INTERNATIONAL STANDARD

ISO/IEC 18000-62

First edition
2012-07-15

Information technology — Radio frequency identification for item management —

Part 62: Parameters for air interface communications at 860 MHz to 960 MHz Type B

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

Partie 62: Paramètres de communications d'une interface radio entre 860 MHz et 960 MHz, Type B

Reference number
ISO/IEC 18000-62:2012(E)

Contents

Page

Foreword

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

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

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

ISO/IEC 18000-62 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

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 describes a passive backscatter radio frequency identification (RFID) system that supports the following system capabilities:

- identification and communication with multiple tags in the field;
- selection of a subgroup of tags for identification or with which to communicate;
- reading from and writing to or rewriting data many times to individual tags;
- user-controlled permanently lockable memory;
- data integrity protection;
- Interrogator-to-tag communications link with error detection;
- tag-to-Interrogator communications link with error detection;
- support for both passive back-scatter tags with or without batteries.

This part of ISO/IEC 18000 specifies the physical and logical requirements for a passive-backscatter, RFID system operating in the 860 MHz to 960 MHz frequency range. The system comprises Interrogators, also known as readers, and tags, also known as labels.

An Interrogator transmits information to a tag by modulating an RF signal in the 860 MHz to 960 MHz frequency range. The tag receives both information and operating energy from this RF signal. Passive tags are those which receive all of their operating energy from the Interrogator's RF waveform. If tags maintain a battery then they may operate using some passive principles; however, they do not necessarily get all their operating energy from the Interrogator's RF waveform.

An Interrogator receives information from a tag by transmitting a continuous-wave (CW) RF signal to the tag; the tag responds by modulating the reflection coefficient of its antenna, thereby backscattering an information signal to the Interrogator. The system is Interrogator-Talks-First (ITF), meaning that a tag modulates its antenna reflection coefficient with an information signal only after being directed to do so by an Interrogator.

Interrogators and tags are not required to talk simultaneously; rather, communications are half-duplex, meaning that Interrogators talk and tags listen, or vice versa.

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

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

The holders of these patent rights have assured 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 on the declared patents may be obtained from:

Contact details	
Patent Holder Legal Name: Impinj, Inc.	**Contact for license application** Name & Department: Chris Diorio, CTO Address: 701 N 34th St Suite 300, Seattle, WA 98103, USA Tel.: +1 206 834 1115 Fax: +1 206 517 5262 E-mail: diorio@impinj.com URL (optional): www.impinj.com
Patent Holder Legal Name: NXP B.V.	**Contact for license application** Name & Department: Aaron Waxler, Intellectual Property & Licensing Address: 411 East Plumeria, San José, CA 95134-1924, USA Tel.: +1 914 860 4296 E-mail: Aaron.Waxler@nxp.com

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

The latest information on IP that may be applicable to this part of ISO/IEC 18000 can be found at www.iso.org/patents

Information technology — Radio frequency identification for item management —

Part 62: Parameters for air interface communications at 860 MHz to 960 MHz Type B

1 Scope

This part of ISO/IEC 18000 defines the air interface for radio frequency identification (RFID) devices operating in the 860 MHz to 960 MHz Industrial, Scientific, and Medical (ISM) band used in item management applications. It 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. It defines the forward and return link parameters for technical attributes including, but not limited to, operating frequency, operating channel accuracy, occupied channel bandwidth, maximum effective isotropic 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. It further defines the communications protocol used in the air interface.

This part of ISO/IEC 18000 specifies the physical and logical requirements for a passive-backscatter, Interrogator-Talks-First (ITF) systems. The system comprises Interrogators, also known as readers, and tags, also known as labels. An Interrogator receives information from a tag by transmitting a continuous-wave (CW) RF signal to the tag; the tag responds by modulating the reflection coefficient of its antenna, thereby backscattering an information signal to the Interrogator. The system is ITF, meaning that a tag modulates its antenna reflection coefficient with an information signal only after being directed to do so by an Interrogator.

In detail, this part of ISO/IEC 18000 contains Type B.

Type B uses Manchester in the forward link and an adaptive binary-tree collision-arbitration algorithm.

This part of ISO/IEC 18000 specifies

— physical interactions (the signalling layer of the communication link) between Interrogators and tags,

— Interrogator and tag operating procedures and commands,

— the collision arbitration scheme used to identify a specific tag in a multiple-tag environment.

2 Conformance

2.1 Claiming conformance

To claim conformance with this part of ISO/IEC 18000, an Interrogator or tag shall comply with all relevant clauses of this part of ISO/IEC 18000, except those marked as “optional”. The Interrogator or tag shall also operate within local radio regulations, which can further restrict operation.

Relevant conformance test methods are provided in ISO/IEC TR 18047-6.

Conformance can also require a license from the owner of any intellectual property utilized by said device.

2.2 Interrogator conformance and obligations

To conform to this part of ISO/IEC 18000, an Interrogator shall

— support Type B

— implement the mandatory commands defined in this part of ISO/IEC 18000;

— modulate/transmit and receive/demodulate a sufficient set of the electrical signals defined in the signalling layer of this part of ISO/IEC 18000 to communicate with conformant tags; and

— operate within the applicable local regulations.

To conform to this part of ISO/IEC 18000, an Interrogator may

— implement any subset of the optional commands defined in this part of ISO/IEC 18000, and

— implement any proprietary and/or custom commands in conformance with this part of ISO/IEC 18000.

To conform to this part of ISO/IEC 18000, the Interrogator shall not

— implement any command that conflicts with this part of ISO/IEC 18000 or any of the parts 61, 63 and 64, or

— require the use of an optional, proprietary, or custom command to meet the requirements of this part of ISO/IEC 18000.

2.3 Tag conformance and obligations

To conform to this part of ISO/IEC 18000, a tag shall:

— support Type B;

— operate over the frequency range from 860 MHz to 960 MHz, inclusive;

— implement the mandatory commands defined in this part of ISO/IEC 18000 for the supported types;

— modulate a backscatter signal only after receiving the requisite command from an Interrogator; and

— conform to local radio regulations.

To conform to this part of ISO/IEC 18000, a tag may

— implement any subset of the optional commands defined in this part of ISO/IEC 18000; and

— implement any proprietary and/or custom commands as defined in 6.2.7.4 and 6.2.7.5.

To conform to this part of ISO/IEC 18000, a tag shall not:

— implement any command that conflicts with this part of ISO/IEC 18000 or any of the parts 61, 63 and 64;

— require the use of an optional, proprietary, or custom command to meet the requirements of this part of ISO/IEC 18000; or

— modulate a backscatter signal unless commanded to do so by an Interrogator using the signalling layer defined in this part of ISO/IEC 18000.

3 Normative references

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

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

ISO/IEC 15961, *Information technology — Radio frequency identification (RFID) for item management — Data protocol: application interface*

ISO/IEC 15962, *Information technology — Radio frequency identification (RFID) for item management — Data protocol: data encoding rules and logical memory functions*

ISO/IEC 18000-1, *Information technology — Radio frequency identification for item management — Part 1: Reference architecture and definition of parameters to be standardized*

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

4 Terms, definitions, symbols and abbreviated terms

4.1 Terms and definitions

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

4.1.1
collision arbitration loop
algorithm used to prepare for and handle a dialogue between an Interrogator and a tag

NOTE This is also known as collision arbitration.

4.1.2
physical layer
data coding and modulation waveforms used in Interrogator-to-tag and tag-to-Interrogator signalling

4.2 Symbols

Cht	carrier high-level tolerance
Clt	carrier low-level tolerance
D	modulation depth of data coding pulse
f_c	frequency of operating field (carrier frequency)
M	number of subcarrier cycles per symbol
Mi	modulation index
Mb	modulation lower tolerance Type B
M_h	RF signal envelope ripple (overshoot)
M_l	RF signal envelope ripple (undershoot)
T_f or $T_{f,10\text{-}90\%}$	RF signal envelope fall time
T_{hf}	FHSS signal envelope fall time

T_{hr} FHSS signal envelope rise time

T_{hs} time for an FHSS signal to settle to within a specified percentage of its final value

T_r or $T_{r,10\text{-}90\%}$ RF signal envelope rise time

Tbmf Manchester fall time

Tbmr Manchester rise time

Tcf carrier fall time

Tcr carrier rise time

Tcs carrier steady state time

Tf fall time

Tfhf carrier FHSS fall time

Tfhr carrier FHSS rise time

Tfhs carrier FHSS steady time

Tr rise time

Trlb return link bit time

$xxxx_2$ binary notation

$xxxx_h$ hexadecimal notation

4.3 Abbreviated terms

AFI application family identifier

ASIC application specific integrated circuit

ASK Amplitude Shit Keying

CRC cyclic redundancy check

CRC-16 sixteen bit CRC

DSSS direct sequence spread spectrum

FHSS frequency hopping spread spectrum

ITF Interrogator-talks-first

NOTE The common usage is RTF (Reader-talks-first) but the more precise term is ITF, which is used throughout this part of ISO/IEC 18000.

LSB least significant bit

MSB most significant bit

NRZ non return to zero

ppm parts per million

RFU reserved for future use

SOF start of frame

Word 16 bits

5 Overview

5.1 Parameter tables

Table 1, Table 2, Table 3 and Table 4 contain the parameters for Type B in accordance with ISO/IEC 18000-1. Detailed description of the operating modes and parameters are specified in the subsequent clauses.

Table 1 — Interrogator to tag link parameters

Ref.	Parameter Name	Description
Int:1	Operating Frequency Range	860 MHz – 960 MHz, as required by the local regulations
Int:1a	Default Operating Frequency	In accordance with the local radio regulations.
Int:1b	Operating Channels (spread-spectrum systems)	In accordance with the local radio regulations.
Int:1c	Operating Frequency Accuracy	In accordance with the local radio regulations.
Int:1d	Frequency Hop Rate (frequency-hopping [FHSS] systems)	Where FHSS is permitted, the hop rate shall be in accordance with the local radio regulations.
Int:1e	Frequency Hop Sequence (frequency-hopping [FHSS] systems)	In accordance with the local radio regulations. Where not specified by such regulations a pseudo-random hopping sequence shall be used that ensures an even distribution of transmissions over available channels.
Int:2	Occupied Channel Bandwidth	In accordance with the local radio regulations.
Int:2a	Minimum Receiver Bandwidth	In accordance with the local radio regulations.
Int:3	Interrogator Transmit Maximum EIRP	In accordance with the local radio regulations.
Int:4	Interrogator Transmit Spurious Emissions	In accordance with the local radio regulations.
Int:4a	Interrogator Transmit Spurious Emissions, In-Band (spread-spectrum systems)	In accordance with the local radio regulations.
Int:4b	Interrogator Transmit Spurious Emissions, Out-of-Band	In accordance with the local radio regulations.
Int:5	Interrogator Transmitter Spectrum Mask	In accordance with the local radio regulations.
Int:6	Timing	See below Int: 6x.
Int:6a	Transmit-to-Receive Turn-Around Time	The Interrogator transmit/receive settling time shall not exceed 85 µs.
Int:6b	Receive-to-Transmit Turn-Around Time	As determined by the communication protocol – refer Tag: 6a.
Int:6c	Dwell Time or Interrogator Transmit Power-On Ramp	1500 µs, maximum settling time
Int:6d	Decay Time or Interrogator Transmit Power-Down Ramp	Maximum 1 ms.
Int:7	Modulation	Amplitude Modulation.
Int:7a	Spreading Sequence (direct-sequence [DSSS] systems)	Not applicable.
Int:7b	Chip Rate (spread-spectrum systems)	Not applicable.

Ref.	Parameter Name	Description
Int:7c	Chip Rate Accuracy (spread-spectrum systems)	Not applicable.
Int:7d	Modulation Depth	Nominal 18% or 100% .
Int:7e	Duty Cycle	In accordance with the local radio regulations.
Int:7f	FM Deviation	Not applicable.
Int:8	Data Coding	Manchester bi-phase
Int:9	Bit Rate	10 kbit/s or 40 kbit/s as constrained by the local radio regulations.
Int:9a	Bit Rate Accuracy	100 ppm
Int:10	Interrogator Transmit Modulation Accuracy	Not applicable..
Int:11	Preamble	Yes, See clause 6.1.7.3
Int:11a	Preamble Length	9 bits. See clause 6.1.7.3
Int:11b	Preamble Waveform(s)	See clause 6.1.7.3
Int:11c	Bit Sync Sequence	See clause 6.1.7.3
Int:11d	Frame Sync Sequence	Not Applicable.
Int:12	Scrambling (spread-spectrum systems)	Not Applicable.
Int:13	Bit Transmission Order	MSB is transmitted first
Int:14	Wake-up process	Presence of an appropriate RF signal at the tag followed by a wake-up command as required by the tag type. See relevant clauses.
Int:15	Polarization	Interrogator dependent. Not defined in this part of ISO/IEC 18000.

Table 2 — Tag to Interrogator link parameters

Ref.	Parameter Name	Description
Tag:1	Operating Frequency Range	860 MHz – 960 MHz, inclusive
Tag:1a	Default Operating Frequency	The tag shall respond to an Interrogator signal within the frequency range specified in Tag: 1.
Tag:1b	Operating Channels (spread-spectrum systems)	The tag shall respond to an Interrogator signal within the frequency range specified in Tag: 1.
Tag:1c	Operating Frequency Accuracy	The tag shall respond to an Interrogator signal within the frequency range specified in Tag: 1.
Tag:1d	Frequency Hop Rate (frequency-hopping [FHSS] systems)	Not applicable.
Tag:1e	Frequency Hop Sequence (frequency-hopping [FHSS] systems)	Not applicable.
Tag:2	Occupied Channel Bandwidth	In accordance with the local regulations
Tag:3	Transmit Maximum EIRP	In accordance with the local regulations
Tag:4	Transmit Spurious Emissions	In accordance with the local regulations
Tag:4a	Transmit Spurious Emissions, In-Band (spread spectrum systems)	In accordance with the local regulations
Tag:4b	Transmit Spurious Emissions, Out-of-Band	In accordance with the local regulations
Tag:5	Transmit Spectrum Mask	In accordance with the local regulations

Ref.	Parameter Name	Description
Tag:6a	Transmit-to-Receive Turn-Around Time	400 μs
Tag:6b	Receive-to-Transmit Turn-Around Time	Range 85 to 460 μs (see clause 6.1.8.2)
Tag:6c	Dwell Time or Transmit Power-On Ramp	Not applicable.
Tag:6d	Decay Time or Transmit Power-Down Ramp	Not applicable.
Tag:7	Modulation	Bi-state amplitude modulated backscatter.
Tag:7a	Spreading Sequence (direct sequence [DSSS] systems)	Not applicable.
Tag:7b	Chip Rate (spread spectrum systems)	Not applicable.
Tag:7c	Chip Rate Accuracy (spread spectrum systems)	Not applicable.
Tag:7d	On-Off Ratio	The tag Delta RCS (Varying Radar Cross Sectional area) affects system performance. A typical value is greater than 0.005 m^2.
Tag:7e	Subcarrier Frequency	Not applicable.
Tag:7f	Subcarrier Frequency Accuracy	Not applicable.
Tag:7g	Subcarrier Modulation	Not applicable.
Tag:7h	Duty Cycle	The tag shall transmit its response when commanded to do so by the Interrogator.
Tag:7i	FM Deviation	Not applicable.
Tag:8	Data Coding	Bi-phase space (FM0)
Tag:9	Bit Rate	Typical 40 kbit/s or 160 kbit/s (subject to tag clock tolerance see Table 8), The return bit rate selection for 160 kbit/s is defined in clause 6.1.7.4.5
Tag:9a	Bit Rate Accuracy	+/- 15% (refer to Table 8)
Tag:10	Tag Transmit Modulation Accuracy (frequency-hopping [FHSS] systems	Not applicable.
Tag:11	Preamble	The preamble is defined in clause 6.1.4.6
Tag:11a	Preamble Length	16 bits made up of a quiet period, followed by sync, followed by a code violation followed by an orthogonal code.
Tag:11b	Preamble Waveform	Bi-phase encoded data ‘1’.
Tag:11c	Bit-Sync Sequence	Included in the preamble.
Tag:11d	Frame-Sync Sequence	Included in the preamble.
Tag:12	Scrambling (spread-spectrum systems)	Not applicable.
Tag:13	Bit Transmission Order	MSB is transmitted first
Tag:14	Reserved	Deliberately left blank.
Tag:15	Polarization	Product design feature. Not defined in this part of ISO/IEC 18000.
Tag:16	Minimum Tag Receiver Bandwidth	860 – 960 MHz

Table 3 — Protocol parameters

Ref.	Parameter Name	Description
P:1	Who talks first	Interrogator
P:2	Tag addressing capability	See clause 6.2.2
P:3	Tag ID	Contained in tag memory and accessible by means of a command.
P:3a	Tag ID Length	64 bits.
P:3b	Tag ID Format	See clause B.1.2
P:4	Read size	Addressable in byte blocks.
P:5	Write Size	Addressable in byte blocks. Writing in blocks of 1, 2, 3 or 4 bytes. See details in relevant clauses.
P:6	Read Transaction Time	A single tag can typically be identified and have its first 128 bits of user memory read in less than 10 ms. This time may vary depending on the data rate used as constrained by the local radio regulations.
P:7	Write Transaction Time	Once a tag has been identified and selected, a 32-bit data block can typically be written in less than 20 ms. This time may vary depending on the data rate used as constrained by the local radio regulations.
P:8	Error detection	Interrogator to tag: CRC-16 Tag to Interrogator: CRC-16
P:9	Error correction	No forward error correction code used. Errors are handled by signalling an error to the Interrogator that then repeats its last transmission.
P:10	Memory size	No minimum user memory size is specified, but if user memory is provided it shall be an integer multiples of 4 bytes.
P:11	Command structure and extensibility	Several command codes are reserved for future use.

Table 4 — Anti-collision parameters

Ref.	Parameter Name	Description
A:1	Type (Probabilistic or Deterministic)	Probabilistic
A:2	Linearity	Essentially linear up to 2^{256} tags depending on size of data content.
A:3	Tag inventory capacity	The algorithm permits the reading of not less than 250 tags in the reading zone of the Interrogator.

6 Type B

6.1 Physical layer and data coding

6.1.1 Interrogator power-up waveform

The Interrogator power-up waveform shall comply with the mask specified in Figure 1 and Table 5.

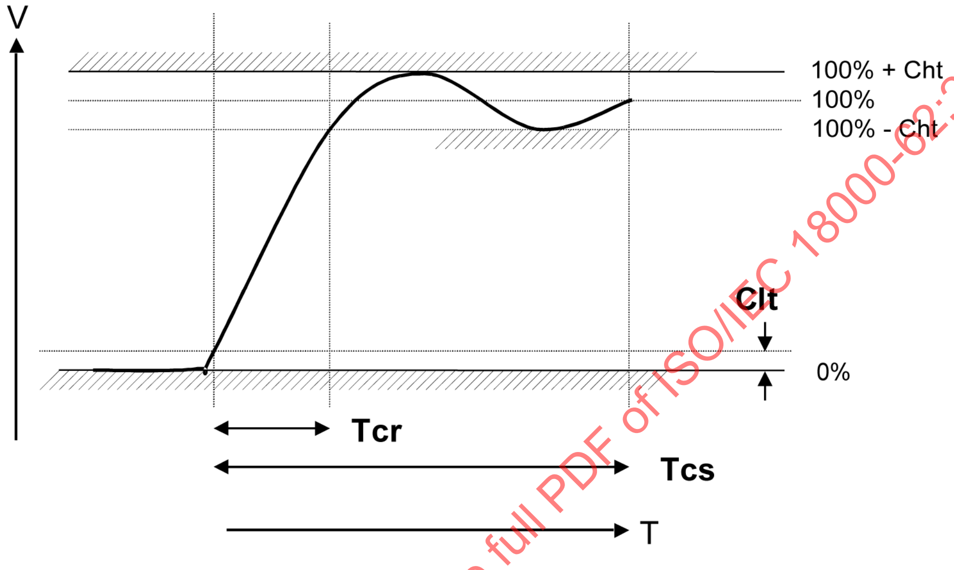


Figure 1 — Interrogator power-up waveform

Table 5 — Interrogator power-up waveform parameter values

Parameter	Min	Max
Tcs		1500µs
Tcr	1 µs	500 µs
Cht		10%
Clt		1%

6.1.2 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 6.

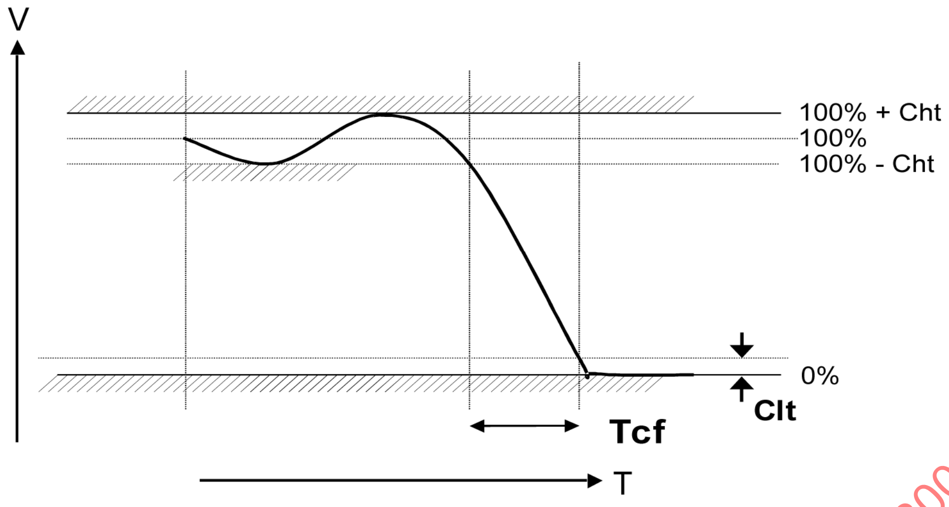


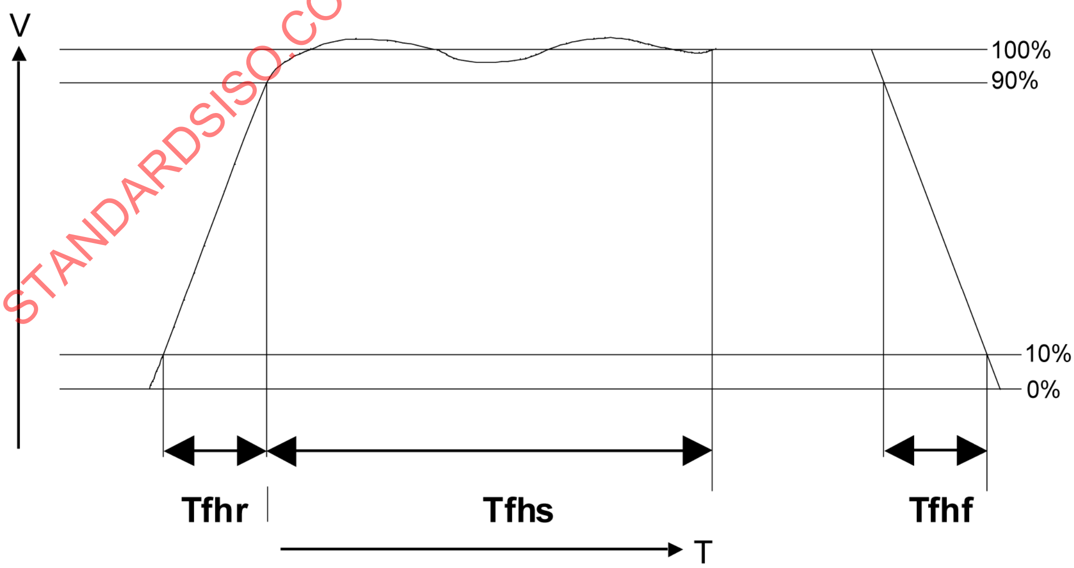
Figure 2 — Interrogator power-down waveform

Table 6 — Interrogator power-down timings

Parameter	Min	Max
Tcf	1 µs	500 µs
Cht		± 5% of steady state (100 %) level
Clt		1 %

6.1.3 Frequency hopping carrier rise and fall times

When the Interrogator operates in the frequency operating hopping spread spectrum mode (FHSS), the carrier rise and fall times shall conform to the characteristics specified in Figure 3 and Table 7. The Interrogator shall complete a frequency hop in a time not exceeding 30 µs (to ensure that the tag is not reset by the frequency hop). The frequency hop is measured from the beginning of Tfhf to the end of Tfhr.



NOTE Ripple is ± 5 % of 100 % of steady state level

Figure 3 — FHSS carrier rise and fall characteristics

Table 7 — FHSS carrier rise and fall parameters

Parameter	Min	Max
Tfhr		30 µs
Tfhs	400 µs	
Tfhf		30 µs

6.1.4 FM0 return link

6.1.4.1 FM0 return link general

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

6.1.4.2 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.

6.1.4.3 Data rate

The return link datarate shall be 40 or 160 kbit/s, which may be selected at the time of tag configuration. The Interrogator shall be able to read and decode the tag reply at either datarate without the need to have prior knowledge of the tag configuration in order to handle mixed populations.

6.1.4.4 Data coding

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

One symbol period Trlb, as specified in Table 8, 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 8 — Return link parameters

Data rate	Trlb	Tolerance	Note
40kbit/s	25 µs	+/-15%	Chip set to 40kbit/s Return Link Data Rate
160kbit/s	6.25 µs	+/-15%	Chip set to 160kbit/s Return Link Data Rate

Coding of data is MSB first. Figure 4 illustrates the coding for the 8 bits of 'B1'.

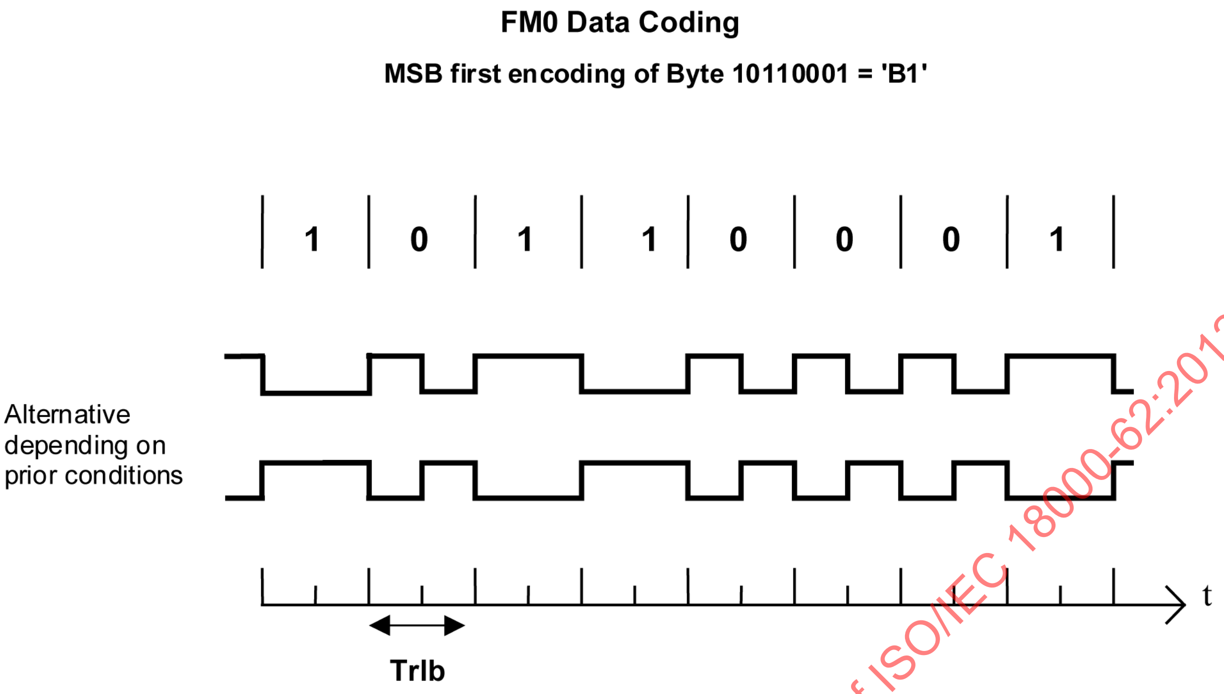


Figure 4 — Tag to Interrogator data coding

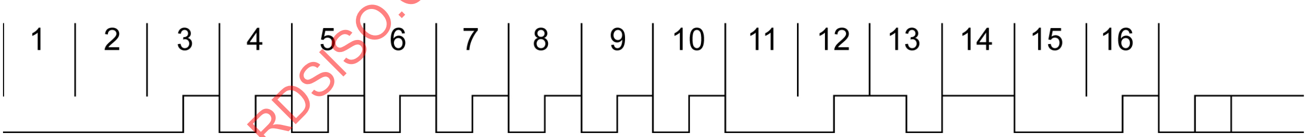
6.1.4.5 Message format

A return Link Message consists of n data bits preceded by the Preamble. 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 Figure 5. There are multiple code violations (sequences not conforming to FM0 rules) that act as a frame marker for the transition from Preamble to Data.

6.1.4.6 Return preamble

The return preamble is a sequence of backscatter modulation specified in Figure 5.



NOTE The high state represents high reflectivity and the low state represents low reflectivity.

Figure 5 — Preamble waveform

Changing the tag's modulator switch from the high impedance state to the low impedance state causes a change in the incident energy to be back-scattered, see Figure 6.

The tag shall execute backscatter, a half-low and half-high sent by the tag defined as follows:

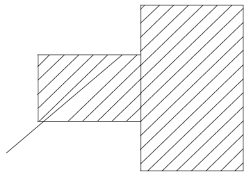


Figure 6 — Return link bit coding

6.1.4.7 Cyclic redundancy check (CRC)

6.1.4.7.1 CRC General

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.

6.1.4.7.2 Interrogator to tag 16-bit CRC-16

6.1.4.7.2.1 Interrogator to tag CRC-16 general

The 16-bit CRC shall be calculated on all the command bits after the SOF up to but not including the first CRC bit.

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.

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

The CRC may be implemented in one of two ways:

6.1.4.7.2.2 Inversion of incoming CRC bits by the tag

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

6.1.4.7.2.3 Non-inversion of incoming CRC bits by the tag

If the received CRC bits are not inverted before clocking into the register, then after the LSB CRC bit is clocked into the register the 16-bit CRC register will have the value $1D0F_h$

6.1.4.7.3 Tag to Interrogator 16-bit CRC-16

6.1.4.7.3.1 Tag to Interrogator CRC-16 general

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

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

The most significant byte shall be transmitted first, see Table 9. The most significant bit of each byte shall be transmitted first.

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.

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

Table 9 — 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	

The CRC may be implemented in one of two ways.

6.1.4.7.3.2 Inversion of incoming CRC bits by the Interrogator

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

6.1.4.7.3.3 Non-inversion of incoming CRC bits by the Interrogator

If the received CRC bits are not inverted before clocking, the CRC register will have the value $1D0F_h$.

6.1.5 Manchester forward link

6.1.5.1 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, as shown in Figure 7, Figure 8, Table 10 and Table 11.

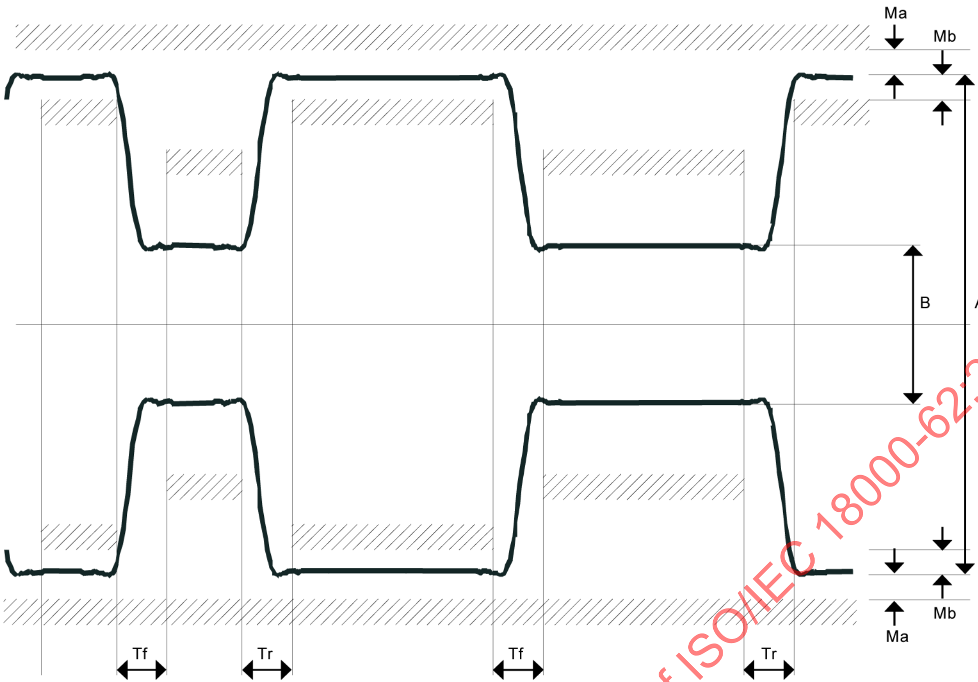


Figure 7 — 100 % modulation (example of 40 kbit/s signal)

Table 10 — Parameter for 100% modulation

Parameter	Minimum	Nominal	Maximum
Mi = (A-B)/(A+B)	90	100	100
Ma	0		0.03 (A-B)
Mb	0		0.03 (A-B)
Tr	0 µs	1.8 µs	0.1 / $f_{Datarate}$
Tf	0 µs	1.8 µs	0.1 / $f_{Datarate}$

NOTE Tr and Tf measured from 10% (A-B) to 90% (A-B)

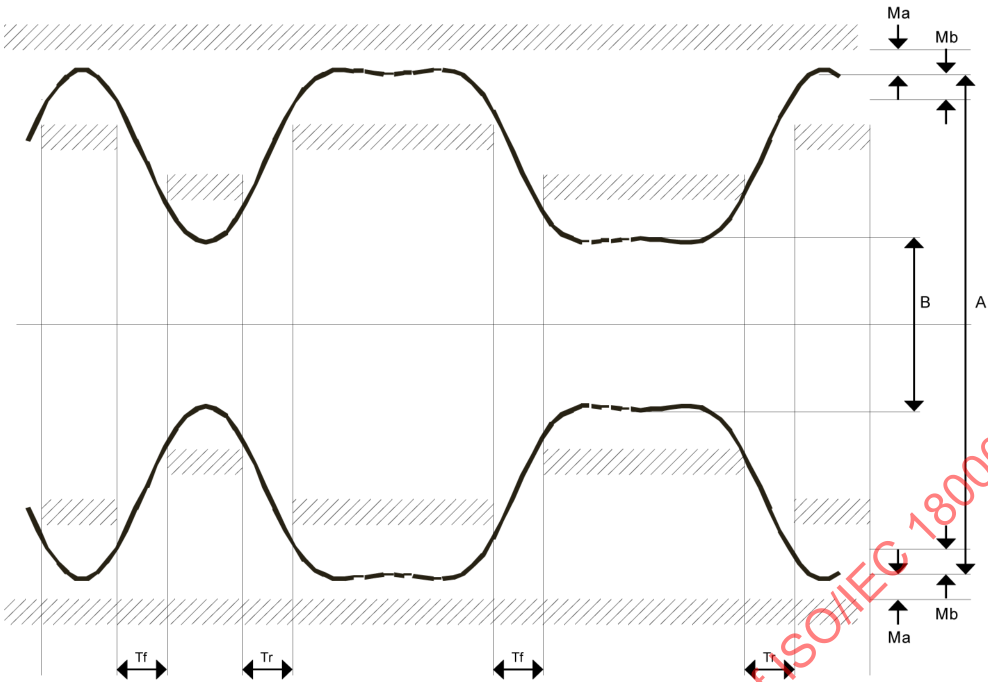


Figure 8 — 18% modulation (example of 8 kbit/s signal)

Table 11 — Parameter for 18% modulation

Parameter	Minimum	Nominal	Maximum
Mi = (A-B)/(A+B)	15%	18%	20%
Ma	0		0.05 (A-B)
Mb	0		0.05 (A-B)
Tr	0 µs		0.17 / $f_{Datarate}$
Tf	0 µs		0.17 / $f_{Datarate}$

NOTE Tr and Tf measured from 10% (A-B) to 90% (A-B)

6.1.5.2 Bit coding of forward link fields

Data is Manchester encoded as per Figure 9.

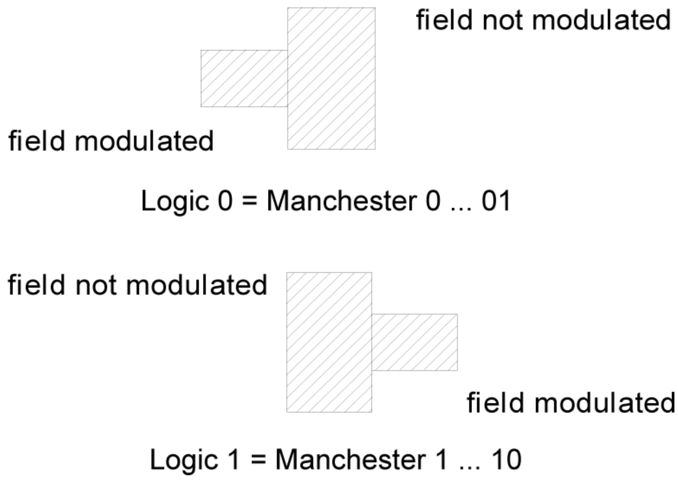


Figure 9 — Forward link bit coding

6.1.6 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 modulation index specification is defined in 6.1.5.1. 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 6.2.7

Each command and each response are contained in a frame. The respective frames are specified in 6.1.7 and 6.1.8. Each command consists of the following fields:

- Preamble Detect (no modulation of the RF carrier),
- Preamble,
- Delimiter,
- Command code,

- Parameter fields (depending on the command),
- Application data fields (depending on the command), and
- CRC-16.

Each response consists of the following fields:

- Quiet (no modulation of the RF carrier),
- Return Preamble,
- Application data fields, and
- CRC-16.

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, to support the frame detection.

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

The MSByte shall be the byte at the specified address. The LSByte 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 MSByte 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).

6.1.7 Command format

6.1.7.1 Command format general

The command frame, as shown in Figure 10, consist of the following fields:

- Preamble Detect,
- Preamble,
- Delimiter,
- Command,
- Parameter (and data fields), and
- CRC-16.

Preamble Detect	Preamble	Delimiter	Command	Parameter	Data	CRC-16

Figure 10 — General command format

6.1.7.2 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.

6.1.7.3 PREAMBLE

The preamble is equivalent to 9 bits of Manchester 0 in NRZ format.

010101010101010101

6.1.7.4 Delimiters

6.1.7.4.1 Delimiters general

Four delimiters are defined.

6.1.7.4.2 Start delimiter 1

In NRZ format; includes Manchester errors; spaces ignored

11 00 11 10 10 - Delimiter 1

6.1.7.4.3 Start delimiter 2

In NRZ format; includes Manchester errors; spaces ignored

01 01 11 00 11 - Delimiter 2

Reserved for future use

6.1.7.4.4 Start delimiter 3

In NRZ format; includes Manchester errors; spaces ignored

00 11 10 01 01 - Delimiter 3

Reserved for future use

6.1.7.4.5 Start delimiter 4

In NRZ format; includes Manchester errors; spaces ignored

11 01 11 00 10 1 - Delimiter 4

Delimiter 4 supports all commands as delimiter 1, however the return data rate is 4 times the forward link data rate. The supported data rates are defined in Clause 5.

6.1.7.5 CRC-16

See Annex A.

6.1.8 Response format

6.1.8.1 Response format general

The response, as shown in Figure 11, consists of the following fields:

- Quiet,
- Return Preamble,
- Data fields, and
- CRC-16.

Quiet	Return Preamble	Data	CRC-16

Figure 11 — General response format

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.

6.1.8.2 QUIET

The tag shall not backscatter for 16 * $T_{return\ data\ rate}$ – 0.75 * $T_{forward\ data\ rate}$. The duration of the quiet time is determined by the communication speed of the return link.

6.1.8.3 CRC-16

See Annex A.

6.1.9 WAIT

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 6.2.7.9.11. 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.

6.1.10 Examples of a command packet

Examples of command packets are given in Figure 12 and Figure 13.

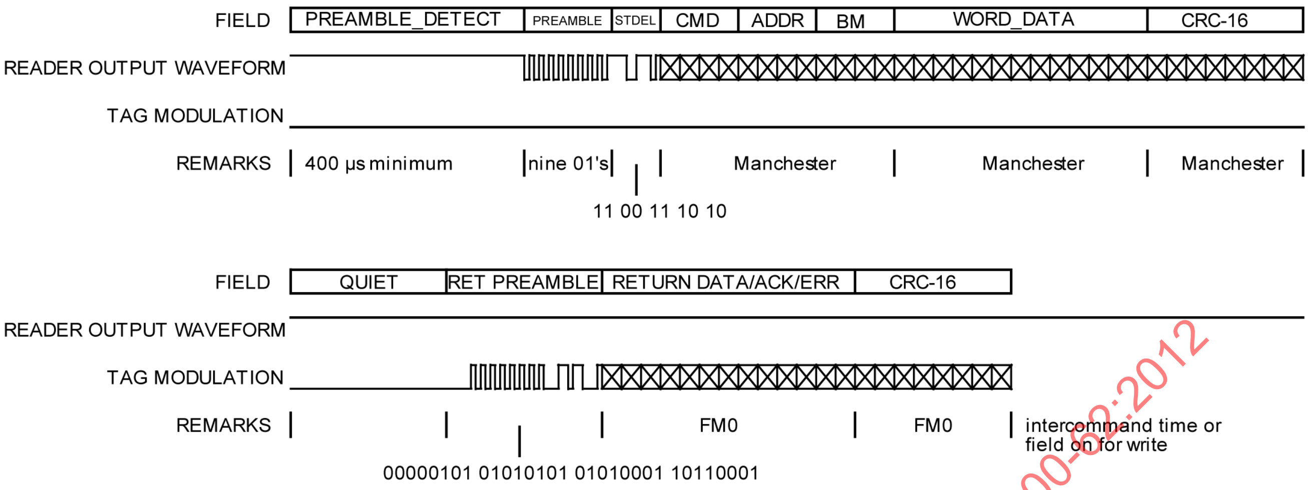


Figure 12 — Sample command/response packets for GROUP_SELECT (40 kbit/s on forward and return link)

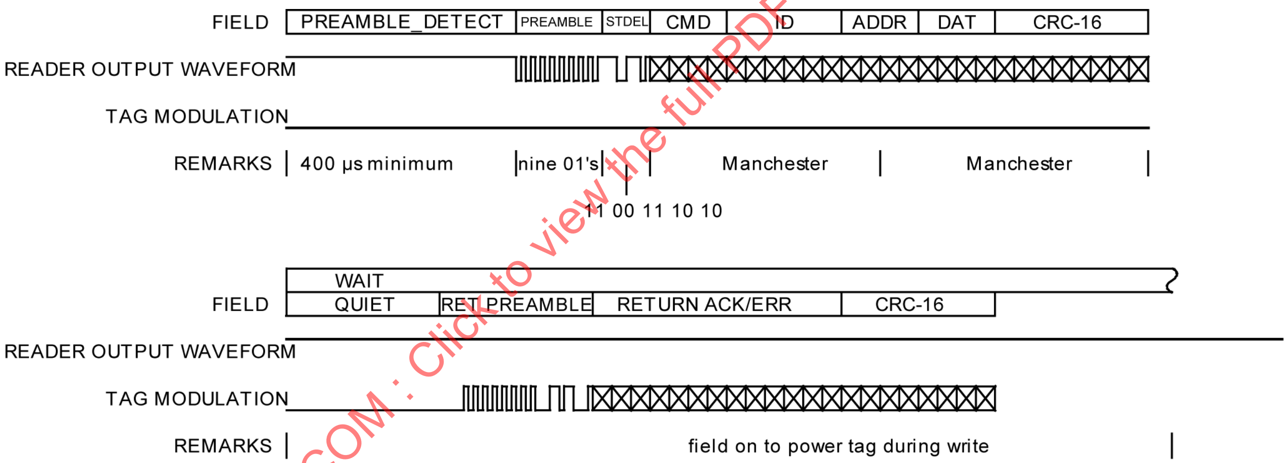


Figure 13 — Sample command/response packets for WRITE (40 kbit/s on forward and return link)

6.1.11 Communication sequences at packet level

Figure 14 and Figure 15 show examples of communication sequences at the packet level. Figure 14 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 mis-calibrate as a result of the spurious signals that may be generated.

Figure 15 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 14 — 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 15 — Command sequence with a hop between response and next command

6.2 Btree protocol and collision arbitration

6.2.1 Definition of data elements, bit and byte ordering

6.2.1.1 Unique ID

See Annex B and Annex C.

6.2.1.2 CRC-16

See Annex A.

6.2.1.3 FLAGS

6.2.1.3.1 FLAGS general

The tag shall support a field of 8 flags. This field is called FLAGS, and is shown in Table 12.

Table 12 — FLAGS

Bit	Name
FLAG1 (LSB)	DE_SB (Data_Exchange Status Bit)
FLAG2	WRITE_OK
FLAG3	BATTERY_POWERED
FLAG4	BATTERY_OK
FLAG5	0 (RFU)
FLAG6	0 (RFU)
FLAG7	0 (RFU)
FLAG8 (MSB)	0 (RFU)

6.2.1.3.2 Data exchange status bit (DE_SB)

The tag shall set this bit when the tag goes into the DATA_EXCHANGE state and keeps 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.

6.2.1.3.3 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.

6.2.1.3.4 BATTERY_POWERED

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

6.2.1.3.5 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.

NOTE BATTERY_POWERED indicates whether the tag should have a battery, while BATTERY_OK reports the status of the battery. BATTERY_POWERED could be hardwired.

6.2.2 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 kbits.

NOTE This structure allows future extensions of the maximum memory capacity, by the use of additional commands to be defined, when required.

6.2.3 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.

6.2.4 Overall protocol description, Btree protocol

6.2.4.1 Tag states

The tag has four major states, as shown in Figure 16:

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.

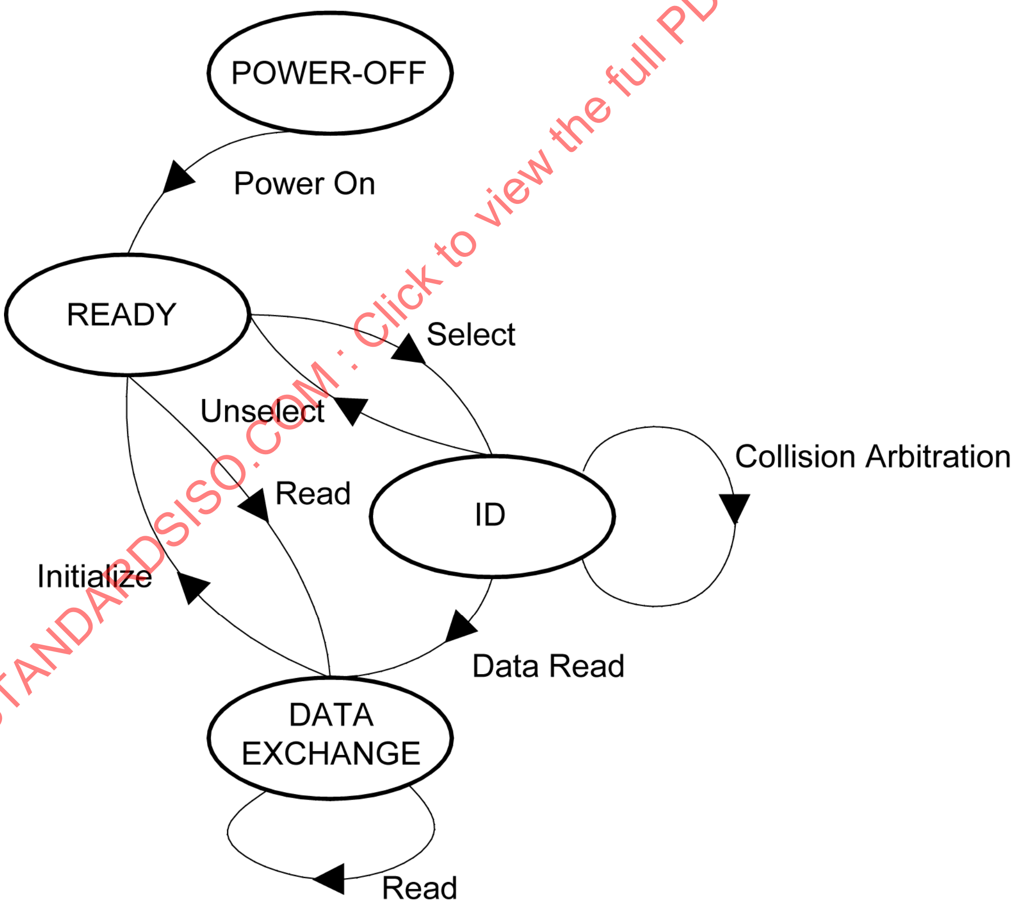


Figure 16 — State diagram

NOTE This diagram does not show that the tag goes into POWER-OFF, state from all other states when the Interrogator field is off and the tag operation is no longer supported by the tag power buffer.

The state diagram only shows an overview of the possible transitions. Details are specified in Table 14.

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 a 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 14.

6.2.4.2 Detailed command processing

As shown in Table 13, commands shall be active in states marked with "X" and neither causes a state change, nor cause a response in the other states.

Table 13 — 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_VERIFY_4BYTE	X	X	X
WRITE	X	X	X
WRITE4BYTE	X	X	X
WRITE4BYTE_MULTIPLE		X	X
WRITE_MULTIPLE		X	X
LOCK			X
QUERY_LOCK	X	X	X
FAIL_O		X	
SUCCESS_O		X	
DATA_READ_O		X	X
READ_FLAGS	X	X	X
READ_VARIABLE	X	X	X
READ_PORT	X	X	X
READ_UNADDRESSED	X	X	X
RESEND_O		X	

Table 14 — 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_VERIFY_4BYTE	ID no match or not WRITE_OK	READY
READY	READ_VERIFY_4BYTE	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
READY	READ_FLAGS	ID no match	READY
READY	READ_FLAGS	ID match	DATA_EXCHANGE
READY	READ_VARIABLE	ID no match	READY
READY	READ_VARIABLE	ID match	DATA_EXCHANGE
READY	READ_PORT	ID no match	READY
READY	READ_PORT	ID match	DATA_EXCHANGE
READY	READ_UNADDRESSED		DATA_EXCHANGE
ID	GROUP_UNSELECT_EQ	≠	ID
ID	GROUP_UNSELECT_NE	=	ID
ID	GROUP_UNSELECT_GT	≤	ID

Table 14 *(continued)*

Current state	Command	Condition	New state
ID	GROUP_UNSELECT_LT	≥	ID
ID	GROUP_UNSELECT_EQ_FLAGS	flag not set	ID
ID	GROUP_UNSELECT_NE_FLAGS	flag set	ID
ID	GROUP_UNSELECT_EQ	=	READY
ID	GROUP_UNSELECT_NE	≠	READY
ID	GROUP_UNSELECT_GT	>	READY
ID	GROUP_UNSELECT_LT	<	READY
ID	GROUP_UNSELECT_EQ_FLAGS	flag set	READY
ID	GROUP_UNSELECT_NE_FLAGS	flag not set	READY
ID	MULTIPLE_UNSELECT	≠ or not WRITE_OK	ID
ID	MULTIPLE_UNSELECT	= and WRITE_OK	READY
ID	GROUP_SELECT_EQ		ID
ID	GROUP_SELECT_NE		ID
ID	GROUP_SELECT_GT		ID
ID	GROUP_SELECT_LT		ID
ID	GROUP_SELECT_EQ_FLAGS		ID
ID	GROUP_SELECT_NE_FLAGS		ID
ID	FAIL		ID
ID	SUCCESS		ID
ID	RESEND		ID
ID	INITIALIZE		READY
ID	READ	ID no match	ID
ID	READ	ID match	DATA_EXCHANGE
ID	DATA_READ	ID no match	ID
ID	DATA_READ	ID match	DATA_EXCHANGE
ID	READ_VERIFY	ID no match or not WRITE_OK	ID
ID	READ_VERIFY	ID match and WRITE_OK	DATA_EXCHANGE
ID	READ_VERIFY_4BYTE	ID no match or not WRITE_OK	ID
ID	READ_VERIFY_4BYTE	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

Table 14 *(continued)*

Current state	Command	Condition	New state
ID	RESEND_O		ID
ID	READ_FLAGS	ID no match	ID
ID	READ_FLAGS	ID match	DATA_EXCHANGE
ID	READ_VARIABLE	ID no match	ID
ID	READ_VARIABLE	ID match	DATA_EXCHANGE
ID	READ_PORT	ID no match	ID
ID	READ_PORT	ID match	DATA_EXCHANGE
ID	FAIL_O		ID
ID	SUCCESS_O		ID
ID	DATA_READ_O	ID no match	ID
ID	DATA_READ_O	ID match	DATA_EXCHANGE
ID	READ_UNADDRESSED		DATA_EXCHANGE
DATA_EXCHANGE	INITIALIZE		READY
DATA_EXCHANGE	READ		DATA_EXCHANGE
DATA_EXCHANGE	DATA_READ		DATA_EXCHANGE
DATA_EXCHANGE	READ_VERIFY		DATA_EXCHANGE
DATA_EXCHANGE	READ_VERIFY_4BYTE		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
DATA_EXCHANGE	READ_UNADDRESSED		DATA_EXCHANGE
DATA_EXCHANGE	DATA_READ_O		DATA_EXCHANGE
DATA_EXCHANGE	READ_FLAGS		DATA_EXCHANGE
DATA_EXCHANGE	READ_VARIABLE		DATA_EXCHANGE
DATA_EXCHANGE	READ_PORT		DATA_EXCHANGE

6.2.5 Collision arbitration

6.2.5.1 Collision arbitration general

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, and
- A random generator (with two possible values 0 or 1).

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:

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

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

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

   All tags receiving FAIL command with 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 Tag ID again.

One of four possibilities now occurs:

4) If more than one tag transmits, the FAIL step 2 repeats. (Possibility 1)

5) 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)

6) 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 sends SUCCESS. All tags in the ID state shall decrement COUNT.

7) 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)

8) 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)

6.2.5.2 Special collision arbitration

In the case that parts of the user data are unique, or the probability of duplicated information is sufficient low, the commands FAIL_O, SUCCESS_O and DATA_READ_O may be used for collision arbitration. While the algorithm is equal to commands without _O, the ID used for the algorithm is either a 32 bit or 64 bit ID beginning at memory address ‘14’.

Additionally to the already mentioned use of GROUP_SELECT and GROUP_UNSELECT these commands may be used in combination with the 32-bit ID option of the _O command. This means in the case that a GROUP_SELECT only selects those tags that have all zeros in the upper 32 bits of a 64 bit Tag ID and after handling those checks whether no tags with having not all zeros in the upper 32 bits of a 64 bit Tag ID are still left.

6.2.6 Commands

Commands are divided into four functional groups:

- Selection commands,
- Identification commands,
- Data transfer commands, and
- Multiple commands.

Further, commands, as shown in Table 15, have one of the following types:

- Mandatory,
- Optional,
- Custom, and
- Proprietary.

Table 15 — Command classes

Code	Class	Number of possible codes
'00' - '0A', '0C', '15', '1E' - '3F'.	Mandatory	47
'0B', '0D' - '0F', '11' - '13', '17' - '1D', '40' - '9F'	Optional	110
'A0' - 'DF'	Custom	64
'10', '14', '16' 'E0' - 'FF'	Proprietary	35

6.2.7 Command types

6.2.7.1 Command types general

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

6.2.7.2 Mandatory

The command codes range from '00' to '0A', '0C', '15' and '1E' 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. Mandatory commands shall be implemented as specified in this part of ISO/IEC 18000.

6.2.7.3 Optional

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

Optional commands are commands that are specified within this part of ISO/IEC 18000. Interrogators shall be technically capable of performing all optional commands that are specified this part of ISO/IEC 18000 (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 this part of ISO/IEC 18000.

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

NOTE The command whose code ranges from ‘0B, 0D to 13 and '17' to '1D' 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 16.

6.2.7.4 Custom

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

Tags support them, at their option, to implement manufacturer specific functions. The only fields that can be customized are the parameters and the data fields.

Any custom command contains as its first parameter the IC manufacturer code. This allows IC manufacturers to implement custom commands without risking duplication of command codes and thus misinterpretation. Custom commands may be enabled by this part of ISO/IEC 18000, but they shall not be specified in this part of ISO/IEC 18000.

Custom commands may be enabled by this part of ISO/IEC 18000, but they shall not be specified in this part of ISO/IEC 18000. A custom command shall not solely duplicate the functionality of any mandatory or optional command defined in this part of ISO/IEC 18000 by a different method.

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

6.2.7.5 Proprietary

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

These commands are used by IC and tag manufacturers 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.

Proprietary commands may be enabled by this part of ISO/IEC 18000, but they shall not be specified in this part of ISO/IEC 18000.

A proprietary command shall not solely duplicate the functionality of any mandatory or optional command defined in this part of ISO/IEC 18000 by a different method.

6.2.7.6 Command codes and format

6.2.7.6.1 Command codes and format general

Command codes and formats are shown in Table 16.

Table 16 — 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'	Recommended	DATA_READ	ID	ADDRESS	none
'0C'	Mandatory	READ	ID	ADDRESS	none
'0D'	Recommended	WRITE	ID	ADDRESS	BYTE_DATA
'0E'	Recommended	WRITE_MULTIPLE	none	ADDRESS	BYTE_DATA
'0F'	Recommended	LOCK	ID	ADDRESS	none
'10'	Proprietary	IC manufacturer 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 4BYTE_ DATA
'1C'	Recommended	WRITE4BYTE_MULTIPLE	ADDRESS	BYTE_MASK	4BYTE_DATA
'1D'	Recommended	READ_VERIFY_4BYTE	ID	ADDRESS	none
'1E'-'3F'	Mandatory	RFU			
'40', '41'	Optional	FAIL_O	none	none	none
'42', '43'	Optional	SUCCESS_O	none	none	none
'44', '45'	Optional	DATA_READ_O	ID	ADDRESS	none
'46', '47'	Optional	RESEND_O	none	none	none

Table 16 *(continued)*

Command code	Type	Command name	Parameters		
‘48’ – ‘4F’	Optional	RFU			
‘50’	Optional	READ_FLAGS	ID	ADDRESS	none
‘51’	Optional	READ_VARIABLE	ID	ADDRESS	LENGTH
‘52’	Optional	READ_PORT	ID	ADDRESS	none
‘53’	Optional	READ_UNADDRESSED		ADDRESS	none
‘54’ – ‘9F’	Optional	RFU			
‘A0’ – ‘DF’	Custom	IC Manufacturer dependent			
‘E0’ – ‘FF’	Proprietary	IC Manufacturer dependent			

6.2.7.6.2 Command Fields

Command fields are shown in Table 17.

Table 17 — 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
LENGTH	1 byte
CRC-16	2 bytes

6.2.7.6.3 Tag responses

Tag responses are shown in Table 18.

Table 18 — Tag responses

Response code	Response name	Response size
'00'	*ACK*NOWLEDGE	1 byte
	*ACK*NOWLEDGE_NOK	1byte
'01'	*ACK*NOWLEDGE_OK	1byte
'FE'	ERROR_NOK	1byte
'FF'	ERROR	1byte
	ERROR_OK	1byte
n/a	WORD_DATA	8 bytes
N/a	VARIABLE DATA	LENGTH bytes
n/a	BYTE_DATA	1byte
	CRC-16	2 bytes
	ID	8 bytes

6.2.7.7 Selection commands

6.2.7.7.1 Selection commands general

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.

6.2.7.7.2 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, or
- GROUP_UNSELECT_LT,

has 3 arguments (parameter and data)

- ADDRESS,
- BYTE_MASK, and
- 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, or
- LT M LOWER THAN D.

The arguments of the comparison are shown in Table 19.

Table 19 — Comparison arguments

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 arguments of the command, are shown in Table 20.

Table 20 — Command arguments

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, and are shown in Table 21.

Table 21 — 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

6.2.7.7.3 Data comparison for selection command on flags

Each Select command of the following commands have two arguments (parameter and data):

- GROUP_SELECT_EQ_FLAGS,
- GROUP_SELECT_NE_FLAGS,
- GROUP_UNSELECT_EQ_FLAGS, or
- GROUP_UNSELECT_NE_FLAGS.

The two arguments (parameter and data) are:

- BYTE_MASK, or
- BYTE_DATA.

The tag shall make two possible comparisons:

- EQ FLAGS EQUAL D, or
- NE FLAGS NOT EQUAL D.

The arguments of the comparison are FLAGS, as defined in 6.2.1.3 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, as shown in Table 22.

Table 22 — 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

NOTE In the formulae below the following symbols are used, = …. is equal to, != … is not equal to, ! … Boolean not

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.

6.2.7.7.4 GROUP_SELECT_EQ

Command code = '00'

On receiving a GROUP_SELECT_EQ command, as shown in Table 23, a tag that is in the READY state shall read the 8-byte memory content beginning at the specified address and compare it with the WORD_DATA sent by the Interrogator. In the case that the memory content is equal to WORD_DATA the tag shall set its internal counter COUNT to 0, read its Tag ID and send back the Tag ID, as shown in Table 24, and go into the state ID.

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

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

Table 23 — GROUP_SELECT_EQ command

Preamble	Delimiter	Command	Address	Mask	WORD_DATA	CRC-16
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 24 — GROUP_SELECT_EQ response in the case of NO error

Preamble	ID	CRC-16
	64 bits	16 bits

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

6.2.7.7.5 GROUP_SELECT_NE

Command code = '01'

On receiving a GROUP_SELECT_NE command, as shown in Table 25, a tag that 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 Tag ID and send back the Tag ID, as shown in Table 26, and go into the state ID.

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

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

Table 25 — GROUP_SELECT_NE command

Preamble	Delimiter	Command	Address	Mask	WORD_DATA	CRC-16
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 26 — GROUP_SELECT_NE response in the case of NO error

Preamble	ID	CRC-16
	64 bits	16 bits

6.2.7.7.6 GROUP_SELECT_GT

Command code = '02'

On receiving a GROUP_SELECT_GT command, as shown in Table 27, a tag that 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 Tag ID and send back the Tag ID, as shown in Table 28, and go into the state ID.

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

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

Table 27 — GROUP_SELECT_GT command

Preamble	Delimiter	Command	Address	Mask	WORD_DATA	CRC-16
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 28 — GROUP_SELECT_GT response in the case of NO error

Preamble	ID	CRC-16
	64 bits	16 bits

6.2.7.7.7 GROUP_SELECT_LT

Command code = '03'

On receiving a GROUP_SELECT_LT command, as shown in Table 29, a tag that 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 Tag ID and send back the Tag ID, as shown in Table 30, and go into the state ID.

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

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

Table 29 — GROUP_SELECT_LT command

Preamble	Delimiter	Command	Address	Mask	WORD_DATA	CRC-16
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 30 — GROUP_SELECT_LT response in the case of NO error

Preamble	ID	CRC-16
	64 bits	16 bits

6.2.7.7.8 GROUP_UNSELECT_EQ

Command code = '04'

On receiving a GROUP_UNSELECT_EQ command, as shown in Table 31, a tag that 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 Tag ID and send back the Tag ID, as shown in Table 32.

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

Table 31 — GROUP_UNSELECT_EQ command

Preamble	Delimiter	Command	Address	Mask	WORD_DATA	CRC-16
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 32 — GROUP_UNSELECT_EQ response in the case of NO error and comparison fails

Preamble	ID	CRC-16
	64 bits	16 bits

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

6.2.7.7.9 GROUP_UNSELECT_NE

Command code = '05'

On receiving a GROUP_UNSELECT_NE command, as shown in Table 33, a tag that 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 Tag ID and send back the Tag ID as shown in Table 34.

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

Table 33 — GROUP_UNSELECT_NE command

Preamble	Delimiter	Command	Address	Mask	WORD_DATA	CRC-16
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 34 — GROUP_UNSELECT_NE response in the case of NO error and comparison fails

Preamble	ID	CRC-16
	64 bits	16 bits

6.2.7.7.10 GROUP_UNSELECT_GT

Command code = '06'

On receiving a GROUP_UNSELECT_GT command, as shown in Table 35, a tag that 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 Tag ID and send back the Tag ID, as shown in Table 36.

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

Table 35 — GROUP_UNSELECT_GT command

Preamble	Delimiter	Command	Address	Mask	WORD_DATA	CRC-16
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 36 — GROUP_UNSELECT_GT response in the case of NO error and comparison fails

Preamble	ID	CRC-16
	64 bits	16 bits

6.2.7.7.11 GROUP_UNSELECT_LT

Command code = '07'

On receiving a GROUP_UNSELECT_LT command, as shown in Table 37, a tag that 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 Tag ID and send back the Tag ID, as shown in Table 38.

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

Table 37 — GROUP_UNSELECT_LT command

Preamble	Delimiter	Command	Address	Mask	WORD_DATA	CRC-16
		8 bits	8 bits	8 bits	64 bits	16 bits

Table 38 — GROUP_UNSELECT_LT response in the case of NO error and comparison fails

Preamble	ID	CRC-16
	64 bits	16 bits

6.2.7.7.12 MULTIPLE_UNSELECT

Command code = '13'

On receiving a MULTIPLE_UNSELECT command, as shown in Table 39, a tag that 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 Tag ID and send back the Tag ID, as shown in Table 40.

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

Table 39 — MULTIPLE_UNSELECT command

Preamble	Delimiter	Command	ADDRESS	BYTE_DATA	CRC-16
		8 bits	8 bits	8 bits	16 bits

Table 40 — MULTIPLE_UNSELECT response in the case that WRITE_OK is reset or BYTE_DATA is not equal to memory content at ADDRESS

Preamble	ID	CRC-16
	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.

6.2.7.7.13 GROUP_SELECT_EQ_FLAGS

Command code = '17'

On receiving a GROUP_SELECT_EQ_FLAGS command, as shown in Table 41, a tag that 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 Tag ID and send back the Tag ID, as shown in Table 42, and go into the state ID.

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

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

Table 41 — GROUP_SELECT_EQ_FLAGS command

Preamble	Delimiter	Command	Mask	BYTE_DATA	CRC-16
		8 bits	8 bits	8 bits	16 bits

Table 42 — GROUP_SELECT_EQ_FLAGS response in the case of NO error

Preamble	ID	CRC-16
	64 bits	16 bits

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

6.2.7.7.14 GROUP_SELECT_NE_FLAGS

Command code = '18'

On receiving a GROUP_SELECT_NE_FLAGS command, as shown in Table 43, a tag that 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 Tag ID and send back the Tag ID, as shown in Table 44, and go into the state ID.

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

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

Table 43 — GROUP_SELECT_NE_FLAGS command

Preamble	Delimiter	Command	Mask	BYTE_DATA	CRC-16
		8 bits	8 bits	8 bits	16 bits

Table 44 — GROUP_SELECT_NE_FLAGS response in the case of NO error

Preamble	ID	CRC-16
	64 bits	16 bits

6.2.7.7.15 GROUP_UNSELECT_EQ_FLAGS

Command code = ‘19’

On receiving a GROUP_UNSELECT_EQ_FLAGS command, as shown in Table 45, a tag that 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 Tag ID and send back the Tag ID, as shown in Table 46.

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

Table 45 — GROUP_UNSELECT_EQ_FLAGS command

Preamble	Delimiter	Command	Mask	BYTE_DATA	CRC-16
		8 bits	8 bits	8 bits	16 bits

Table 46 — GROUP_UNSELECT_EQ_FLAGS response in the case of NO error and comparison fails

Preamble	ID	CRC-16
	64 bits	16 bits

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

6.2.7.7.16 GROUP_UNSELECT_NE_FLAGS

Command code = ‘1A’

On receiving a GROUP_UNSELECT_NE_FLAGS command, as shown in Table 47, a tag that 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 Tag ID and send back the Tag ID, as shown in Table 48.

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

Table 47 — GROUP_UNSELECT_NE_FLAGS command

Preamble	Delimiter	Command	Mask	BYTE_DATA	CRC-16
		8 bits	8 bits	8 bits	16 bits

Table 48 — GROUP_UNSELECT_NE_FLAGS response in the case of NO error and comparison fails

Preamble	ID	CRC-16
	64 bits	16 bits

6.2.7.8 Identification commands

6.2.7.8.1.1 Identification commands general

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

6.2.7.8.1.2 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, as shown in Table 49, 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 count is at FF, then the count remains unchanged for further FAIL commands.

If the resulting COUNT value is 0, then the tag shall read its Tag ID and sent back it in the response, as shown in Table 50.

Table 49 — FAIL command

Preamble	Delimiter	Command	CRC-16
		8 bits	16 bits

Table 50 — FAIL response in the case that COUNT stays zero

Preamble	ID	CRC-16
	64 bits	16 bits

6.2.7.8.1.3 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, and
- After a DATA_READ moves an identified tag to DATA_EXCHANGE, SUCCESS causes the next subset of selected but unidentified tags to transmit.

A tag shall only accept a SUCCESS command, as shown in Table 51, if it is in the ID state. In the case that the tag's internal counter COUNT is not zero, the internal counter shall be decreased by 1.

If the resulting COUNT value is 0, then the tag shall read its Tag ID and sent back it in the response, as shown in Table 52.

Table 51 — SUCCESS command

Preamble	Delimiter	Command	CRC-16
		8 bits	16 bits

Table 52 — SUCCESS response in the case that COUNT is zero

Preamble	ID	CRC-16
	64 bits	16 bits

6.2.7.8.1.4 RESEND

Command code = '15'

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

A tag shall only accept a RESEND command, as shown in Table 53, if it is in the ID state. If the COUNT value is 0, then the tag shall read its Tag ID and sent back it in the response, as shown in Table 54.

Table 53 — RESEND command

Preamble	Delimiter	Command	CRC-16
		8 bits	16 bits

Table 54 — RESEND response in the case that COUNT is zero

Preamble	ID	CRC-16
	64 bits	16 bits

6.2.7.8.1.5 INITIALIZE

Command code = '0A'

On receiving a INITIALIZE command, as shown in Table 55 a tag shall go into the READY state and reset the Data_Exchange_Status_Bit.

It shall not send any response.

Table 55 — INITIALIZE command

Preamble	Delimiter	Command	CRC-16
		8 bits	16 bits

6.2.7.8.1.6 FAIL_O

Command code = ‘40’ or ’41’

The identification algorithm uses FAIL_O 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_O command, as shown in Table 56, if it is in the ID state. In the case that the tag's internal counter COUNT is not zero, or the random generator result is 1, the internal counter shall be increased by 1, unless COUNT is FF.

If the resulting COUNT value is 0, then the tag shall read the memory at address ‘14’ sent back either 32 bit (for command code ‘40’) or 64 bit (for command code ‘41’) in the response, as shown in Table 57.

Table 56 — FAIL_O command

Preamble	Delimiter	Command	CRC-16
		8 bits	16 bits

Table 57 — FAIL_O response in the case that COUNT stays zero

Preamble	WORD_DATA	CRC-16
	32 or 64 bits	16 bits

6.2.7.8.1.7 SUCCESS_O

Command code = ‘42’ or ‘43’

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

- When all tags receiving FAIL_O backed off and did not transmit, SUCCESS_O causes those same tags to transmit again, and
- After a DATA_READ_O moves an identified tag to DATA_EXCHANGE, SUCCESS_O causes the next subset of selected but unidentified tags to transmit.

A tag shall only accept a SUCCESS_O command, as shown in Table 58 if it is in the ID state. In the case that the tag's internal counter COUNT is not zero it shall be decreased by 1.

If the resulting COUNT value is 0, then the tag shall read the memory at address ‘14’ sent back either 32 bit (for command code ‘40’) or 64 bit (for command code ‘41’) in the response, as shown in Table 59.

Table 58 — SUCCESS_O command

Preamble	Delimiter	Command	CRC-16
		8 bits	16 bits

Table 59 — SUCCESS_O response in the case that COUNT is zero

Preamble	WORD_DATA	CRC-16
	32 or 64 bits	16 bits

6.2.7.8.1.8 RESEND_O

Command code = '46' or '47'

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

A tag shall only accept a RESEND_O command, as shown in Table 60, if it is in the ID state. If the COUNT value is 0, then the tag shall read the memory at address '10' sent back either 32 bit (for command code '46') or 64 bit (for command code '47') in the response, as shown in Table 61.

Table 60 — RESEND_O command

Preamble	Delimiter	Command	CRC-16
		8 bits	16 bits

Table 61 — RESEND_O response in the case that COUNT is zero

Preamble	ID	CRC-16
	32 or 64 bits	16 bits

6.2.7.9 Data transfer commands

6.2.7.9.1 Data transfer commands general

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

6.2.7.9.2 READ

Command code = '0C'

On receiving the READ command, as shown in Table 62, the tag shall compare the sent ID with its Tag ID. In the case that the ID is equal to the Tag ID, 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, as shown in Table 63. Further, the tag shall mark the byte at ADDRESS lockable. In the case that ID is not equal to Tag ID or any other error the tag shall not send a reply.

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

Table 62 — Read command

Preamble	Delimiter	Command	ID	Address	CRC-16
		8 bits	64 bits	8 bits	16 bits

Table 63 — Read response in the case of NO error

Preamble	WORD_DATA	CRC-16
	64 bits	16 bits

6.2.7.9.3 DATA_READ

Command code = ‘0B’

On receiving the DATA_READ command, as shown in Table 64, the tag shall only if it is in either the state ID or the state DATA_EXCHANGE compare the sent ID with its Tag ID. In the case that the ID is equal to the Tag ID, 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, as shown in Table 65. Further, the tag shall mark the byte at ADDRESS lockable. In the case that the tag is not in the state READY, or ID is not equal to Tag ID or any other error the tag shall not send a reply.

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

Table 64 — DATA_READ command

Preamble	Delimiter	Command	ID	Address	CRC-16
		8 bits	64 bits	8 bits	16 bits

Table 65 — DATA_READ response in the case of NO error

Preamble	WORD_DATA	CRC-16
	64 bits	16 bits

6.2.7.9.4 DATA_READ_O

Command code = ‘44’, or ‘45’

On receiving the DATA_READ_O command, as shown in Table 66, the tag shall compare the sent DATA with its memory contents starting at address ‘14’ if the tag is in either the ID state or the DATA_EXCHANGE state. The amount of data compared shall be determined by the command code. If the received command code is '44', the tag shall logically compare the 32 bits. If command code is '45', the tag shall logically compare 64 bits. If the data strings (32 or 64 bits) are equal, then the tag shall transition from any state except READY to the DATA_EXCHANGE state, read the 8 byte memory content beginning at the specified address and send back its contents in the response, as shown in Table 67. Further, the tag shall mark the byte at ADDRESS lockable. If the tag is in the READY state, or the data strings are not equal, or any other error the tag shall not send a reply.

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

Table 66 — DATA_READ_O command

Preamble	Delimiter	Command	DATA	Address	CRC-16
		8 bits	32 or 64 bits	8 bits	16 bits

Table 67 — DATA_READ_O response in the case of NO error

Preamble	WORD_DATA	CRC-16
	64 bits	16 bits

6.2.7.9.5 READ_FLAGS

Command code = ‘50’

On receiving the READ_FLAGS command, as shown in Table 68, the tag shall from any state move to the DATA_EXCHANGE state, read the FLAGS and send back its content in the response, as shown in Table 69.

Table 68 — READ_FLAGS command

Preamble	Delimiter	Command	ID	CRC-16
		8 bits	64 bits	16 bits

Table 69 — READ_FLAGS in the case of NO error

Preamble	BYTE_DATA	CRC-16
	8 bits	16 bits

6.2.7.9.6 READ_VARIABLE

Command code = ‘51’

On receiving the READ_VARIABLE command, as shown in Table 70, the tag shall compare the sent ID with its Tag ID. In the case that the ID is equal to the Tag ID, the tag shall from any state move to the DATA_EXCHANGE state, read the specified length of the memory content beginning at the specified address and send back its content in the response, as shown in Table 71. In the case that ID is not equal to Tag ID or any other error the tag shall not send a reply.

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

Table 70 — READ_VARIABLE command

Preamble	Delimiter	Command	ID	Address	Length	CRC-16
		8 bits	64 bits	8 bits	8 bits	16 bits

Table 71 — READ_VARIABLE response in the case of NO error

Preamble	(Length + 1) * BYTE_DATA	CRC-16
	(Length + 1) * 8 bits	16 bits

6.2.7.9.7 READ_PORT

Command code = ‘52’

On receiving the READ_PORT command, as shown in Table 72, the tag shall compare the sent ID with its Tag ID. In the case that the ID is equal to the Tag ID, the tag shall from any state move to the DATA_EXCHANGE, read the 8 bit memory content beginning at the specified port address and send back the memory content in the response, as shown in Table 73. In the case that ID is not equal to Tag ID or any other error the tag shall not send a reply.

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

Table 72 — READ_PORT command

Preamble	Delimiter	Command	ID	Address	CRC-16
		8 bits	64 bits	8 bits	16 bits

Table 73 — READ_PORT response in the case of NO error

Preamble	BYTE_DATA	CRC-16
	8 bits	16 bits

Selecting port 0 shall select the flags as specified in 6.2.1.3, and Table 12.

Ports 1, 2, … to ‘FF’ are reserved for future use.

6.2.7.9.8 READ_UNADDRESSED

Command code = ‘53’

On receiving the READ_UNADDRESSED command, as shown in Table 74, the tag shall from any state move to the DATA_EXCHANGE state, read the 16-byte memory content beginning at the specified address and send back its content in the response, as shown in Table 75.

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

Table 74 — READ_UNADDRESSED command

Preamble	Delimiter	Command	Address	CRC-16
		8 bits	8 bits	16 bits

Table 75 — READ_UNADDRESSED response in the case of NO error

Preamble	WORD_DATA	WORD_DATA	CRC-16
	64 bits	64 bits	16 bits

6.2.7.9.9 READ_VERIFY

Command code = '12'

On receiving the READ_VERIFY command, as shown in Table 76 the tag shall compare the sent ID with its Tag ID. In the case that the ID is equal to the Tag ID 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, as shown in Table 77. Further, the tag shall mark the byte at ADDRESS lockable. In the case that ID is not equal to Tag ID, 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 76 — READ_VERIFY command

Preamble	Delimiter	Command	ID	Address	CRC-16
		8 bits	64 bits	8 bits	16 bits

Table 77 — READ_VERIFY response in the case of NO error

Preamble	BYTE_DATA	CRC-16
	8 bits	16 bits

6.2.7.9.10 READ_VERIFY_4BYTE

Command code = '1D'

On receiving the READ_VERIFY_4BYTE command, as shown in Table 78, the tag shall compare the sent ID with its Tag ID. In the case that the ID is equal to the Tag ID 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, as shown in Table 79. Further, the tag shall mark the byte at ADDRESS lockable. Bytes at ADDRESS+1, ADDRESS+2 and ADDRESS+3 shall not be marked lockable.

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

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

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

Table 78 — READ_VERIFY_4BYTE command

Preamble	Delimiter	Command	ID	Address	BYTE_MASK	CRC-16
		8 bits	64 bits	8 bits	8 bits	16 bits

Table 79 — READ_VERIFY_4BYTE response in the case of NO error

Preamble	BYTE_DATA	CRC-16
	8 bits	16 bits

6.2.7.9.11 WRITE

Command code = ‘0D’

On receiving the WRITE command, as shown in Table 80, the tag shall compare the sent ID with its Tag ID. In the case that the ID is equal to the Tag ID, 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, as shown in Table 82. Further, the tag shall mark the byte at ADDRESS lockable. In the case that the memory is unlocked, it shall send back the *ACK*NOWLEDGE, as shown in Table 81, and program the data into 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 80 — Write command

Preamble	Delimiter	Command	ID	Address	BYTE_ DATA	CRC-16
		8 bits	64 bits	8 bits	8 bits	16 bits

Table 81 — WRITE response in the case of NO error

Preamble	ACKNOWLEDGE	CRC-16
	8 bits	16 bits

Table 82 — WRITE response in the case of locked memory

Preamble	ERROR	CRC-16
	8 bits	16 bits

6.2.7.9.12 WRITE4BYTE

Command code = ‘1B’

On receiving the WRITE4BYTE command, as shown in Table 83, the tag shall compare the sent ID with its Tag ID. In the case that the ID is equal to the Tag ID, 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 memory bytes is locked, it shall send back the ERROR response, as shown in Table 85. In the case that all bytes are unlocked, it shall send back the *ACK*NOWLEDGE, as shown in Table 84, and program the data into the specified memory. In all other cases the tag shall not send a 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).

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

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

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 83 — WRITE4BYTE command

Preamble	Delimiter	Command	ID	Address	Byte_Mask	Data	CRC-16
		8 bits	64 bits	8 bits	8 bits	32 bits	16 bits

Table 84 — WRITE4BYTE response in the case of NO error

Preamble	ACKNOWLEDGE	CRC-16
	8 bits	16 bits