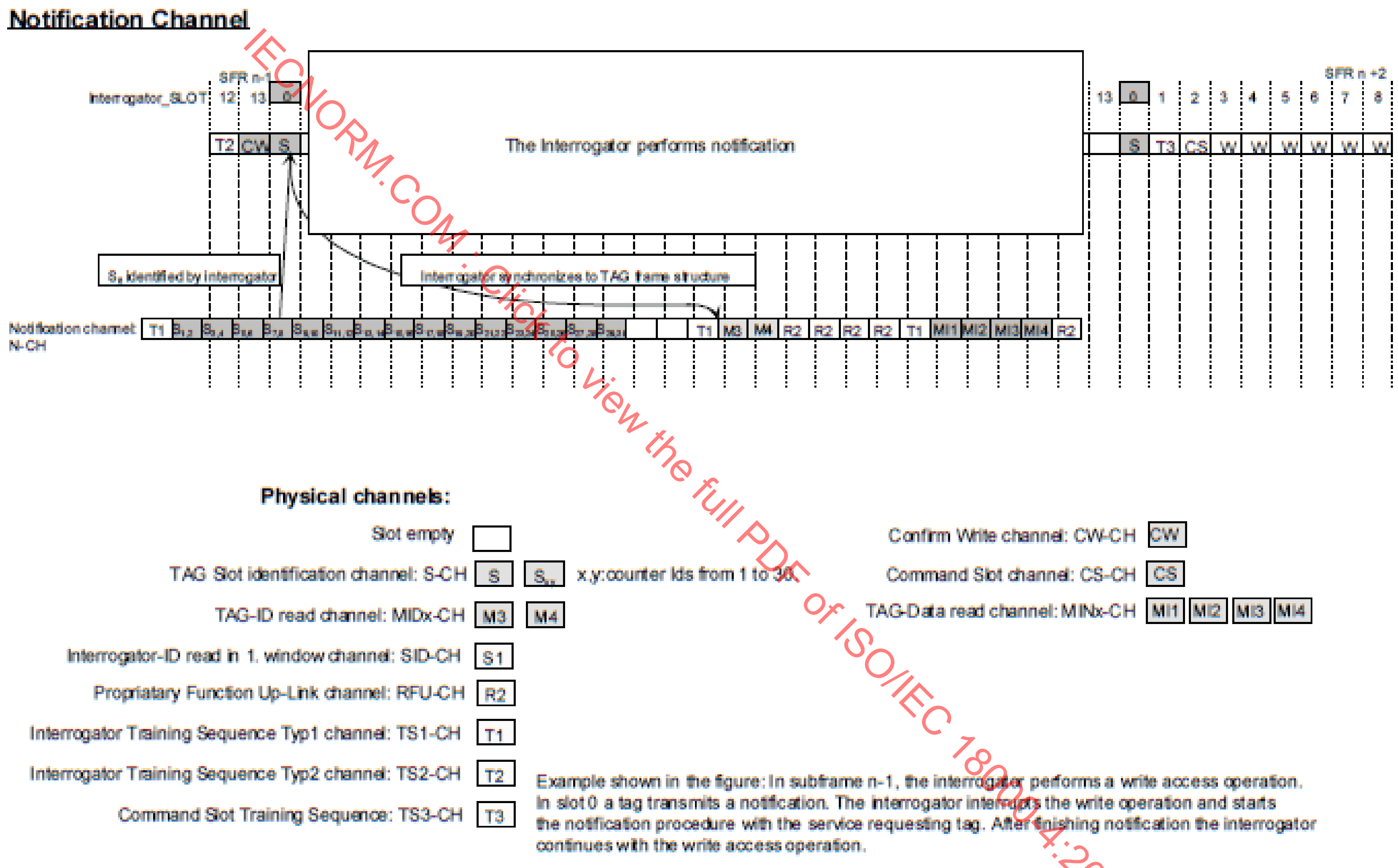

Notification Channel
Interrogator_SLOT
SFR n-1
12
13
0
SFR n+2
13
0
1
2
3
4
5
6
7
8
T2
CW
S
T3
CS
W
The interrogator performs notification.
S_x identified by interrogator
Interrogator synchronizes to TAG frame structure
Notification channel
N-CH
T1
M1
M2
R2
S1
R1
Physical channels:
Slot empty
TAG slot identification channel: S-CH
x.y counter Ids from 1 to 30.
TAG-ID read channel: MIDx-CH
Interrogator-ID read in 1. window channel: SID-CH
Reserved Function Forward-Link channel: RFD-CH
Reserved Function Return-Link channel: RFU-CH
Interrogator Training Sequence Typ1 channel: TS1-CH
Interrogator Training Sequence Typ2 channel: TS2-CH
Command Slot Training Sequence: TS3-CH
Confirm Write channel: CW-CH
Command Slot channel: CS-CH
Example shown in the figure: In subframe n-1, the interrogator performs a write access operation. In slot 0 a tag transmits a notification. The interrogator interrupts the write operation and starts the notification procedure with the service requesting tag. After finishing notification the interrogator continues with the write access operation.

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

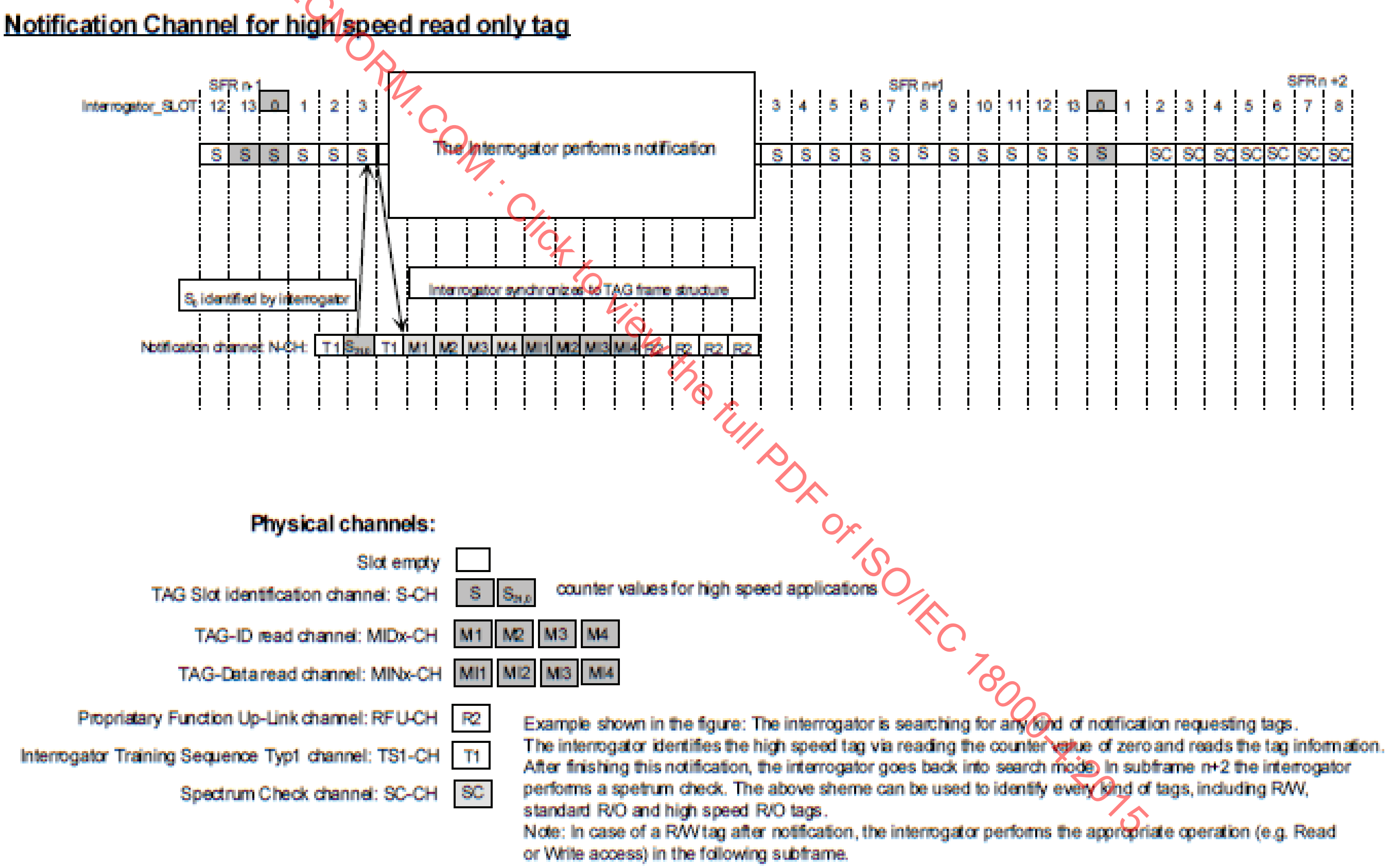
7.4.2.6 Frame structure for the notification channel in case of an R/O-tag


Notification Channel
SFR n-1
Interrogator_SLOT
12
13
0
T2
CW
S
The Interrogator performs notification
SFR n +2
13
0
1
2
3
4
5
6
7
8
S
T3
CS
W
W
W
W
W
W
S_x identified by interrogator
Interrogator synchronizes to TAG frame structure
Notification channel
N-CH
T1
$S_{1,2}$
$S_{3,4}$
$S_{5,6}$
$S_{7,8}$
$S_{9,10}$
$S_{11,12}$
$S_{13,14}$
$S_{15,16}$
$S_{17,18}$
$S_{19,20}$
$S_{21,22}$
$S_{23,24}$
$S_{25,26}$
$S_{27,28}$
$S_{29,30}$
T1
M3
M4
R2
R2
R2
R2
T1
MI1
MI2
MI3
MI4
R2
Physical channels:
Slot empty
TAG Slot identification channel: S-CH
S
$S_{x,y}$
x,y:counter Ids from 1 to 30
TAG-ID read channel: MIDx-CH
M3
M4
Interrogator-ID read in 1. window channel: SID-CH
S1
Propriatary Function Up-Link channel: RFU-CH
R2
Interrogator Training Sequence Typ1 channel: TS1-CH
T1
Interrogator Training Sequence Typ2 channel: TS2-CH
T2
Command Slot Training Sequence: TS3-CH
T3
Confirm Write channel: CW-CH
CW
Command Slot channel: CS-CH
CS
TAG-Data read channel: MINx-CH
MI1
MI2
MI3
MI4

Example shown in the figure: In subframe n-1, the interrogator performs a write access operation. In slot 0 a tag transmits a notification. The interrogator interrupts the write operation and starts the notification procedure with the service requesting tag. After finishing notification the interrogator continues with the write access operation.

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


Notification Channel for high speed read only tag
SFR n+1
SFR n+1
SFR n+2
Interrogator_SLOT
12 13 0 1 2 3 3 4 5 6 7 8 9 10 11 12 13 0 1 2 3 4 5 6 7 8
S S S S S S S S S S S S S S S S S S S
SC SC SC SC SC SC SC
The Interrogator performs notification
Interrogator synchronizes to TAG frame structure
S_t identified by interrogator
Notification channel N-CH:
T1 $S_{H,t}$ T1 M1 M2 M3 M4 MI1 MI2 MI3 MI4 R2 R2 R2 R2
Physical channels:
Slot empty
TAG Slot identification channel: S-CH
S $S_{H,0}$
counter values for high speed applications
TAG-ID read channel: MIDx-CH
M1 M2 M3 M4
TAG-Data read channel: MINx-CH
MI1 MI2 MI3 MI4
Propriatary Function Up-Link channel: RFU-CH
R2
Interrogator Training Sequence Typ1 channel: TS1-CH
T1
Spectrum Check channel: SC-CH
SC

Example shown in the figure: The interrogator is searching for any kind of notification requesting tags.
The interrogator identifies the high speed tag via reading the counter value of zero and reads the tag information.
After finishing this notification, the interrogator goes back into search mode. In subframe n+2 the interrogator performs a spetrum check. The above sheme can be used to identify every kind of tags, including R/W, standard R/O and high speed R/O tags.
Note: In case of a R/W tag after notification, the interrogator performs the appropriate operation (e.g. Read or Write access) in the following subframe.

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

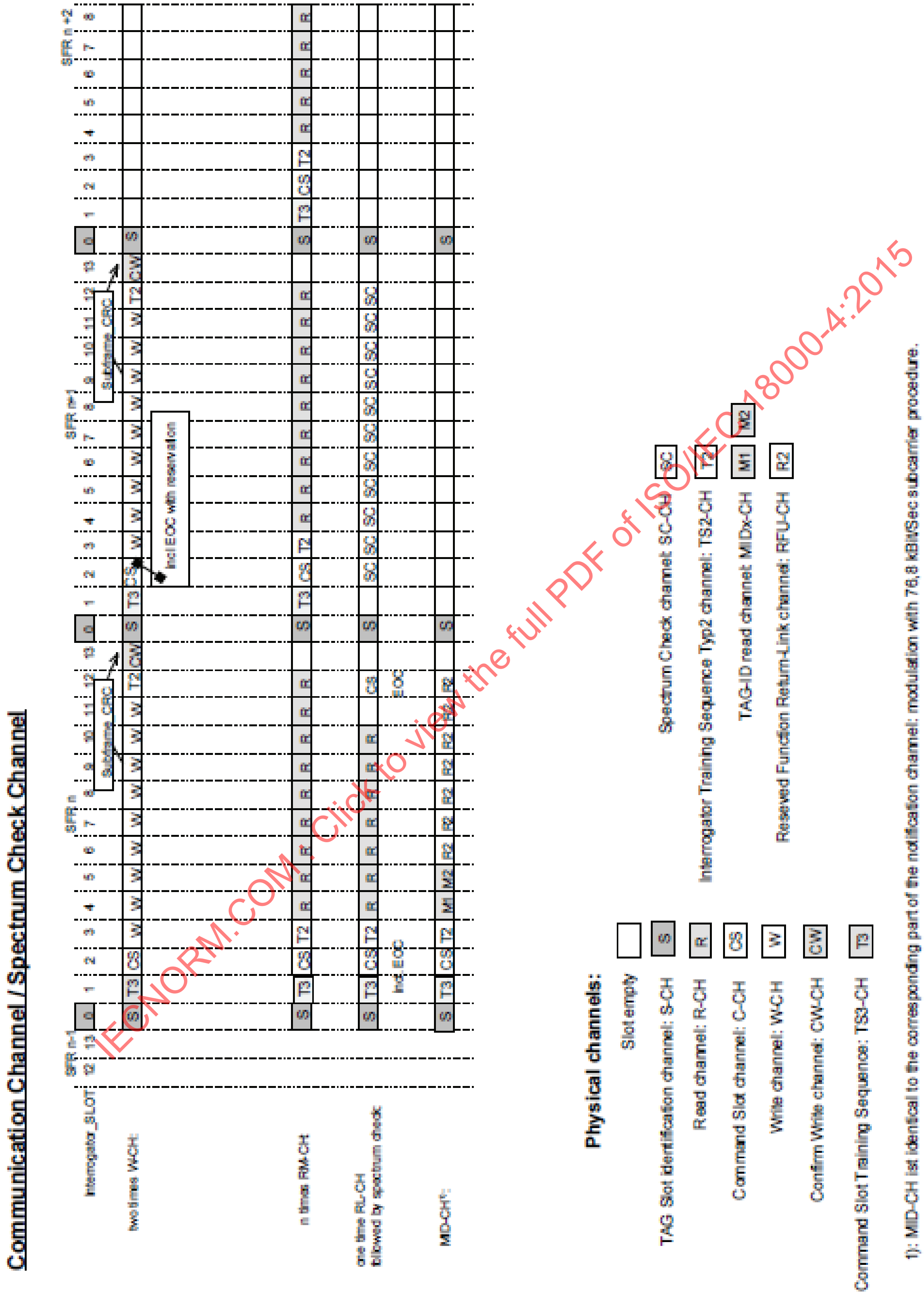

Communication Channel / Spectrum Check Channel
Interrogator_SLOT
SFR n-1
SFR n
SFR n+1
SFR n +2
0 1 2 3 4 5 6 7 8 9 10 11 12 13
two times W-CH:
n times RM-CH
one time RL-CH followed by spectrum check
MID-CH1):
Subframe CRC
incl EOC with reservation
incl EOC
EOC
S T3 CS T2 W R M1 M2 R2 CW SC
Physical channels:
Slot empty
TAG Slot identification channel: S-CH
Read channel: R-CH
Command Slot channel: C-CH
Write channel: W-CH
Confirm Write channel: CW-CH
Command Slot Training Sequence: TS3-CH
Spectrum Check channel SC-CH
Interrogator Training Sequence Typ2 channel: TS2-CH
TAG-ID read channel MIDx-CH
Reserved Function Return-Link channel: RFU-CH
1): MID-CH ist identical to the corresponding part of the notification channel: modulation with 76,8 kBit/Sec subcarrier procedure.

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 synchronisation 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 for 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 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 sub frame 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 sub frame 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 sub frame 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 sub frame was free of errors or not.
		Interrogator Training Sequence Type 2 channel: TS2-CH (return link)	This channel is used only for ease the implementation of the hardware.
		Command Slot Training Sequence: TS3-CH (forward link)	This channel is used only for 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 synchronisation 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 synchronisation (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 synchronisation 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.

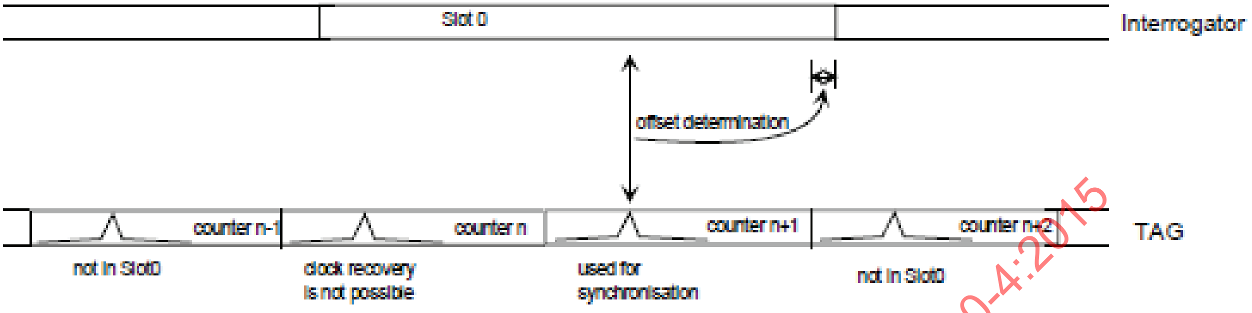


Figure 21 — S-CH position with regard to slot0

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

Table 78 — Sub frame 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 — Sub frame 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.

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.

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 the 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

Position of half slot	Position of slot0																																								Number of correlation bits
	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	22
2	0	1	0	1	0	0	1	1	0	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	21
3	1	0	1	0	0	1	1	0	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	20
4	0	1	0	0	1	1	0	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	20
5	1	0	0	1	1	0	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	20
6	0	0	1	1	0	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	20
7	0	1	1	0	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	20
8	1	1	0	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	20
9	1	0	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	20
10	0	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	1	20
11	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	1	0	20
12	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	1	0	0	20
13	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	1	0	0	0	20
14	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	1	0	0	0	0	20
15	#N	#N				P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	21
16	#N	#N			P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+2	22
17	#N	#N		P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+2	#N+2	22
18	#N	#N	P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+2		#N+2	22
19	#N	P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+2			#N+2	22
20	P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+1					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+2				#N+2	22
1	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N					P	T	0	1	0	1	0	0	1	1	0	0	0	0	0	#N+2					P	22

Dark Grey: Clock recovery
Boxed white: Bits with uncertainty
Light Grey: Correlator bits
Dashed boxed: Position of the correlation peak
P: Parity
#N: Time counters
T: Tail

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

Sub frame assignment: not relevant, since MIDx-CH is a part of the notification channel. The position in the sub frame 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.

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 zeros 0.....0h

— Divide in GF(2) the polynomial

$$ID_{31}x^{53}+ID_{30}x^{52}+\ldots+ID_{0}x^{22}$$

by the generator polynomial

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

obtain as remainder the polynomial

$$CRC_{21}x^{21}+\ldots+CRC_{0}x^{0}$$

— 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

$$ID_{31}x^{53}+\ldots+ID_{0}x^{22}+CRC_{21}x^{21}+\ldots+CRC_{0}x^{0}$$

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 64bits 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

Sub frame assignment: not relevant, since MINx-CH is a part of the notification channel. The position in the sub frame 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.

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: This 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 zeros 0.....0h

— Divide in GF(2) the polynomial

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

by the generator polynomial

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

obtain as remainder the polynomial

$$CRC_{21}x^{21}+\ldots+CRC_0x^0$$

— Attach the CRC bits (CRC_{21}, ..., CRC_0) to the end of the databits (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

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

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 384kbit/s

Sub frame assignment: not relevant, since SID-CH is a part of the notification channel. The position in the sub frame 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 synchronisation, a sequence (TSC1) with a length of 16 is used. The default correlator threshold for word synchronisation is 13 (the value must exceed 13: two bit errors maximum).

Table 87 — Slot assignment for SID-CH part1

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 part2

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 part3

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 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 zeros 0…..0h
- Divide in GF(2) the polynomial

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

  by the generator polynomial

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

  obtain as remainder the polynomial

  $$CRC_{14}x^{14}+\ldots+CRC_0x^{0}$$

- Attach the CRC bits (CRC_{14}, …, CRC_0) to the end of the databits (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

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

  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.

Sub frame 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 synchronisation and for error protection.

Data transmission: forward link with 384 kbit/s.

Table 90 — Sub frame 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 — Sub frame 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 synchronisation, a sequence (TSC1) with a length of 26 is used. The default correlator threshold for word synchronisation is 24 (The value must 24: two bit errors maximum). The remaining 162 bits are coded net data bits.

Table 92 — Slot assignment for CS-CH part1

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 part2

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 part3

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 part4

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 clauses 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 sub frames 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 zeros 0.....0h
- Divide in GF(2) the polynomial

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

  by the generator polynomial

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

  obtain as remainder the polynomial

  $$CRC_{43}x^{43}+\ldots+CRC_0x^0$$

- Attach the CRC bits (CRC_{43}, …, CRC_0) to the end of the databits (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

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

  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 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 synchronisation and for error protection.

Data transmission: return link with 384 kbit/s.

Sub frame assignment:

Table 96 — Sub frame 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 — Sub frame 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	CRC over previous 48 bits	Net data bit	CRC over bits B96 ... B49	Tail bit
(48 bits)	(44 bits)	(48 bits)	(44 bits)	
B188...B141	B140...B97	B96...B49	B48...B5	B4...B0
D_{47}, ... D_0	CRC_{43},...,CRC_0	D_{47}, ... D_0	CRC_{43},...,CRC_0	

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 zeros 0.....0h

— Divide in GF(2) the polynomial

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

by the generator polynomial

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

obtain as remainder the polynomial

$$CRC_{43}x^{43} + \ldots + CRC_0x^0$$

— Attach the CRC bits (CRC_{43}, ..., CRC_0) to the end of the databits (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

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

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 synchronisation and for error protection.

Data transmission: forward link with 384 kbit/s.

Table 100 — Sub frame 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 synchronisation 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	CRC over previous 128 bits
(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 zeros 0.....0h
- Divide in GF(2) the polynomial

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

  by the generator polynomial

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

  obtain as remainder the polynomial

  $$CRC_{43}x^{43}+\ldots+CRC_0x^0$$

- Attach the CRC bits (CRC_{43}, ..., CRC_0) to the end of the databits (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

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

  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

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

Function: This channel signals whether the transmission of data from interrogator to tag in a given sub frame was free of errors or not. For a transmission to be error-free, all the slots must 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.

Table 104 — Sub frame 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 interrogator with 2*26 bits consecutively (sequence TSC1). If the received CRC was not o.k., a word containing nothing but 0´s shall be generated with a length of 2*26. 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 must 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 for 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.

Sub frame 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 shall be sent.

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

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

Table 108 *(continued)*

												•	

Table 109 — Sub frame 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 for ease the implementation of the hardware.

Data transmission: forward link with 384kbit/s. Last 30 bits are an alternating series of 2 zeros and 2 ones started with 2 ones.

Sub frame assignment: always transmitted only before CS-CH if CS-CH transmitted in Slot2.

Table 111 — Sub frame 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.

Sub frame assignment: only if there is neither communication nor notification.

Table 113 — Sub frame 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 Synchronisation and CRC patterns

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

Channel type		Clock	Word	Description
N-CH	S-CH (R/W-tag)	0101	001100000	Word incl. clock
	S-CH * (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	
	CS-CH	010101010101	00100101110000100010010111	
	TS3-CH	Not relevant	Not relevant	
SC-CH		Not relevant	Not relevant	
* The clock recovery can work also with the capital letters: 0101: the same sequence as for a R/W-tag				

Table 115 — CRC parameterisation 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 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 therefore 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 part of ISO/IEC 18000. 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 sub frame 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 sub frame 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 sub frame 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

Table 116 — Command codes in slot 2

Name	Type	Command code				EOC	Function
		B161 ... B158				B157	
Wait	Mandatory	0	0	0	0	x	Sub frame is not filled with data. An EOC in slot12 is not to be expected. This is a 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	Sub frame 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	Sub frame filled with read data. The communication in this sub frame cannot be terminated.
Write	Mandatory	1	1	0	0	0	Sub frame filled with write data. The communication in this sub frame shall not be terminated. A feedback on the validity of the data received by tag in this sub frame shall be sent to interrogator on the CW-CH channel.
						1	The communication in this sub frame shall be terminated when the CRCs signal valid data for all the slots. A feedback on the validity of the data received in this sub frame shall be sent to interrogator on the CW-CH channel. Once the data has been written to RAM, the tag shall return to the sleep mode.

Table 116 *(continued)*

Name	Type	Command code				EOC	Function
		B161 … B158				B157	
Init	Optional	1	1	1	1	x	A feedback on the validity of the data received in this sub frame (INIT byte) shall be sent to interrogator on the CW-CH channel. The communication cannot be terminated during initialisation. During initialisation the tag must be polled in each respective sub frame to find out whether or not the initialisation has been terminated
Reserved for future use	Optional	0	1	0	1	x	
IC Mfg dependent	Custom	0	0	1	0	x	
		0	1	0	0		
		1	0	1	1		
		1	1	0	1		
		1	1	1	0		
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 slot 12

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		

NOTE 1 In the case of command decoding error, the tag shall try ten 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.

NOTE 2 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 vendors 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 power integrated into the device itself (battery, power scavenging capability), and in general have an expected life span of between three and five 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 must 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 and regulatory requirements of the appropriate government regulations.

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 complaint 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 the ISO/IEC/IEEE 8802-15-4 standard.

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

This standard utilizes 16 channels (channel 11 to channel 26) over the 2,405 – 2,483 GHz spectrum, supporting a 250000 bits per second data rate using QPSK modulation. (see ISO/IEC/IEEE 8802-15-4, section 10)

Channel conflict resolution shall be attained utilizing Carrier Sense Multiple Access (CSMA) with a hold off up to 10 ms. (See ISO/IEC/IEEE 8802-15-4, section 5.1.1.4)

This protocol 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 summarises the network topology:

a) Ocean container devices described in this specification 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

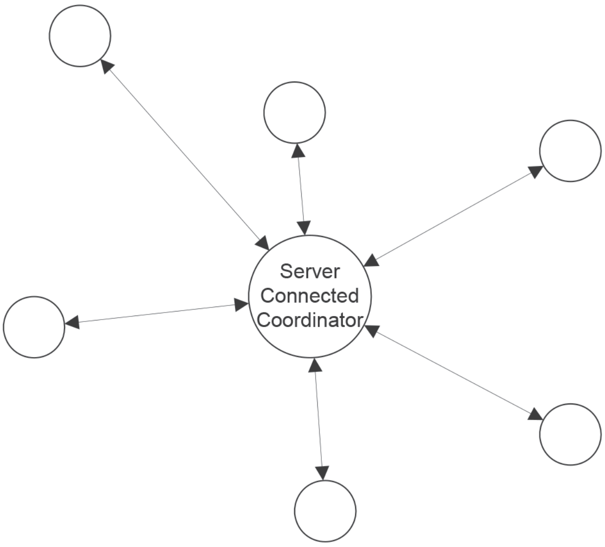


Figure 22 — Star Topology

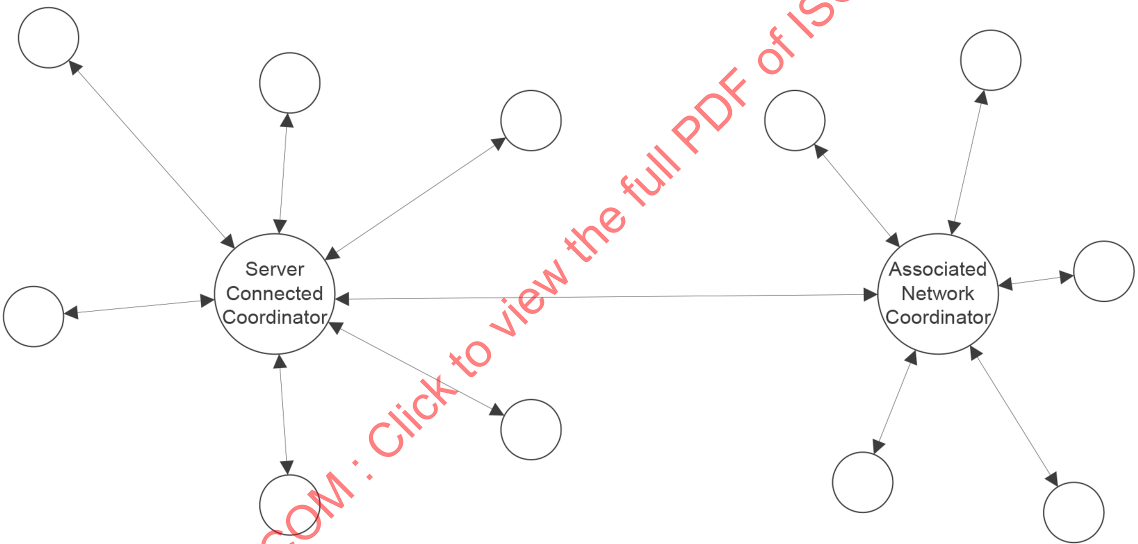


Figure 23 — Trunk Topology

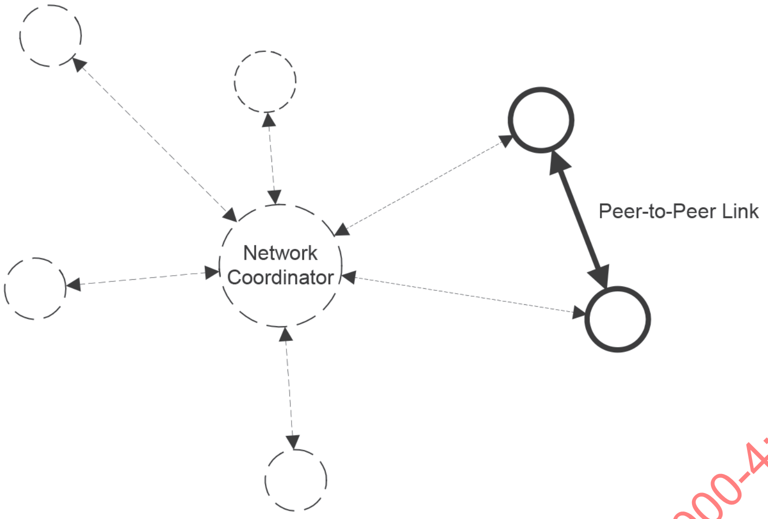


Figure 24 — Peer-to-Peer Topology

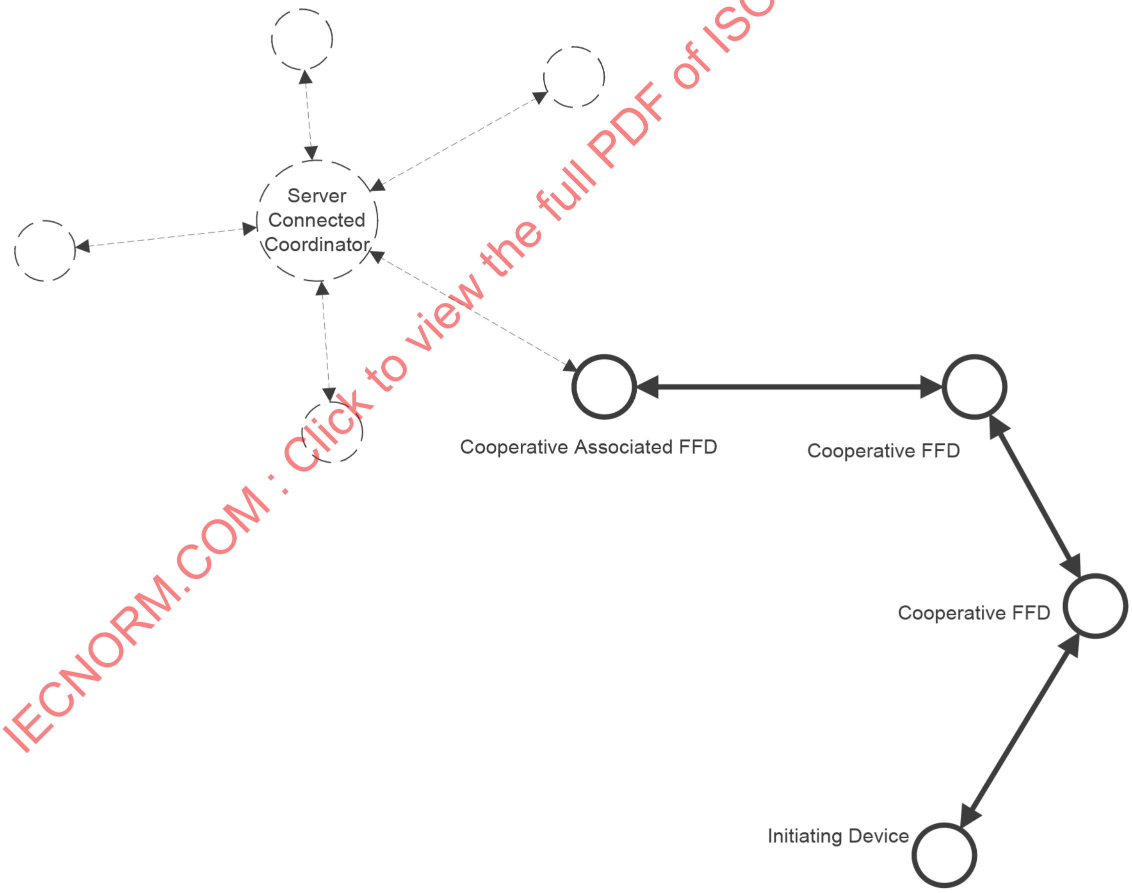


Figure 25 — Mesh Topology

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 one second.

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 a SCC. The network coordinator shall emit a series of periodic NDB packets to initiate network connections. When a device receives a 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 behavior 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 standard defines 4 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 a 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 (sub-clause 8.10.4).

8.6.1 Trunk Coordinator Requirements

Before a trunk coordinator may operate in a star network, it shall first associate with a 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 a NSM to its SCC. This NSM shall indicate that it wishes to associate as a trunk coordinator. Any trunk coordinator associated with a 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 is 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 sub-clause 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.

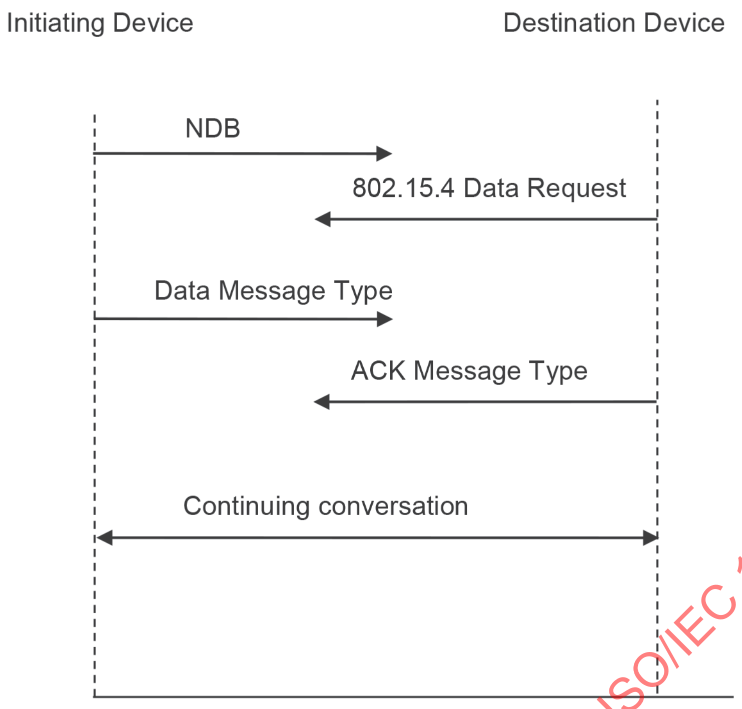


Figure 26 — Network Access for Peer to Peer Topology

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 ISO/IEC/IEEE 8802-15-4.

Cooperative FFDs shall maintain an active receiver for a minimum of five (5) seconds 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 (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 five (5) seconds 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 is 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.

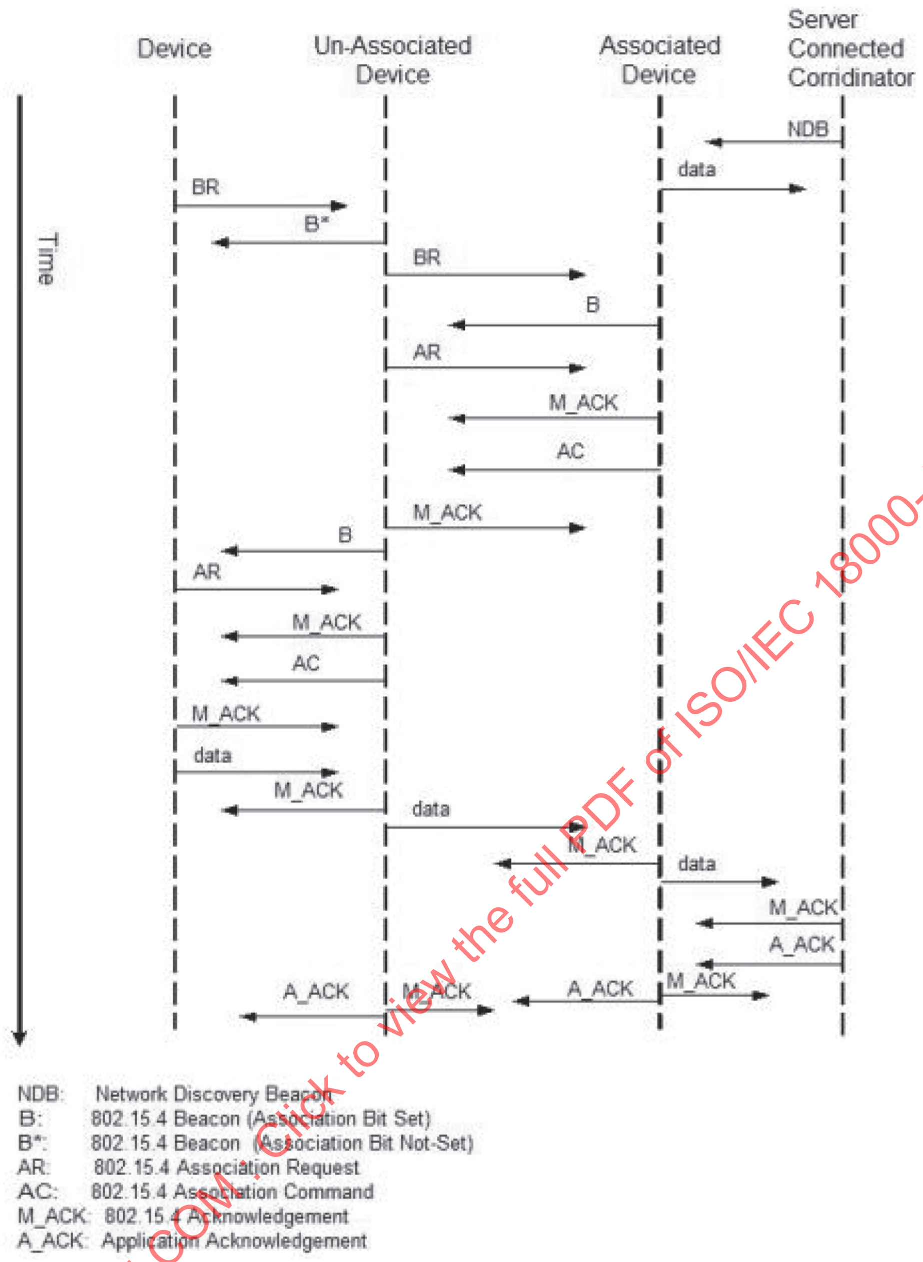
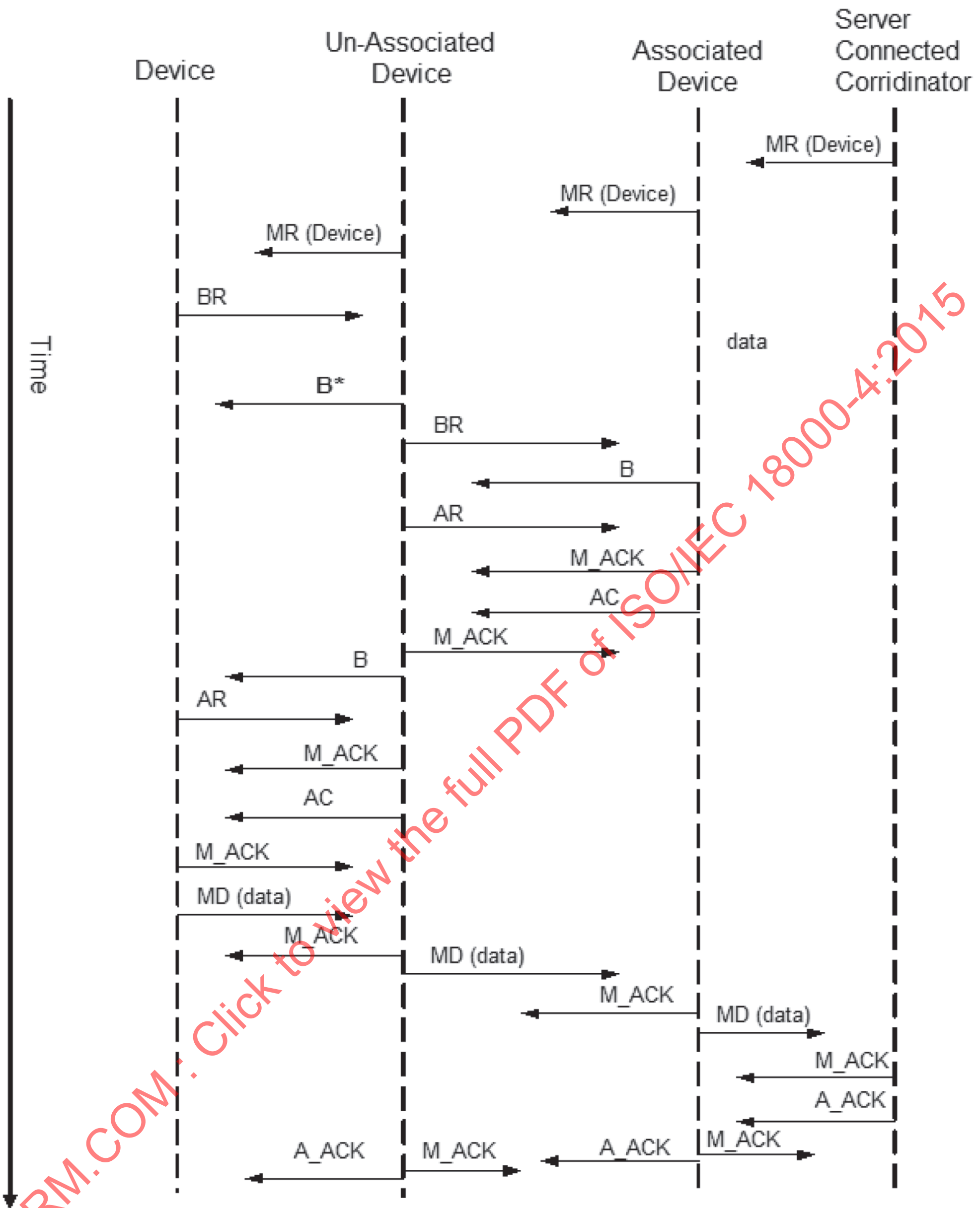

Device
Un-Associated Device
Associated Device
Server Connected Corridinator
Time
NDB
data
BR
B*
BR
B
AR
M_ACK
AC
M_ACK
B
AR
M_ACK
AC
M_ACK
data
M_ACK
data
M_ACK
data
M_ACK
A_ACK
A_ACK
M_ACK
A_ACK
M_ACK
NDB: Network Discovery Beacon
B: 802.15.4 Beacon (Association Bit Set)
B*: 802.15.4 Beacon (Association 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 sub-clause. 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).



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 specification defines message types that ensure interoperability between a diverse set of devices and manufacturers. Within message type categories, devices may send user data that is routed using this specification. Although messages are not limited by the use of this specification, these messages must 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 specification shall meet the message header specified in ISO/IEC/IEEE 8802-15-4. This clause and its sub-clauses 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 sub-clauses of clause 8.9 for details.

This specification defines message types that ensure interoperability between a diverse set of devices and manufacturers. Within message type categories, devices may send proprietary data that is routed using this specification. Although proprietary messages are not limited by the use of this specification, proprietary messages must 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 specification shall meet the message header specified in ISO/IEC/IEEE 8802-15-4. This clause and its sub-clauses 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 8.9.8 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 Type Octet Format. The data message is variable depending on the message type. See Figure 29 Description of Data Packet showing Payload Section and the sub-clauses of clause 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 vendor 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.

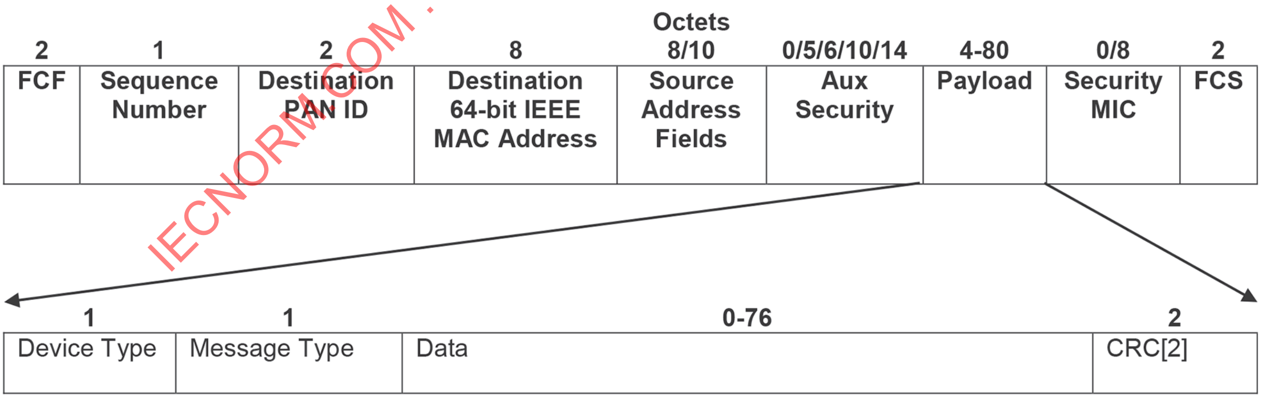


Figure 29 — General Description of Data Packet showing Payload Section

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 one second 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

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

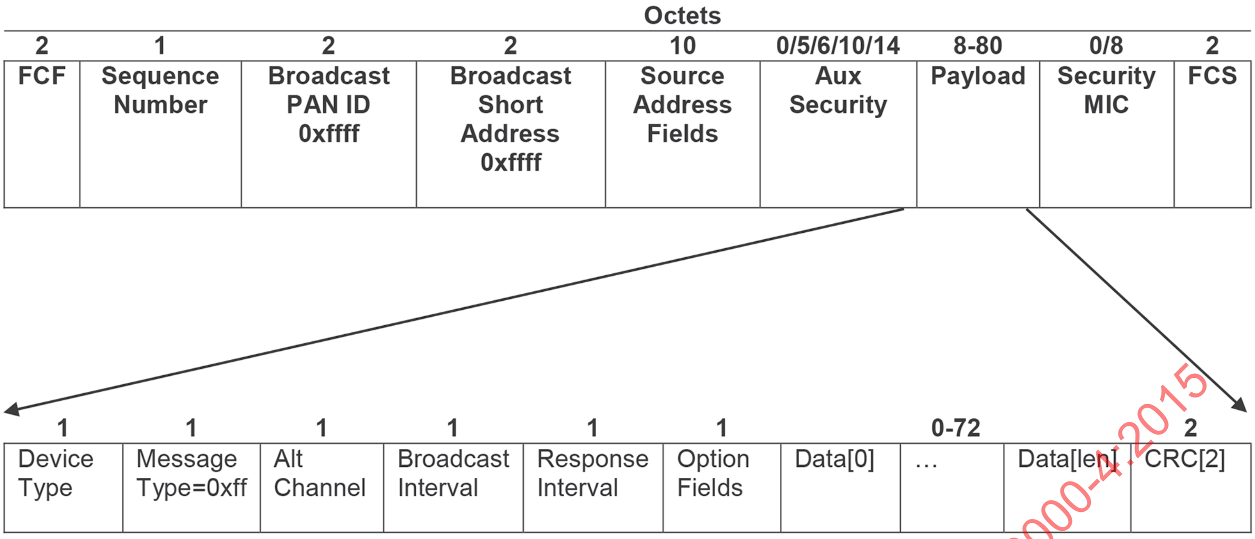
NDB Timing

— NDBs are emitted periodically to announce network availability

— NDBs are emitted at a high rate for a 1 (one) second period to allow remote devices to support deep duty cycles

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

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

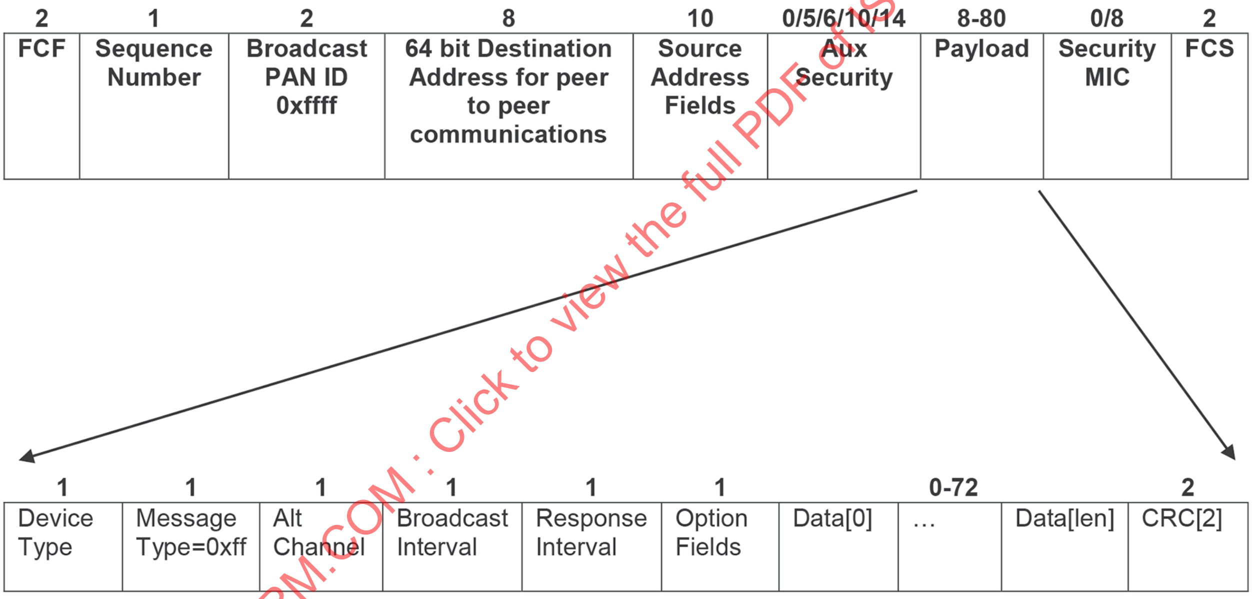


Octets

2	1	2	2	10	0/5/6/10/14	8-80	0/8	2
FCF	Sequence Number	Broadcast PAN ID 0xffff	Broadcast Short Address 0xffff	Source Address Fields	Aux Security	Payload	Security MIC	FCS

1	1	1	1	1	1	0-72			2
Device Type	Message Type=0xff	Alt Channel	Broadcast Interval	Response Interval	Option Fields	Data[0]	…	Data[len]	CRC[2]

Figure 30 — Broadcast NDB Data Frame Description



2	1	2	8	10	0/5/6/10/14	8-80	0/8	2
FCF	Sequence Number	Broadcast PAN ID 0xffff	64 bit Destination Address for peer to peer communications	Source Address Fields	Aux Security	Payload	Security MIC	FCS

1	1	1	1	1	1	0-72			2
Device Type	Message Type=0xff	Alt Channel	Broadcast Interval	Response Interval	Option Fields	Data[0]	…	Data[len]	CRC[2]

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 second burst of NDB frames)
Response Interval	Integer value of NDB burst (see Figure 38) that a tag receives before it must communicate status to the network coordinator
Option Fields	See Table 122
Data[0]	Option payload data length including the 2 byte CRC
Data[1]-[71]	Option payload data content

Table 120 *(continued)*

Byte Name	Byte Definition
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 sub-clauses 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 0-4

- Integer value (lower 5 bits) representing alternate ISO/IEC/IEEE 8802-15-4 channel
- Valid channel values: 11-26

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

- 1-254: 1-254 seconds between NDB frame transmissions
- 255: No NDB retransmissions

The typical values are:

- Long range network coordinator (800-1000 meters) : 60 = 60 seconds
- Short range network coordinator (50-100 meters): 15 = 15 seconds

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
- 1-255: respond every (1-255) NDB bursts

8.9.1.4 Option Fields and Data

The option field description is shown in Table 122. The subsequent sub-clauses 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 1-12
- DD: day 1-31
- YY: year 0-99
- Wd: Weekday 0-6
- HH: 0-23
- MM:0-59
- SS: 0-59

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 contains 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.

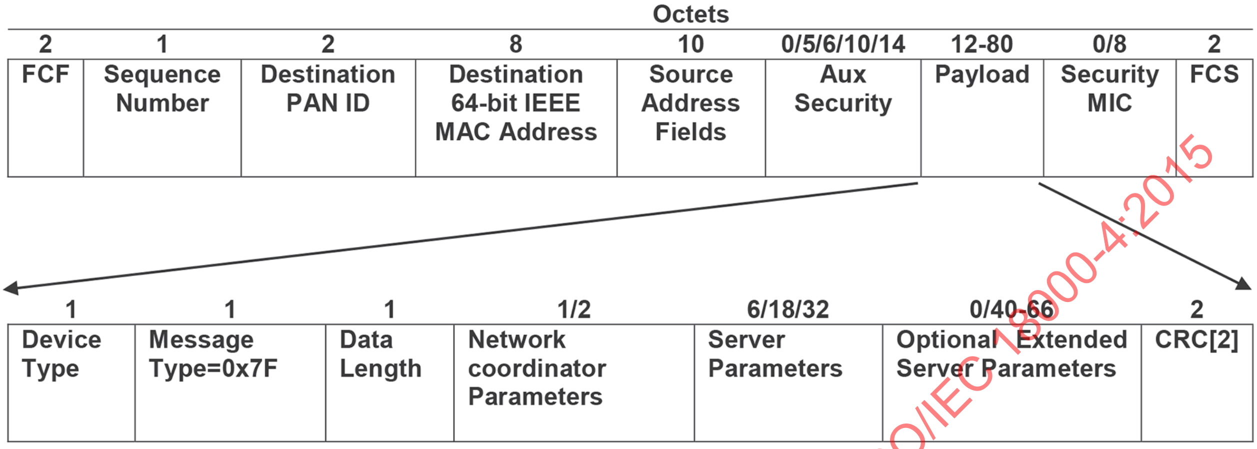
8.9.2 Network Status Message (NSM)

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.



Octets

2	1	2	8	10	0/5/6/10/14	12-80	0/8	2
FCF	Sequence Number	Destination PAN ID	Destination 64-bit IEEE MAC Address	Source Address Fields	Aux Security	Payload	Security MIC	FCS

1	1	1	1/2	6/18/32	0/40-66	2
Device Type	Message Type=0x7F	Data Length	Network coordinator Parameters	Server Parameters	Optional Extended Server Parameters	CRC[2]

Figure 32 — Network Status Message Packet Description

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

Table 123 — Network Status Message Payload Description

Name	Description
Device Type	Identifies the sender device type as defined in this specification
Message Type	Identifies the message type as defined in this specification. 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
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 sub-clause 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 it's receiver on before going to power savings mode. Used for commanding.

- 0: Commands pending NDB/data request exchange must 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 is 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 4-6)

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 section sub-clause 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 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 vendor 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 recalculate length and CRC values replaces 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, Table 95.

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 acknowledgment 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 acknowledgment 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

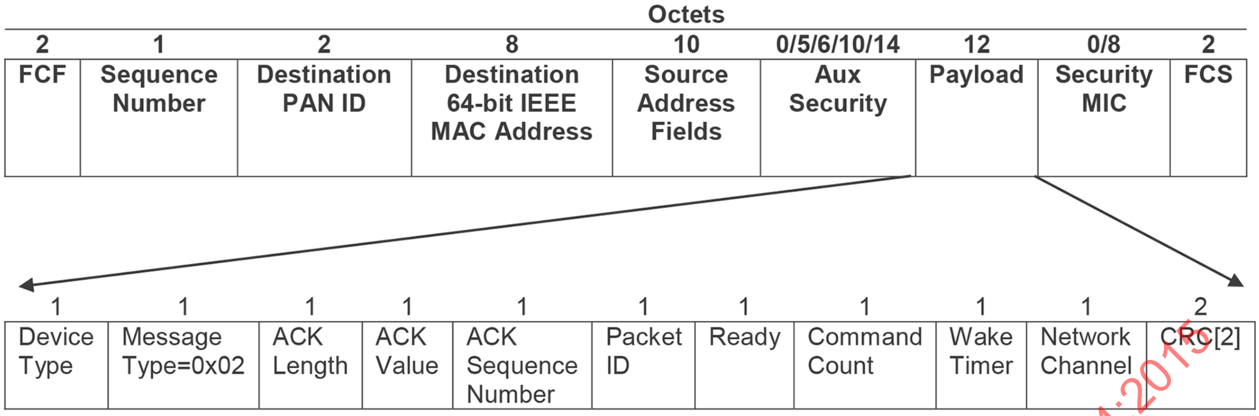


Figure 33 — Application ACK/NAK Packet Format

Table 127 — Application ACK/NAK Packet Payload Format

Device Type	Identifies device type of ACK sender
Message Type	0x02 indicates this is an acknowledgment 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
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.