

BLUEBOX RFID System



COMMUNICATION PROTOCOL



Profibus

Preface

iDTRONIC GmbH (IDTRONIC) reserves the right to make changes to its products or services or to discontinue any product or service at any time without notice. IDTRONIC provides customer assistance in various technical areas, but, does not have full access to data concerning the use and applications of customer's products. Therefore, IDTRONIC assumes no liability and is not responsible for customer applications or product or software design or performance relating to systems or applications incorporating IDTRONIC products. In addition, IDTRONIC assumes no liability and is not responsible for infringement of patents and/or any other intellectual or industrial property rights of third parties, which may result from assistance provided by IDTRONIC. IDTRONIC products are not designed, intended, authorized or warranted to be suitable for life support applications or any other life critical applications that could involve potential risk of death, personal injury or severe property or environmental damage. With the edition of this document, all previous editions become void. Indications made in this manual may be changed without previous notice. Composition of the information in this manual has been done to the best of our knowledge. IDTRONIC does not guarantee the correctness and completeness of the details given in this manual and may not be held liable for damages ensuing from incorrect or incomplete information. Since, despite all our efforts, errors may not be completely avoided, we are always grateful for your useful tips. The installation instructions given in this manual are based on advantageous boundary conditions. IDTRONIC does not give any guarantee promise for perfect function in cross environments. The companies or products mentioned in this document might be brands or brand names of the different suppliers or their subsidiaries in any country. This document may be downloaded onto a computer, stored and duplicated as necessary to support the use of the related IDTRONIC products. Any other type of duplication, circulation or storage on data carriers in any manner not authorized by IDTRONIC represents a violation of the applicable copyright laws and shall be prosecuted.

iDTRONIC GmbH
Ludwig-Reichling-Straße 4
67059 Ludwigshafen
Germany/Deutschland

Issue 1.06
– 15. January 2021 –

Phone: +49 621 6690094-0
Fax: +49 621 6690094-9
E-Mail: info@idtronic.de
Web: idtronic.de

Subject to alteration without prior notice.
© Copyright iDTRONIC GmbH 2021
Printed in Germany

Safety Instructions / Warning - Read before start-up!

- The device may only be used for the intended purpose designed by the manufacturer. The operation manual should be conveniently kept available at all times for each user.
- Unauthorized changes and the use of spare parts and additional devices that have not been sold or recommended by the manufacturer may cause fire, electric shocks or injuries. Such unauthorized measures shall exclude any liability by the manufacturer.
- The liability-prescriptions of the manufacturer in the issue valid at the time of purchase are valid for the device. The manufacturer shall not be held legally responsible for inaccuracies, errors, or omissions in the manual or automatically set parameters for a device or for an incorrect application of a device.
- Repairs may be executed by the manufacturer only.
- Only qualified personnel should carry out installation, operation, and maintenance procedures.
- Use of the device and its installation must be in accordance with national legal requirements and local electrical codes.
- When working on devices the valid safety regulations must be observed.

This manual applies to the following devices:

Description:	Order Number:
Read / write LF RFID device with integrated antenna. Profibus communication interface.	5223L
Read / write LF RFID device with one external antenna. Profibus communication interface.	5233L
Read / write LF RFID device with two external antennas. Profibus communication interface.	5243L
Read / write HF RFID device with integrated antenna. Profibus communication interface.	5223H
Read / write HF RFID device with one external antenna. Profibus communication interface.	5233H
Read / write HF RFID device with two external antennas. Profibus communication interface.	5243H
Read / write UHF RFID device with integrated antenna. Profibus communication interface. EU1 (865 MHz ... 868 MHz) version.	5223U-S



Description:

Read / write UHF RFID device with one external antenna. Profibus communication interface. EU1 (865 MHz ... 868 MHz) version.

Read / write UHF RFID device with one external antenna. Profibus communication interface. EU1 (865 MHz ... 868 MHz) version.

Read / write UHF RFID device with up to four external antennas. ProfiBus communication interface. EU1 (865 MHz ... 868 MHz) version.

Order Number:

5239U-S

5239U-C

5233U

This manual is valid as of firmware version:

Order Number	Carrier	Front End
5223L	2.01	3.17d
5233L	2.01	3.17d
5243L	2.01	3.17d
5223H	2.10	1.23
5233H	2.10	1.23
5243H	2.10	1.23
5223U-S	3.10	1.35M
5239U-S	3.10	1.35M
5239U-C	3.10c	1.35M
5233U	2.10	1.35Q

Table of Contents

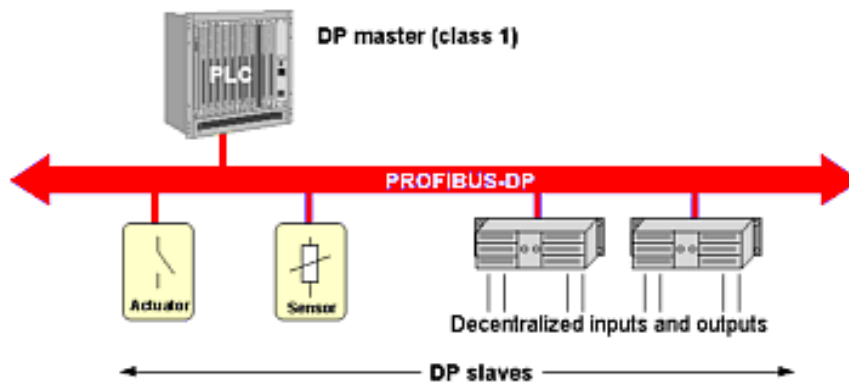
1	Introduction.....	7
2	Communication Features	9
2.1	Device Reset	12
2.2	Read Serial Number	12
2.3	Read Firmware Version	13
2.4	Read Temperature	14
2.5	Read Date/Time.....	15
2.6	Write Date/Time	16
2.7	Write General Parameters	16
2.8	Write Configuration Parameters	17
2.9	Set Default Parameters.....	18
2.10	Read General Parameters.....	18
2.11	Read Configuration Parameters.....	19
2.12	'RF Reading' Test	20
2.13	'RF Power' Test	20
2.14	'RF Sensitivity' Test	21
2.15	Read Reflected Power	23
2.16	Read RSSI Power	24
2.17	Digital Output Activation	25
2.18	Read Device Status	25
2.19	RF Deactivation.....	26
2.20	RF Activation	26
2.21	Antennas Auto-Tuning	26
2.22	Buffer Data Request	26
2.23	Queue Data Request.....	32
2.24	Write Data to a EM4305 Transponder	36
2.25	Read ID Code of a EM4305 Transponder	37
2.26	Write Data to a T5557 Transponder	38
2.27	Read ID Code of a T5557 Transponder.....	39
2.28	Write Data to a Q5 Transponder.....	40
2.29	Read ID Code of a Q5 Transponder.....	41
2.30	Write Data to a HITAG S Transponder.....	42
2.31	Read ID Code of a HITAG 1 /HITAG S Transponder.....	43
2.32	Read a Page of a HITAG 1 / HITAG S Transponder.....	44
2.33	Write a Page of a HITAG 1 / HITAG S Transponder	46
2.34	Read ID Code of a HITAG 2 Transponder.....	47
2.35	Read a Page of a HITAG 2 Transponder.....	48
2.36	Write a Page of a HITAG 2 Transponder	50
2.37	'Reset' Command for TITAN Transponder	51
2.38	'Login' Command for TITAN Transponder.....	52
2.39	'Write Password' Command for TITAN Transponder	53
2.40	'Standard Read' Command for TITAN Transponder	55
2.41	'Selective Read' Command for TITAN Transponder	56
2.42	'Write Word' Command for TITAN Transponder	58

2.43	'Write Several Words' Command for TITAN Transponder	59
2.44	'Read After Write Word' Command for TITAN Transponder	60
2.45	ISO 15963 Transponders 'Inventory' Command	62
2.46	Read a Data Block of an ISO 15693 Transponder	63
2.47	Write a Data Block of an ISO 15693 Transponder	64
2.48	Lock a Data Block of an ISO 15693 Transponder	65
2.49	ISO 15963 Transponder 'Get System Info' Command	67
2.50	ISO 15963 Transponder 'General Protocol' Command	68
2.51	ISO 14443A Transponder 'Inventory' Command	70
2.52	Read a Data Block of a MIFARE 1k/4k (UID 4) Transponder	71
2.53	Write a Data Block of a MIFARE 1k/4k (UID 4) Transponder	73
2.54	Read a Data Block of a MIFARE 1k/4k (UID 7) Transponder	74
2.55	Write a Data Block of a MIFARE 1k/4k (UID 7) Transponder	76
2.56	Read a Data Page of a MIFARE Ultralight transponder	77
2.57	Write a Data Page of a MIFARE Ultralight Transponder	78
2.58	Read a Data Page of a NTAG213/215/216 transponder	80
2.59	Write a Data Page of a NTAG213/215/216 Transponder	81
2.60	ISO 14443A-4 Transponder 'RATS' Command	82
2.61	ISO 14443A-4 Transponder 'Generic Command'	84
2.62	ISO 14443B Transponder 'Inventory' Command	85
2.63	Read a Data Block of a SR 176 Transponder	86
2.64	Write a data block of a SR 176 transponder	87
2.65	PicoPass Transponders 'Inventory' Command	89
2.66	ISO 18000-63 Transponder 'Inventory' Command	90
2.67	Program EPC of an ISO 18000-63 Transponder	93
2.68	Read Data of an ISO 18000-63 Transponder	95
2.69	Write Data of an ISO 18000-63 Transponder	96
2.70	Lock Data of an ISO 18000-63 Transponder	99
2.71	'Kill' Command of an ISO 18000-63 Transponder	101
2.72	'QT Read' Command of a Monza 4QT Transponder	102
2.73	'QT Write' Command of a Monza 4QT Transponder	104
2.74	'Read Sensor Code' Command of a Magnus Sx Transponder	105
2.75	'Read On-Chip RSSI' Command of a Magnus Sx Transponder	106
2.76	'Read Temperature Code' Command of a Magnus S3 Transponder	108
2.77	'Spontaneous' Message	109
3	Examples	111
3.1	Read Firmware Version	111
3.2	Buffer Data Request	113
3.3	Queue Data Request	136
4	Document Revision History	147
A.	Supported Commands Table	149

1 Introduction

This document describes the message format of the ProfiBus communication protocol used by the host and the reader in order to issuing commands and reply with responses.

PROFIBUS DP (Dezentrale Pheripherie) was developed for high-speed communication between central controllers (typically PLC) and remote devices (I/O, drives, actuators, sensors, ...).

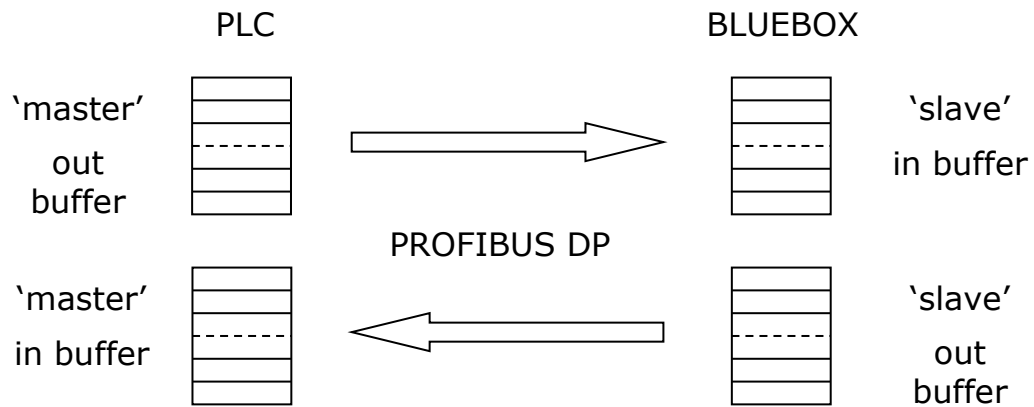


More devices can be connected on the bus, each one has a different address (the address is a configurable parameter). Once configured, the communication with devices is cyclic.

Therefore the **BLUEBOX** is a remote device ('slave') with an input buffer and an output buffer (the length of the buffers is a configurable parameter). Similarly, in order to talk to the **BLUEBOX**, the PLC ('master') has an output buffer (same size of the corresponding input buffer of the **BLUEBOX**) and an input buffer (same size of the corresponding output buffer of the **BLUEBOX**). The input buffer of the **BLUEBOX** can only be modified by the 'master' (PLC) while the output buffer of the **BLUEBOX** can only be modified by the 'slave' that is the **BLUEBOX**. The aim of the cyclic communication is to keep up to date the corresponding buffers at the 'master' side and at the 'slave' side.

At the application level, a specific protocol has been defined for enabling the delivery of control messages from the PLC to the **BLUEBOX** and reply messages from the **BLUEBOX** to the PLC.

For the developers: a .GSD (Generic Station Description) is provided that defines the characteristics of the device and is necessary for the configuration of the ProfiBus network.

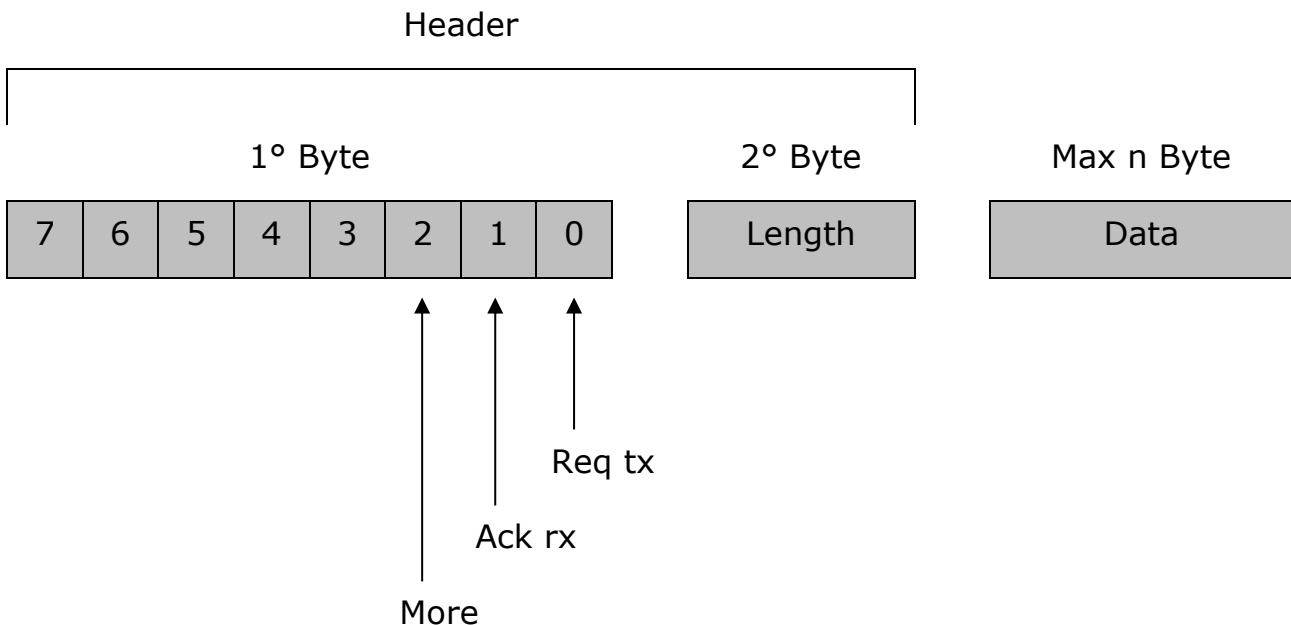


2 Communication Features

There are two types of data packets used in the communication:

- Outgoing data packets from the 'master' (PLC) to the 'slave' in order to send a command or an answer to the **BLUEBOX**
- Incoming data packets to the 'master' (PLC) from the 'slave' (**BLUEBOX**) carrying for example the answer to a command

The outgoing data packets from the 'master' assume the following structure:



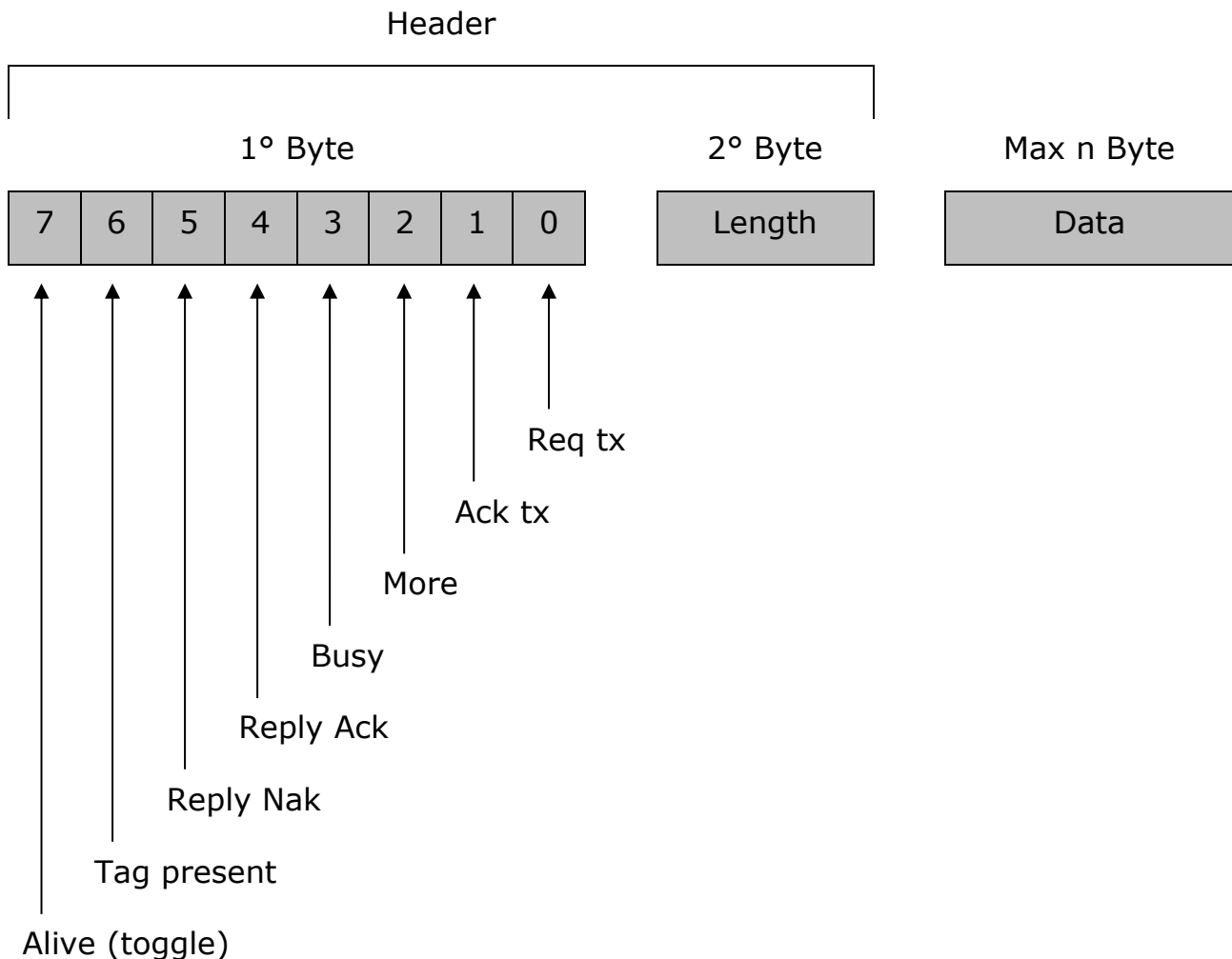
The outgoing data packet consists of a header (first two bytes) and a data buffer of n bytes.

The first byte of the header consists of the following flags:

- Bit 7 ... 3: Not used;
- Bit 2: More, set to '1' means that the message is composed of several data packets;
- Bit 1: Ack rx, reception acknowledge;
- Bit 0: Req tx, transmission request.

The second byte (Length) of the header specifies the number of data bytes in the data buffer.

The incoming data packets to the 'master' assume the following structure:



Also the incoming data packet consists of a header (first two bytes) and a data buffer of n bytes.

The first byte of the header consists of the following flags:

- Bit 7: Alive, toggles every second and means that the 'slave' is running correctly;
- Bit 6: Tag present, set to '1' by the 'slave' means that a transponder is present near the antenna (detected by the 'continuous' identification activity);
- Bit 5: Reply Nak, set to '1' by the 'slave' means that an error has occurred;
- Bit 4: Reply Ack, set to '1' by the 'slave' means that the received command has been processed;

- Bit 3: Busy, set to '1' by the 'slave' means that the 'slave' is processing the command message from the 'master';
- Bit 2: More, set to '1' by the 'slave' means that the message is composed of several data packets;
- Bit 1: Req tx, reception request (from 'slave');
- Bit 0: Ack tx, transmission acknowledge.

The second byte (Length) of the header specifies the number of data bytes in the data buffer.

The messages consist of one or more data packets. If the length of the message is shorter than n bytes, the message will be composed of only 1 data packet. If the length of the message is bigger than n bytes, the message will be composed of more than 1 data packet; in this case the header of all the transmitted data packets, apart the last one, will present at '1' the flag 'More' indicating that the message is not completed and another data packet will follow.

The communication between 'master' and 'slave' for a command message take place with the following handshake:

1. The 'master' loads the buffer with the command message and subsequently sets to '1' the flag 'Req tx' to inform the 'slave' that a data packet is ready to be acquired
2. The 'slave' acquires the data packet from the 'master' and confirm the completion of the operation by setting to '1' the flag 'Ack tx'
3. After having received the acknowledgment of the completion of the operation through the flag 'Ack tx' at '1', the 'master' resets to '0' the flag 'Req tx'
4. After having verified that the flag 'Req tx' is reset to '0', also the 'slave' resets to '0' the flag 'Ack tx'
5. During the execution time of the received command, the 'slave' sets to '1' the flag 'Busy' to inform the 'master' that it is not temporarily not available for further communication

In the case of a message length that needs more than one data packet, the previous handshake will be repeated for every data packet until the end of the message.

The answer of the 'slave' to a command message from the 'master' can take place through a full answer message or in a short form depending of the type of command. In the case of a short form answer, it take place through the setting

to '1' of the flag 'Reply ack' or the flag 'Reply nak' (in function of the result of the execution of the command).

The communication between 'slave' and 'master' for an answer message take place with the following handshake:

1. The 'slave' loads the buffer with the answer message and subsequently sets to '1' the flag 'Req rx' to inform the 'master' that a data packet is ready to be acquired
2. The 'master' acquires the data packet from the 'slave' and confirm the completion of the operation by setting to '1' the flag 'Ack rx'
3. After having received the acknowledgment of the completion of the operation through the flag 'Ack rx' at '1', the 'slave' resets to '0' the flag 'Req rx'
4. After having verified that the flag 'Req rx' is reset to '0', also the 'master' resets to '0' the flag 'Ack rx'

In the case of a message length that needs more than one data packet, the previous handshake will be repeated for every data packet until the end of the message.

2.1 Device Reset

This command is used to restart the **BLUEBOX** (the device has the same behavior like when it is powered up).

Byte	Value	Description	Notes
1	0x30	Command code	

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command don't 'produce' data).

2.2 Read Serial Number

This command is used to get the SN code of the **BLUEBOX** (unique for each device and assigned during the production process), the SN is constituted by 6 bytes.

Byte	Value	Description	Notes
1	0x2A	Command code	
2	0x01		

If the command fails, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

Byte	Value	Description	Notes
1	0x2A	Command code	
2	0x01		
2+1	0x..	Serial number, 1st byte	The SN is 6 bytes long
2+i	0x..	Serial number, i th byte	i < 6
2+6	0x..	Serial number, 6th byte	i = 6



The SN is a numeric code constituted by 12 digits, the bytes of the SN are BCD-coded and so every byte encodes 2 digits.

2.3 Read Firmware Version

This command used to get the firmware version loaded on the **BLUEBOX**.

Byte	Value	Description	Notes
1	0x34	Command code	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

Byte	Value	Description	Notes
1	0x34	Command code	
1+1	0x..	Firmware version, 1st ASCII char	The fw version is 16 chars long
1+i	0x..	Firmware version, i th ASCII char	i < 16
1+16	0x..	Firmware version, 16th ASCII char	i = 16

The 16 bytes (2 to 17) are a string of 16 ASCII chars that defines the firmware version.

It is also possible to get the firmware version of the reader module/s mounted in the **BLUEBOX**.

Byte	Value	Description	Notes
1	0x34	Command code	
2	0x..	Module number (0x01 or 0x02)	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

Byte	Value	Description	Notes
1	0x34	Command code	
2	0x..	Module number	0x01 or 0x02
2+1	0x..	Firmware version, 1st ASCII char	The fw version is 64 chars long
2+i	0x..	Firmware version, i th ASCII char	i < 64
2+64	0x..	Firmware version, 64th ASCII char	i = 64

The 64 bytes (3 to 66) are a string of 64 ASCII chars that defines the firmware version.

2.4 Read Temperature

This command sends back the internal temperature of the **BLUEBOX** measured by the on board temperature sensor.

Byte	Value	Description	Notes
1	0x3A	Command code	

If the command fails, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

Byte	Value	Description	Notes
1	0x3A	Command code	
2	0x..	Integer value of the temperature in °C	

Byte	Value	Description	Notes
3	0x..	Fractional value of the temperature. Bits 7, 6, 5 encode the fractional value in steps of 0.125 °C: <ul style="list-style-type: none"> • 00000000b → .000 °C • 00100000b → .125 °C • ... • 11100000b → .875°C 	



28h, E0h -> 40.875 °C

2.5 Read Date/Time

This command sends back the date/time of the **BLUEBOX** available on the internal real time clock device.

Byte	Value	Description	Notes
1	0x28	Command code	

If the command fails, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

Byte	Value	Description	Notes
1	0x28	Command code	
2	0x..	Year value thousands and hundreds. BCD encoded byte.	
3	0x..	Year value tens and units. BCD encoded byte.	
4	0x..	Month value tens and units. BCD encoded byte.	
5	0x..	Day value tens and units. BCD encoded byte.	
6	0x..	Hour value tens and units. BCD encoded byte.	

Byte	Value	Description	Notes
7	0x..	Minute value tens and units. BCD encoded byte.	
8	0x..	Second value tens and units. BCD encoded byte.	

2.6 Write Date/Time

This command is used to set the date/time of the **BLUEBOX** in the internal real time clock device.

Byte	Value	Description	Notes
1	0x29	Command code	
2	0x..	Year value thousands and hundreds. BCD encoded byte.	
3	0x..	Year value tens and units. BCD encoded byte.	
4	0x..	Month value tens and units. BCD encoded byte.	
5	0x..	Day value tens and units. BCD encoded byte.	
6	0x..	Hour value tens and units. BCD encoded byte.	
7	0x..	Minute value tens and units. BCD encoded byte.	
8	0x..	Second value tens and units. BCD encoded byte.	

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command don't 'produce' data).

2.7 Write General Parameters

This command is used to set the operating parameters of the **BLUEBOX**.

Byte	Value	Description	Notes
1	0x2F	Command code	

Byte	Value	Description	Notes
2	0x..	General parameter byte 1.	See the device user manual for details.
3	0x..	General parameter byte 2.	See the device user manual for details.
4	0x..	General parameter byte 3.	See the device user manual for details.
5	0x..	General parameter byte 4.	See the device user manual for details.
6	0x..	General parameter byte 5.	See the device user manual for details.
7	0x..	General parameter byte 6.	See the device user manual for details.
8	0x..	General parameter byte 7.	See the device user manual for details.

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command don't 'produce' data).



After the command execution, the **BLUEBOX** resets itself to apply the new parameters.

2.8 Write Configuration Parameters

This command is used to set the configuration parameters of the **BLUEBOX**.

Byte	Value	Description	Notes
1	0x3D	Command code	
2	0x..	The configuration page (0x00 ... 0x0F, 0x80 ... 0x87, 0xC0 ... 0xCF).	
2+1	0x..	Configuration parameter byte 1.	See the device user manual for details.
2+i	0x..	Configuration parameter byte i.	See the device user manual for details.

Byte	Value	Description	Notes
2+n	0x..	Configuration parameter byte n. <ul style="list-style-type: none"> n=7 if configuration page is 0x00 ... 0x0F n=14 if configuration page is 0x80 ... 0x87 n=nvar(240) if configuration page is 0xC0 ... 0xCF 	See the device user manual for details.

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command don't 'produce' data).

2.9 Set Default Parameters

This command is used to set the default values of the communication and operating parameters of the **BLUEBOX**.

Byte	Value	Description	Notes
1	0x31	Command code	

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command don't 'produce' data).



After the command execution, the **BLUEBOX** resets itself to apply the new parameters.

2.10 Read General Parameters

This command is used to get the values of the operating parameters of the **BLUEBOX**.

Byte	Value	Description	Notes
1	0x2A	Command code	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

Byte	Value	Description	Notes
1	0x2A	Command code	
2	0x..	General parameter byte 1.	See the device user manual for details.
3	0x..	General parameter byte 2.	See the device user manual for details.
4	0x..	General parameter byte 3.	See the device user manual for details.
5	0x..	General parameter byte 4.	See the device user manual for details.
6	0x..	General parameter byte 5.	See the device user manual for details.
7	0x..	General parameter byte 6.	See the device user manual for details.
8	0x..	General parameter byte 7.	See the device user manual for details.

2.11 Read Configuration Parameters

This command is used to get the values of the configuration parameters of the **BLUEBOX**.

Byte	Value	Description	Notes
1	0x3C	Command code	
2	0x..	The configuration page (0x00 ... 0x0F, 0x80 ... 0x87, 0xC0 ... 0xCF).	

If the command fails, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

Byte	Value	Description	Notes
1	0x3C	Command code	
1+1	0x..	Configuration parameter byte 1.	See the device user manual for details.
1+i	0x..	Configuration parameter byte i.	See the device user manual for details.

Byte	Value	Description	Notes
1+n	0x..	Configuration parameter byte n. <ul style="list-style-type: none"> n=7 if configuration page is 0x00 ... 0x0F n=14 if configuration page is 0x80 ... 0x87 n=nvar(240) if configuration page is 0xC0 ... 0xCF 	See the device user manual for details.

2.12 'RF Reading' Test

In 'continuous' mode, this command is used to activate/deactivate the 'reading test' mode.

Byte	Value	Description	Notes
1	0xD7	Command code	
2	0x..	To activate/deactivate the 'RF reading' test mode: <ul style="list-style-type: none"> 0x00: To deactivate 'RF reading' test mode; 0x01: To activate 'RF reading' test mode. 	

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command don't 'produce' data).



The 'RF reading' test mode setting is stored in EEPROM and resumed at every restart of the **BLUEBOX**.

2.13 'RF Power' Test

This command is used to easily and quickly test the minimum RF output power needed to read a tag in a fixed position. The reader sweeps from the minimum RF output power to maximum RF output power or until it finds a tag, increasing the RF power of 1 dB every 500ms with fixed Q selection algorithm and Q=0.

Byte	Value	Description	Notes
1	0xDA	Command code	

Byte	Value	Description	Notes
2	0x..	Antenna to use for test: <ul style="list-style-type: none"> • 0x01 -> Antenna 1. • 0x02 -> Antenna 2. • 0x03 -> Antenna 3. • 0x04 -> Antenna 4. 	
3	0x..	RF channel to use for test: <ul style="list-style-type: none"> • 0x01 ... 0x0A if ETSI region is selected; • 0x01 ... 0x32 if FCC region is selected. 	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

a) if a tag has been identified

Byte	Value	Description	Notes
1	0xDA	Command code	
2	0x00	Status Ok	
3	0x..	Minimum RF output power needed to read the tag.	

b) if no tag has been found

Byte	Value	Description	Notes
1	0xDA	Command code	
2	0x01	Status transponder not present	

2.14 'RF Sensitivity' Test

This command is used to easily and quickly test the minimum RF input sensitivity needed to read a tag in a fixed position. The reader sweeps from the minimum RF input sensitivity to maximum RF input sensitivity or until it finds a tag, increasing the RF sensitivity of 1 dB every 500ms with fixed Q selection algorithm and Q=0.

Byte	Value	Description	Notes
1	0xDB	Command code	

Byte	Value	Description	Notes
2	0x..	Antenna to use for test: <ul style="list-style-type: none"> • 0x01 -> Antenna 1. • 0x02 -> Antenna 2. • 0x03 -> Antenna 3. • 0x04 -> Antenna 4. 	
3	0x..	RF channel to use for test: <ul style="list-style-type: none"> • 0x01 ... 0x0A if ETSI region is selected. • 0x01 ... 0x32 if FCC region is selected. 	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

a) if a tag has been identified

Byte	Value	Description	Notes
1	0xDB	Command code	
2	0x00	Status success	
3	0x..	Minimum RF input sensitivity needed to read the tag	

b) if no tag has been found

Byte	Value	Description	Notes
1	0xDB	Command code	
2	0x01	Status transponder not present	

2.15 Read Reflected Power

This command is used to read the approximation of the antenna reflected power to easily check the antenna connection.

Byte	Value	Description	Notes
1	0xFE	Command code	
2	0x..	Antenna to use for test: <ul style="list-style-type: none"> • 0x01 -> Antenna 1. • 0x02 -> Antenna 2. • 0x03 -> Antenna 3. • 0x04 -> Antenna 4. 	
3	0x..	The frequency to test in MHz in the range 840 ... 960 MHz. MSB.	The frequency is 3 bytes length
4	0x..		
5	0x..	The frequency to test in MHz in the range 840 ... 960 MHz. LSB.	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

Byte	Value	Description	Notes
1	0xFE	Command code	
2	0x00	Status success	
3	0x..	The I-channel RSSI value	
4	0x..	The Q-channel RSSI value	
5	0x..	The G value used to calculate the reflected power as defined below	

And the reflected power is calculated as follows

$$mixDC = \sqrt{Ich^2 + Qch^2}$$

$$Pin(dBm) = 20 \log \left(\frac{mixDC}{G} \right)$$

2.16 Read RSSI Power

This command is used to read the approximation of the RF signal strength received by the antenna to easily check the presence or not of external RF sources.

Byte	Value	Description	Notes
1	0xFD	Command code	
2	0x..	Antenna to use for test: <ul style="list-style-type: none"> • 0x01 -> Antenna 1. • 0x02 -> Antenna 2. • 0x03 -> Antenna 3. • 0x04 -> Antenna 4. 	
3	0x..	The frequency to test in MHz in the range 840 ... 960 MHz. MSB.	The frequency is 3 bytes length
4	0x..		
5	0x..	The frequency to test in MHz in the range 840 ... 960 MHz. LSB.	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

Byte	Value	Description	Notes
1	0xFD	Command code	
2	0x00	Status success	
3	0x..	The I-channel RSSI value	
4	0x..	The Q-channel RSSI value	
5	0x..	The G value used to calculate the RSSI power as defined below	

And the RSSI power is calculated as follows

$$meanRSSI = \frac{Ich + Qch}{2}$$

$$Pin(dBm) = 2.1 * meanRSSI - G$$

2.17 Digital Output Activation

This command is used to activate each individual output and also to set the duration in case of impulsive use.

Byte	Value	Description	Notes
1	0x37	Command code	
2	0x..	Antenna to use for test: <ul style="list-style-type: none"> • 0x01 -> Antenna 1. • 0x02 -> Antenna 2. • 0x03 -> Antenna 3. • 0x04 -> Antenna 4. 	
3	0x..	Activation time: <ul style="list-style-type: none"> • 0x01 ... 0x63 (1 ... 99 seconds) -> 'Impulsive' output activation • 0x81 -> 'Continuous' activation • 0x80 -> Deactivation 	

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command don't 'produce' data).

2.18 Read Device Status

The **BLUEBOX** will answer to this command with a series of information about the current status and particularly about the digital inputs status.

Byte	Value	Description	Notes
1	0x36	Command code	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers:

Byte	Value	Description	Notes
1	0x36	Command code	
2	0x..	BLUEBOX status byte 1. ASCII encoded byte.	See the device user manual for details.
3	0x..	BLUEBOX status byte 2. ASCII encoded byte.	See the device user manual for details.

2.19 RF Deactivation

In 'continuous' mode, this command is used to suspend the activity of the RF antennas connected to the **BLUEBOX**; see also 'RF activation' command.

Byte	Value	Description	Notes
1	0x38	Command code	

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command don't 'produce' data).

2.20 RF Activation

In 'continuous' mode, this command is used to resume the activity of the RF antennas connected to the **BLUEBOX**; see also 'RF Deactivation' command.

Byte	Value	Description	Notes
1	0x39	Command code	

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command don't 'produce' data).

2.21 Antennas Auto-Tuning

This command is used to start an auto-tuning procedure on the RF output channels to improve the reading performances of the **BLUEBOX**.

Byte	Value	Description	Notes
1	0xD4	Command code	

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command don't 'produce' data).

2.22 Buffer Data Request

This command sends back the code of the eventual transponder that is present in the buffer. When 'continuous' mode is enabled, the reply is immediate because the **BLUEBOX** sends back the data hold in the buffer that is managed by the 'continuous' identification activity; otherwise, the **BLUEBOX** performs readily the identification task under time out protection and sends back the result of the operation.

Byte	Value	Description	Notes
1	0x05	Command code	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) for LF devices with only 1 antenna:

Byte	Value	Description	Notes
1	0x..	Transponder code, 1st byte	The transponder code is 5 bytes long for SHORT, 10 bytes long for MEDIUM, 20 bytes long for LONG
i	0x..	Transponder code, i-th byte	i < 5 for SHORT i < 10 for MEDIUM i < 20 for LONG
n	0x..	Transponder code, n th byte	n = 5 for SHORT n = 10 for MEDIUM n = 20 for LONG



The case 'no transponder' is represented by a code composed by 5 null bytes (0x00, 0x00, 0x00, 0x00, 0x00).

b) for LF devices with 2 antennas:

Byte	Value	Description	Notes
1	0x..	Length in bytes of the code of the transponder identified by antenna nr. 1.	n = 5 for SHORT n = 10 for MEDIUM n = 20 for LONG
1+1	0x..	Transponder code identified by antenna nr. 1, 1st byte.	i = 1
1+i	0x..	Transponder code identified by antenna nr. 1, i-th byte.	i < 5 for SHORT i < 10 for MEDIUM i < 20 for LONG

Byte	Value	Description	Notes
1+n	0x..	Transponder code identified by antenna nr. 1, n-th byte.	n = 5 for SHORT n = 10 for MEDIUM n = 20 for LONG
2+n	0x..	Length in bytes of the code of the transponder identified by antenna nr. 2.	m = 5 for SHORT m = 10 for MEDIUM m = 20 for LONG
2+n+1	0x..	Transponder code identified by antenna nr. 2, 1st byte.	j = 1
2+n+j	0x..	Transponder code identified by antenna nr. 2, j-th byte.	j < 5 for SHORT j < 10 for MEDIUM j < 20 for LONG
2+n+m	0x..	Transponder code identified by antenna nr. 2, m-th byte.	m = 5 for SHORT m = 10 for MEDIUM m = 20 for LONG



In case of no transponder identified by one or both antennas the transponder code is replaced by five null bytes (0x00) while retaining the packet format described above.

c) for HF devices with only 1 antenna:

Byte	Value	Description	Notes
1	0x..	Transponder type. See Annex A for tag type and UID length table	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i-th byte	i < n
1+n	0x..	Transponder UID, n-th byte	n = UID length (See Annex A for tag type and UID length table)



The case 'no transponder' is represented by a code composed by 5 null bytes (0x00, 0x00, 0x00, 0x00, 0x00).

d) for HF devices with 2 antennas:

Byte	Value	Description	Notes
1	0x..	Transponder type read by antenna 1. See Annex A for tag type and UID length table	
1+1	0x..	Tsp UID read by antenna 1, 1 st byte	
1+i	0x..	Tsp UID read by antenna 1, i-th byte	$i < n$
1+n	0x..	Tsp UID read by antenna 1, n-th byte	$n = \text{UID length}$ (See Annex A for tag type and UID length table)
2+n	0x..	Transponder type read by antenna 2. See Annex A for tag type and UID length table.	
2+n+1	0x..	Tsp UID read by antenna 2, 1 st byte	
2+n+i	0x..	Tsp UID read by antenna 2, i-th byte	$i < m$
2+n+m	0x..	Tsp UID read by antenna 2, n-th byte	$m = \text{UID length}$ (See Annex A for tag type and UID length table)



In case of no transponder identified by one or both antennas the transponder code is replaced by five null bytes (0x00) while retaining the packet format described above.

e) for UHF devices in case of at least one ISO 18000-63 (EPC Class-1 Generation-2) transponder present:

Byte	Value	Description	Notes
1	0x..	Transponder type: <ul style="list-style-type: none"> 0x02: ISO 18000-63 (EPC Class-1 Generation-2). 	Optional parameter present only if the tag type information flag in the general parameters is active. See the device user manual for more info.
2	0x..	1st transponder code, 1st byte.	
...	...		
n+1	0x..	1st transponder code, n-th byte.	
...	0x..	1st seen RSSI Q value in dB of the 1st tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	1st seen RSSI I value in dB of the 1st tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Last seen RSSI Q value in dB of the 1st tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Last seen RSSI I value in dB of the 1st tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Reading antenna of the 1st tag: <ul style="list-style-type: none"> 0x01 -> Antenna 1. 0x02 -> Antenna 2. 0x03 -> Antenna 3. 0x04 -> Antenna 4. 	Optional parameter present only if the reading antenna information flag in the general parameters is active. See the device user manual for more info.
...	0x..	Gate crossing direction for the 1st identified tag:	Optional parameter present only if 'gate' mode is active.

Byte	Value	Description	Notes
		<ul style="list-style-type: none"> 0x01 -> Crossing from input 1 to input 2. 0x02 -> Crossing from input 2 to input 1. 	See the device user manual for more info.
...	...		
...	0x..	Transponder type: <ul style="list-style-type: none"> 0x02: ISO 18000-63 (EPC Class-1Generation-2). 	Optional parameter present only if the tag type information flag in the general parameters is active. See the device user manual for more info.
...	0x..	m-th transponder code, 1st byte.	
...	...		
...	0x..	m-th transponder code, n-th byte.	
...	0x..	1st seen RSSI Q value in dB of the m-th tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	1st seen RSSI I value in dB of the m-th tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Last seen RSSI Q value in dB of the m-th tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Last seen RSSI I value in dB of the m-th tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Reading antenna of the m-th tag: <ul style="list-style-type: none"> 0x01 -> Antenna 1. 0x02 -> Antenna 2. 0x03 -> Antenna 3. 0x04 -> Antenna 4. 	Optional parameter present only if the reading antenna information flag in the general parameters is active. See the device user manual for more info.

Byte	Value	Description	Notes
...	0x..	Gate crossing direction for the m-th identified tag: <ul style="list-style-type: none"> 0x01 -> Crossing from input 1 to input 2. 0x02 -> Crossing from input 2 to input 1. 	Optional parameter present only if 'gate' mode is active. See the device user manual for more info.
...	...		



The case 'no transponder' is represented by a code composed by 5 null bytes (0x00, 0x00, 0x00, 0x00, 0x00).

2.23 Queue Data Request

In 'continuous' mode, when the **BLUEBOX** finds a 'new' transponder, it inserts its code in the FIFO queue. This command sends back the first code present in the queue. After executing the command, the code must be deleted from the queue, otherwise each time you make a data request from the queue the same code will be returned.

Byte	Value	Description	Notes
1	0x06	Command code	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) for LF devices with only 1 antenna:

Byte	Value	Description	Notes
1	0x..	Transponder code, 1st byte	The transponder code is 5 bytes long for SHORT, 10 bytes long for MEDIUM, 20 bytes long for LONG
i	0x..	Transponder code, i th byte	i < 5 for SHORT i < 10 for MEDIUM i < 20 for LONG

Byte	Value	Description	Notes
n	0x..	Transponder code, n th byte	n = 5 for SHORT n = 10 for MEDIUM n = 20 for LONG



If the queue is empty, the answer message will consist of 5 null bytes (0x00, 0x00, 0x00, 0x00, 0x00).

b) for LF devices with 2 antennas:

Byte	Value	Description	Notes
1	0X..	Transponder code, 1st byte.	n = 5 for SHORT n = 10 for MEDIUM n = 20 for LONG
i	0x..	Transponder code, i-th byte.	i < 5 for SHORT i < 10 for MEDIUM i < 20 for LONG
n	0x..	Transponder code, n-th byte.	i = n
n+1	0x..	Antenna that have identified the tag	0x01: Antenna 1 0x02: Antenna 2



If the queue is empty, the answer message will consist of 5 null bytes (0x00, 0x00, 0x00, 0x00, 0x00).

c) for HF devices with only 1 antenna:

Byte	Value	Description	Notes
1	0x..	Transponder type. See Annex A for tag type and UID length table	
1+1	0x..	Transponder UID, 1 st byte	

Byte	Value	Description	Notes
1+i	0x..	Transponder UID, i-th byte	$i < n$
1+n	0x..	Transponder UID, n-th byte	n = UID length (See Annex A for tag type and UID length table)



If the queue is empty, the answer message will consist of 5 null bytes (0x00, 0x00, 0x00, 0x00, 0x00).

d) for HF devices with 2 antennas:

Byte	Value	Description	Notes
1	0x..	Transponder type. See Annex A for tag type and UID length table	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i-th byte	$i < n$
1+n	0x..	Transponder UID, n-th byte	n = UID length (See Annex A for tag type and UID length table)
2+n	0x..	Antenna that have identified the tag	0x01: Antenna 1 0x02: Antenna 2



If the queue is empty, the answer message will consist of 5 null bytes (0x00, 0x00, 0x00, 0x00, 0x00).

e) for UHF devices in case of one ISO 18000-63 (EPC Class-1 Generation-2) transponder present with n bytes ID length:

Byte	Value	Description	Notes
1	0x..	Transponder type: <ul style="list-style-type: none"> 0x02: ISO 18000-63 (EPC Class-1Generation-2). 	Optional parameter present only if the tag type information flag in the general parameters is active. See the device user manual for more info.
2	0x..	Transponder code, 1st byte.	
...	...		
n+1	0x..	Transponder code, n-th byte.	
...	0x..	1st seen RSSI Q value in dB of the tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	1st seen RSSI I value in dB of the tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Reading antenna of the identified tag: <ul style="list-style-type: none"> 0x01 -> Antenna 1. 0x02 -> Antenna 2. 0x03 -> Antenna 3. 0x04 -> Antenna 4. 	Optional parameter present only if the reading antenna information flag in the general parameters is active. See the device user manual for more info.
...	0x..	Gate crossing direction for the identified tag: <ul style="list-style-type: none"> 0x01 -> Crossing from input 1 to input 2. 0x02 -> Crossing from input 2 to input 1. 	Optional parameter present only if 'gate' mode is active. See the device user manual for more info.



If the queue is empty, the answer message will consist of 5 null bytes (0x00, 0x00, 0x00, 0x00, 0x00).

To delete the received code from the queue, the 'master' reply to the **BLUEBOX** with:

Byte	Value	Description	Notes
1	0x07	Command code	

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command doesn't 'produce' data).

2.24 Write Data to a EM4305 Transponder

This command is used to write data on the EM4305 transponder with the following possible formats:

- **EM4305 BLUEBOX SHORT**, the code is constituted by 40 bits divided into 10 nibbles (UNIQUE compatible) giving 5 bytes
- **EM4305 BLUEBOX MEDIUM**, the code is constituted by 80 bits divided into 20 nibbles giving 10 bytes
- **EM4305 BLUEBOX LARGE**, the code is constituted by 160 bits divided into 40 nibbles giving 20 bytes

Byte	Value	Description	Notes
1	0x19	Command code	
1+1	0x..	Transponder code, 1st byte	The transponder code is 5 bytes long for SHORT, 10 bytes long for MEDIUM, 20 bytes long for LONG
1+i	0x..	Transponder code, i th byte	i < 5 for SHORT i < 10 for MEDIUM i < 20 for LONG
1+n	0x..	Transponder code, n th byte	n = 5 for SHORT n = 10 for MEDIUM n = 20 for LONG



For devices with 2 antennas, the command code 0x19 is used to work with the antenna nr 1 while the command code 0x69 is used to work with antenna nr 2.

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command doesn't 'produce' data).

2.25 Read ID Code of a EM4305 Transponder

This command is used to get the ID code of the EM4305 transponder, constituted by 4 bytes.

Byte	Value	Description	Notes
1	0x18	Command code	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the ID has been successfully read:

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x00	Status Ok	
2+1	0x..	Transponder identification code, 1st byte	The transponder ID code is 4 bytes long
2+i	0x..	Transponder identification code, i th byte	
2+4	0x..	Transponder identification code, 4th byte	i = 4

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x02	Status transponder present with errors	

c) if no transponder is present:

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x18 is used to work with the antenna nr 1 while the command code 0x68 is used to work with antenna nr 2.

2.26 Write Data to a T5557 Transponder

This command is used to write data on the T5557 transponder with the following possible formats:

- **T5557 BLUEBOX SHORT**, the code is constituted by 40 bits divided into 10 nibbles (UNIQUE compatible) giving 8 bytes
- **T5557 BLUEBOX MEDIUM**, the code is constituted by 80 bits divided into 20 nibbles giving 10 bytes
- **T5557 BLUEBOX LARGE**, the code is constituted by 160 bits divided into 40 nibbles giving 20 bytes

Byte	Value	Description	Notes
1	0x1D	Command code	
1+1	0x..	Transponder code, 1st byte	The transponder code is 5 bytes long for SHORT, 10 bytes long for MEDIUM, 20 bytes long for LONG
1+i	0x..	Transponder code, i th byte	i < 5 for SHORT i < 10 for MEDIUM i < 20 for LONG
1+n	0x..	Transponder code, n th byte	n = 5 for SHORT n = 10 for MEDIUM n = 20 for LONG



For devices with 2 antennas, the command code 0x1D is used to work with the antenna nr 1 while the command code 0x6D is used to work with antenna nr 2.

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command doesn't 'produce' data).

2.27 Read ID Code of a T5557 Transponder

This command is used to get the ID code of the T5557 transponder, constituted by 8 bytes.

Byte	Value	Description	Notes
1	0x1C	Command code	

If the command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the ID has been successfully read:

Byte	Value	Description	Notes
1	0x1C	Command code	
2	0x00	Status Ok	
2+i	0x..	Transponder identification code, 1st byte	The transponder ID code is 8 bytes long
2+i	0x..	Transponder identification code, i th byte	
2+8	0x..	Transponder identification code, 8th byte	i = 8

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x1C	Command code	
2	0x02	Status transponder present with errors	

c) if no transponder is present:

Byte	Value	Description	Notes
1	0x1C	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x1C is used to work with the antenna nr 1 while the command code 0x6C is used to work with antenna nr 2.

2.28 Write Data to a Q5 Transponder

This command is used to write data on the Q5 transponder with the following possible formats:

- **Q5 BLUEBOX SHORT**, the code is constituted by 40 bits divided into 10 nibbles (UNIQUE compatible) giving 5 bytes
- **Q5 BLUEBOX MEDIUM**, the code is constituted by 80 bits divided into 20 nibbles giving 10 bytes
- **Q5 BLUEBOX LARGE**, the code is constituted by 160 bits divided into 40 nibbles giving 20 bytes

Byte	Value	Description	Notes
1	0X21	Command code	
1+1	0x..	Transponder code, 1st byte	The transponder code is 5 bytes long for SHORT, 10 bytes long for MEDIUM, 20 bytes long for LONG
1+i	0x..	Transponder code, i th byte	i < 5 for SHORT i < 10 for MEDIUM i < 20 for LONG
1+n	0x..	Transponder code, n th byte	n = 5 for SHORT n = 10 for MEDIUM n = 20 for LONG



For devices with 2 antennas, the command code 0x21 is used to work with the antenna nr 1 while the command code 0x71 is used to work with antenna nr 2.

The answer to this command is given in short form through the 'Reply Ack' / 'Reply Nak' flags (this command doesn't 'produce' data).

2.29 Read ID Code of a Q5 Transponder

This command is used to get the ID code of the Q5 transponder, constituted by 5 bytes.

Byte	Value	Description	Notes
1	0X20	Command code	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the ID has been successfully read:

Byte	Value	Description	Notes
1	0X20	Command code	
2	0x00	Status Ok	
2+1	0x..	Transponder identification code, 1st byte	The transponder ID code is 5 bytes long
2+i	0x..	Transponder identification code, i th byte	
2+5	0x..	Transponder identification code, 5th byte	
			i < 5
			i = 5

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0X20	Command code	
2	0x02	Status transponder present with errors	

c) if no transponder is present:

Byte	Value	Description	Notes
1	0X20	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x20 is used to work with the antenna nr 1 while the command code 0x70 is used to work with antenna nr 2.

2.30 Write Data to a HITAG S Transponder

This command is used to write data on the HITAG S transponder with the following possible formats:

- **HITAG S BLUEBOX SHORT**, the code is constituted by 40 bits divided into 10 nibbles (UNIQUE compatible) giving 5 bytes
- **HITAG S BLUEBOX MEDIUM**, the code is constituted by 80 bits divided into 20 nibbles giving 10 bytes

Byte	Value	Description	Notes
1	0x23	Command code	
1+1	0x..	Transponder code, 1st byte	The transponder code is 5 bytes long for SHORT, 10 bytes long for MEDIUM
1+i	0x..	Transponder code, i th byte	i < 5 for SHORT i < 10 for MEDIUM
1+n	0x..	Transponder code, n th byte	n = 5 for SHORT n = 10 for MEDIUM

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the it has been successfully written:

Byte	Value	Description	Notes
1	0x23	Command code	
2	0x00	Status Ok	

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x23	Command code	

Byte	Value	Description	Notes
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x23	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x23 is used to work with the antenna nr 1 while the command code 0x73 is used to work with antenna nr 2.

2.31 Read ID Code of a HITAG 1 /HITAG S Transponder

This command is used to get the ID code of the HITAG 1 / HITAG S transponder, constituted by 4 bytes.

Byte	Value	Description	Notes
1	0X22	Command code	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the ID has been successfully read:

Byte	Value	Description	Notes
1	0X22	Command code	
2	0x00	Status Ok	
3	0x..	Transponder type: <ul style="list-style-type: none"> • 0x01: HITAG S 256 • 0x02: HITAG S 2048 • 0x03: HITAG 1 	

Byte	Value	Description	Notes
3+1	0x..	Transponder identification code, 1st byte	The transponder ID code is 4 bytes long
3+i	0x..	Transponder identification code, i th byte	$i < 4$
3+4	0x..	Transponder identification code, 4th byte	$i = 4$

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0X22	Command code	
2	0x02	Status transponder present with errors	

c) if no transponder is present:

Byte	Value	Description	Notes
1	0X22	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x22 is used to work with the antenna nr 1 while the command code 0x72 is used to work with antenna nr 2.

2.32 Read a Page of a HITAG 1 / HITAG S Transponder

This command is used to get a data page of the HITAG 1 / HITAG S transponder, constituted by 32 bits (4 bytes). Note that it is necessary to know the ID code of the transponder. Refer to the related datasheet to get more information about the HITAG transponder.

Byte	Value	Description	Notes
1	0x24	Command code	
1+1	0x..	Transponder identification code, 1st byte	The transponder ID code is 4 bytes long
1+i	0x..	Transponder identification code, i th byte	$i < 4$
1+4	0x..	Transponder identification code, 4th byte	$i = 4$
6	0x..	Page of transponder to be read	0x00 ... 0x3F

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the page has been successfully read:

Byte	Value	Description	Notes
1	0x24	Command code	
2	0x00	Status Ok	
2+1	0x..	Transponder data, 1st byte	Every transponder page is 4 bytes long
2+i	0x..	Transponder data, i th byte	$i < 4$
2+4	0x..	Transponder data, 4th byte	$i = 4$

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x24	Command code	
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x24	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x24 is used to work with the antenna nr 1 while the command code 0x74 is used to work with antenna nr 2.

2.33 Write a Page of a HITAG 1 / HITAG S Transponder

This command is used to write a data page of the HITAG 1 / HITAG S transponder, constituted by 32 bits (4 bytes). Note that it is necessary to know the ID code of the transponder. Refer to the related datasheet to get more information about the HITAG transponder.

Byte	Value	Description	Notes
1	0x25	Command code	
1+1	0x..	Transponder identification code, 1st byte	The transponder ID code is 4 bytes long
1+i	0x..	Transponder identification code, i th byte	$i < 4$
1+4	0x..	Transponder identification code, 4th byte	$i = 4$
6		Transponder page to be written	0x08 ... 0x3F
6+1	0x..	Transponder data, 1st byte	Every transponder page is 4 bytes long
6+i	0x..	Transponder data, i th byte	$i < 4$
6+4	0x..	Transponder data, 4th byte	$i = 4$

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the page is been successfully written:

Byte	Value	Description	Notes
1	0x25	Command code	
2	0x00	Status Ok	

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x25	Command code	
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x25	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x25 is used to work with the antenna nr 1 while the command code 0x75 is used to work with antenna nr 2.

2.34 Read ID Code of a HITAG 2 Transponder

This command is used to get the ID code of the HITAG 2 transponder, constituted by 4 bytes.

Byte	Value	Description	Notes
1	0X48	Command code	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the ID has been successfully read:

Byte	Value	Description	Notes
1	0X48	Command code	
2	0x00	Status Ok	
2+1	0x..	Transponder identification code, 1st byte	The transponder ID code is 4 bytes long
2+i	0x..	Transponder identification code, i th byte	$i < 4$

Byte	Value	Description	Notes
2+4	0x..	Transponder identification code, 4th byte	i = 4

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0X48	Command code	
2	0x02	Status transponder present with errors	

c) if no transponder is present:

Byte	Value	Description	Notes
1	0X48	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x48 is used to work with the antenna nr 1 while the command code 0x98 is used to work with antenna nr 2.

2.35 Read a Page of a HITAG 2 Transponder

This command is used to get a data page of the HITAG 2 transponder, constituted by 32 bits (4 bytes). Note that it is necessary to know the ID code of the transponder. Refer to the related datasheet to get more information about the HITAG transponder.

Byte	Value	Description	Notes
1	0x4A	Command code	
1+1	0x..	Transponder identification code, 1st byte	The transponder ID code is 4 bytes long
1+i	0x..	Transponder identification code, i th byte	i < 4

Byte	Value	Description	Notes
1+4	0x..	Transponder identification code, 4th byte	$i = 4$
6	0x..	Page of transponder to be read	0x00 ... 0x07

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the page has been successfully read:

Byte	Value	Description	Notes
1	0x4A	Command code	
2	0x00	Status Ok	
2+1	0x..	Transponder data, 1st byte	Every transponder page is 4 bytes long
2+i	0x..	Transponder data, i th byte	$i < 4$
2+4	0x..	Transponder data, 4th byte	$i = 4$

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x4A	Command code	
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x4A	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x4A is used to work with the antenna nr 1 while the command code 0x9A is used to work with antenna nr 2.

2.36 Write a Page of a HITAG 2 Transponder

This command is used to write a data page of the HITAG 2 transponder, constituted by 32 bits (4 bytes). Note that it is necessary to know the ID code of the transponder. Refer to the related datasheet to get more information about the HITAG transponder.

Byte	Value	Description	Notes
1	0x4B	Command code	
1+1	0x..	Transponder identification code, 1st byte	The transponder ID code is 4 bytes long
1+i	0x..	Transponder identification code, i th byte	$i < 4$
1+4	0x..	Transponder identification code, 4th byte	$i = 4$
6		Transponder page to be written	0x00 ... 0x07
6+1	0x..	Transponder data, 1st byte	Every transponder page is 4 bytes long
6+i	0x..	Transponder data, i th byte	$i < 4$
6+4	0x..	Transponder data, 4th byte	$i = 4$

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the page is been successfully written:

Byte	Value	Description	Notes
1	0x4B	Command code	
2	0x00	Status Ok	

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x4B	Command code	
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x4B	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x4B is used to work with the antenna nr 1 while the command code 0x9B is used to work with antenna nr 2.

2.37 'Reset' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to reset the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder. If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be resumed as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

Byte	Value	Description	Notes
1	0x40	Command code	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the command has been successfully executed:

Byte	Value	Description	Notes
1	0x40	Command code	
2	0x00	Status Ok	

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x40	Command code	

Byte	Value	Description	Notes
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x40	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x40 is used to work with the antenna nr 1 while the command code 0x60 is used to work with antenna nr 2.

2.38 'Login' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to log in the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder. If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

Byte	Value	Description	Notes
1	0x41	Command code	
1+1	0x..	Transponder password, 1st byte	The transponder password is 4 bytes long
1+i	0x..	Transponder password, i th byte	$i < 4$
1+4	0x..	Transponder password, 4th byte	$i = 4$

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the command has been successfully executed:

Byte	Value	Description	Notes
1	0x41	Command code	
2	0x00	Status Ok	

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x41	Command code	
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x41	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x41 is used to work with the antenna nr 1 while the command code 0x61 is used to work with antenna nr 2.

2.39 'Write Password' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to set the password of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder. If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

Byte	Value	Description	Notes
1	0x41	Command code	
1+1	0x..	Transponder password, 1st byte	The transponder password is 4 bytes long
1+i	0x..	Transponder password, i th byte	$i < 4$
1+4	0x..	Transponder password, 4th byte	$i = 4$
5+1	0x..	New password to be set, 1st byte	The transponder password is 4 bytes long
5+i	0x..	New password to be set, i th byte	$i < 4$
5+4	0x..	New password to be set, 4th byte	$i = 4$

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the command has been successfully executed:

Byte	Value	Description	Notes
1	0x42	Command code	
2	0x00	Status Ok	

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x42	Command code	
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x42	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1 while the command code 0x62 is used to work with antenna nr 2.

2.40 'Standard Read' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to get the data relative to 'standard read' mode of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder. If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

Byte	Value	Description	Notes
1	0x43	Command code	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the command has been successfully executed:

Byte	Value	Description	Notes
1	0x43	Command code	
2	0x00	Status Ok	
2+1	0x..	First long word read, 1st byte	Every long word is 4 bytes long
2+i	0x..	First long word read, i th byte	i < 4
2+4	0x..	First long word read, 4th byte	i = 4
...	...		
...	0x..	Last long word read, 1st byte	Every long word is 4 bytes long
...	0x..	Last long word read, i th byte	i < 4
...	0x..	Last long word read, 4th byte	i = 4

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x43	Command code	
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x43	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x43 is used to work with the antenna nr 1 while the command code 0x63 is used to work with antenna nr 2.

2.41 'Selective Read' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to read the data relative to 1 or more long word/s of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder. If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

Byte	Value	Description	Notes
1	0x44	Command code	
2	0x..	Address of the first long word to be read	
3	0x..	Address of the last long word to be read	

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the command has been successfully executed:

Byte	Value	Description	Notes
1	0x44	Command code	
2	0x00	Status Ok	
2+1	0x..	Last long word read, 1st byte	Every long word is 4 bytes long
2+i	0x..	Last long word read, i th byte	i < 4
2+4	0x..	Last long word read, 4th byte	i = 4
...	...		
...	0x..	First long word read, 1st byte	Every long word is 4 bytes long
...	0x..	First long word read, i th byte	i < 4
...	0x..	First long word read, 4th byte	i = 4

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x44	Command code	
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x44	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x44 is used to work with the antenna nr 1 while the command code 0x64 is used to work with antenna nr 2.

2.42 'Write Word' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to write the data relative to a long word of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder. If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

Byte	Value	Description	Notes
1	0x45	Command code	
2	0x..	Address of the long word to be written	
2+1	0x..	The long word to be written, 1st byte	Every long word is 4 bytes long
2+i	0x..	The long word to be written, i th byte	$i < 4$
2+4	0x..	The long word to be written, 4th byte	$i = 4$

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers, if a transponder is present and the command has been successfully executed:

Byte	Value	Description	Notes
1	0x45	Command code	
2	0x00	Status Ok	

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x45	Command code	
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x45	Command code	

Byte	Value	Description	Notes
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x45 is used to work with the antenna nr 1 while the command code 0x65 is used to work with antenna nr 2.

2.43 'Write Several Words' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to write the data relative to more long words of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder. If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be resumed as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

Byte	Value	Description	Notes
1	0x45	Command code	
2	0x..	Address of the long word to be written	
2+1	0x..	The first long word to be written, 1st byte	Every long word is 4 bytes long
2+1	0x..	The first long word to be written, i th byte	$i < 4$
2+4	0x..	The first long word to be written, 4th byte	$i = 4$
...	...		
...	0x..	The last long word to be written, 1st byte	Every long word is 4 bytes long
...	0x..	The last long word to be written, i th byte	$i < 4$
...	0x..	The last long word to be written, 4th byte	$i = 4$

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers, if a transponder is present and the command has been successfully executed:

Byte	Value	Description	Notes
1	0x46	Command code	
2	0x00	Status Ok	

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x46	Command code	
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x46	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x46 is used to work with the antenna nr 1 while the command code 0x66 is used to work with antenna nr 2.

2.44 'Read After Write Word' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to write and read back the data relative to a long word of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder. If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

Byte	Value	Description	Notes
1	0x47	Command code	
2	0x..	Address of the long word to be written	
2+1	0x..	The long word to be written, 1st byte	Every long word is 4 bytes long
2+i	0x..	The long word to be written, i th byte	$i < 4$
2+4	0x..	The long word to be written, 4th byte	$i = 4$

If command is not successfully executed, the answer is given in short form through the 'Reply Ack' / 'Reply Nak' flags; otherwise, the **BLUEBOX** answers,

a) if a transponder is present and the command has been successfully executed:

Byte	Value	Description	Notes
1	0x47	Command code	
2	0x00	Status Ok	
2+1	0x..	Transponder long word read back, 1st byte	Every long word is 4 bytes long
2+i	0x..	Transponder long word read back, i th byte	$i < 4$
2+4	0x..	Transponder long word read back, 4th byte	$i = 4$

b) if a transponder is present but errors occurred:

Byte	Value	Description	Notes
1	0x47	Command code	
2	0x02	Status transponder present with errors	

c) if a transponder is not present:

Byte	Value	Description	Notes
1	0x47	Command code	
2	0x01	Status transponder not present	



For devices with 2 antennas, the command code 0x47 is used to work with the antenna nr 1 while the command code 0x67 is used to work with antenna nr 2.

2.45 ISO 15693 Transponders 'Inventory' Command

This command is used to get the UID code of the identified ISO 15693 transponders that are present near the antenna/s.

Byte	Value	Description	Notes
1	0x10	Command code	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) case of absence of transponder:

Byte	Value	Description	Notes
1	0x10	Command code	
2	0x01	Status no transponder	

b) case of presence of transponder/s:

Byte	Value	Description	Notes
1	0x10	Command code	
2	0x00	Status Ok	
2+1	0x..	Transponder 1 UID, 1 st byte	
2+i	0x..	Transponder 1 UID, i-th byte	$i < 8$
2+8	0x..	Transponder 1 UID, 8 th byte	
...	
...	0x..	Transponder n UID, 1 st byte	
...	0x..	Transponder n UID, j-th byte	$j < 8$
2+n*8	0x..	Transponder n UID, 8 th byte	



For devices with 2 antennas, the command code 0x10 is used to work with the antenna nr 1 while the command code 0x90 is used to work with antenna nr 2.

2.46 Read a Data Block of an ISO 15693 Transponder

This command is used to get a data block of a known (UID) ISO 15693 transponder. Note that the number of bytes of a block and the number of blocks depends on the transponder type; for example, the **NXP I CODE SLI** transponder is organized in blocks of 4 bytes, the **Fujitsu MB89R118** transponder is organized in blocks of 8 bytes, for more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x11	Command code	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i-th byte	$i < 8$
1+8	0x..	Transponder UID, 8 th byte	
10	0x..	Block number	Max value depends on the specific transponder

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the addressed transponder is not present:

Byte	Value	Description	Notes
1	0x11	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x11	Command code	
2	0x02	Status transponder present with errors	

c) if the addressed transponder is present:

Byte	Value	Description	Notes
1	0x11	Command code	
2	0x00	Status Ok	
2+1	0x..	Data of the block, 1 st byte	
2+i	0x..	Data of the block, i-th byte	$i < n$
2+n	0x..	Data of the block, n-th byte	n depends on the specific transponder



For devices with 2 antennas, the command code 0x11 is used to work with the antenna nr 1 while the command code 0x91 is used to work with antenna nr 2.

2.47 Write a Data Block of an ISO 15693 Transponder

This command is used to write a data block of a known (UID) ISO 15693 transponder. Note that the number of bytes of a block depends on the transponder type; for example, the **NXP I CODE SLI** transponder is organized in blocks of 4 bytes, the **Fujitsu MB89R118** transponder is organized in blocks of 8 bytes, for more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x12	Command code	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i-th byte	$i < 8$
1+8	0x..	Transponder UID, 8 th byte	
10	0x..	Block number	Max value depends on the specific transponder
10+1	0x..	Data to be written in the block, 1 st byte	
10+i		Data to be written in the block, i th byte	$i < n$
10+n		Data to be written in the block, n th byte	n depends on the specific transponder

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the addressed transponder is not present:

Byte	Value	Description	Notes
1	0x12	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x12	Command code	
2	0x02	Status transponder present with errors	

c) if the addressed transponder is present and the block has been correctly written:

Byte	Value	Description	Notes
1	0x12	Command code	
2	0x00	Status Ok	



For devices with 2 antennas, the command code 0x12 is used to work with the antenna nr 1 while the command code 0x92 is used to work with antenna nr 2.

2.48 Lock a Data Block of an ISO 15693 Transponder

This command is used to lock a data block of a known (UID) ISO 15693 transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x13	Command code	
1+1	0x..	Transponder UID, 1 st byte	

Byte	Value	Description	Notes
1+i	0x..	Transponder UID, i-th byte	i < 8
1+8	0x..	Transponder UID, 8 th byte	
10	0x..	Block number	Max value depends on the specific transponder

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the addressed transponder is not present:

Byte	Value	Description	Notes
1	0x13	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x13	Command code	
2	0x02	Status transponder present with errors	

c) if the addressed transponder is present:

Byte	Value	Description	Notes
1	0x13	Command code	
2	0x00	Status Ok	



For devices with 2 antennas, the command code 0x13 is used to work with the antenna nr 1 while the command code 0x93 is used to work with antenna nr 2.

2.49 ISO 15963 Transponder 'Get System Info' Command

This command is used to get the system info data block of a known (UID) ISO 15693 transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x14	Command code	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i-th byte	i < 8
1+8	0x..	Transponder UID, 8 th byte	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the addressed transponder is not present:

Byte	Value	Description	Notes
1	0x14	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x14	Command code	
2	0x02	Status error	

c) if the addressed transponder is present:

Byte	Value	Description	Notes
1	0x14	Command code	
2	0x00	Status Ok	
3	0x..	Info Flags: single bits are dedicated to specify the presence of the following fields (0 → absent, 1 → present): <ul style="list-style-type: none"> • Bit 7...4: Not used • Bit 3: IC Reference (1 byte) 	

Byte	Value	Description	Notes
		<ul style="list-style-type: none"> • Bit 2: Memory Size (2 bytes) • Bit 1: AFI (1 byte) • Bit 0: DSFID (1 byte) 	
4+1	0x..	Transponder UID, 1 st byte	
4+i	0x..	Transponder UID, i th byte	i < 8
4+8	0x..	Transponder UID, 8 th byte	
...	0x..	DSFID	Present only if bit 0 of Info Flags is set
...	0x..	AFI	Present only if bit 1 of Info Flags is set
...	0x..	Memory Size – Block Size in bytes 0x00 (1 byte) ... 0x1F (32 bytes)	Present only if bit 2 of Info Flags is set
...	0x..	Memory Size – Number of Blocks 0x00 (1 block) ... 0xFF (256 blocks)	Present only if bit 2 of Info Flags is set
...	0x..	IC Reference	Present only if bit 3 of Info Flags is set



For devices with 2 antennas, the command code 0x14 is used to work with the antenna nr 1 while the command code 0x94 is used to work with antenna nr 2.

2.50 ISO 15693 Transponder 'General Protocol' Command

This command allows to send any ISO 15693 general format protocol command (flags field, command code field, parameters fields, application data fields) to a ISO 15693 transponder and to receive, in case of successful operation, the response of the transponder (flag field, parameters fields, data fields). For more details see the specific transponder data sheet and ISO 15693 protocol. If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as this command is used; it will be resumed automatically when another type of command will be executed.

Byte	Value	Description	Notes
1	0x15	Command code	
1+1	0x..	Data to send to the tag, 1 st byte	
1+i	0x..	Data to send to the tag, i-th byte	$i < n$
1+n	0x..	Data to send to the tag, n-th byte	n depends on the specific protocol command

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x15	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x15	Command code	
2	0x02	Status error	

c) if the transponder is present and the command is successfully executed:

Byte	Value	Description	Notes
1	0x15	Command code	
2	0x00	Status Ok	
2+1	0x..	Data received from the tag, 1 st byte	
2+i	0x..	Data received from the tag, i-th byte	$i < n$
2+n	0x..	Data received from the tag, n-th byte	n depends on the specific protocol command response



For devices with 2 antennas, the command code 0x15 is used to work with the antenna nr 1 while the command code 0x95 is used to work with antenna nr 2.

2.51 ISO 14443A Transponder 'Inventory' Command

This command is used to get the UID code of a ISO 14443A transponder - **MIFARE Ultralight, MIFARE 1k (UID 4), MIFARE 4k (UID 4), MIFARE 1k (UID 7), MIFARE 4k (UID 7), MIFARE Desfire, MIFARE PLUS 2k, MIFARE Plus 4k, NTAG213/215/216** - that is present near the antenna.

Byte	Value	Description	Notes
1	0x18	Command code	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) case of absence of transponder:

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x01	Status no transponder	

b) case of presence of a transponder:

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x00	Status Ok	
2+1	0x..	Transponder 1 type. See Annex A for tag type and UID length table	
2+1+1	0x..	Transponder 1 UID, 1 st byte	
2+1+i	0x..	Transponder 1 UID, i-th byte	$i < n$
2+1+n	0x..	Transponder 1 UID, n-th byte	$n = \text{UID length (See Annex A for tag type and UID length table)}$

Byte	Value	Description	Notes
...	
...	0x..	Transponder k type. See Annex A for tag type and UID length table	
...	0x..	Transponder k UID, 1 st byte	
...	0x..	Transponder k UID, j-th byte	$j < m$
...	0x..	Transponder k UID, m-th byte	$m = \text{UID length (See Annex A for tag type and UID length table)}$



For devices with 2 antennas, the command code 0x18 is used to work with the antenna nr 1 while the command code 0x98 is used to work with antenna nr 2.

2.52 Read a Data Block of a MIFARE 1k/4k (UID 4) Transponder

This command is used to get a data block (16 bytes) of a known (UID) **MIFARE 1k/4k (UID 4)** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x19	Command code	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i-th byte	$i < 4$
1+4	0x..	Transponder UID, 4 th byte	
6	0x..	Key type: <ul style="list-style-type: none"> 0x00: Key A 0x01: Key B 	
6+1	0x..	Key code, 1 st byte	
6+j	0x..	Key code, j-th byte	$j < 6$
6+6	0x..	Key code, 6 th byte	
13	0x..	Block number:	

Byte	Value	Description	Notes
		<ul style="list-style-type: none"> 0x00 ... 0x3F for MIFARE 1k 0x00 ... 0xFF for MIFARE 4k 	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x19	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x19	Command code	
2	0x02	Status transponder present with errors	

c) if the transponder is present and the command is successfully executed:

Byte	Value	Description	Notes
1	0x19	Command code	
2	0x00	Status Ok	
2+1	0x..	Data of the block, 1 st byte	
2+i	0x..	Data of the block, i-th byte	i < 16
2+16	0x..	Data of the block, 16 th byte	



For devices with 2 antennas, the command code 0x19 is used to work with the antenna nr 1 while the command code 0x99 is used to work with antenna nr 2.

2.53 Write a Data Block of a MIFARE 1k/4k (UID 4) Transponder

This command is used to write a data block (16 bytes) of a known (UID) **MIFARE 1k/4k (UID 4)** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x1A	Command code	
1+1	0x..	transponder UID, 1 st byte	
1+i	0x..	transponder UID, i-th byte	$i < 4$
1+4	0x..	transponder UID, 4 th byte	
6	0x..	Key type: <ul style="list-style-type: none"> 0x00: Key A 0x01: Key B 	
6+1	0x..	Key code, 1 st byte	
6+j	0x..	Key code, j-th byte	$j < 6$
6+6	0x..	Key code, 6 th byte	
13	0x..	Block number: <ul style="list-style-type: none"> 0x00 ... 0x3F for MIFARE 1k 0x00 ... 0xFF for MIFARE 4k 	
13+1	0x..	Data to be written in the block, 1 st byte	
13+k	0x..	Data to be written in the block, k th byte	$k < 16$
13+16	0x..	Data to be written in the block, 16 th byte	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x1A	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x1A	Command code	
2	0x02	Status transponder present with errors	

c) if the transponder is present and the command is successfully executed:

Byte	Value	Description	Notes
1	0x1A	Command code	
2	0x00	Status Ok	



For devices with 2 antennas, the command code 0x1A is used to work with the antenna nr 1 while the command code 0x9A is used to work with antenna nr 2.

2.54 Read a Data Block of a MIFARE 1k/4k (UID 7) Transponder

This command is used to get a data block (16 bytes) of a known (UID) **MIFARE 1k/4k (UID 7)** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x1D	Command code	
1+1	0x..	transponder UID, 1 st byte	
1+i	0x..	transponder UID, i-th byte	$i < 7$
1+7	0x..	transponder UID, 7 th byte	
9	0x..	Key type: <ul style="list-style-type: none"> 0x00: Key A 0x01: Key B 	
9+1	0x..	Key code, 1 st byte	
9+j	0x..	Key code, j-th byte	$j < 6$
9+6	0x..	Key code, 6 th byte	

Byte	Value	Description	Notes
16	0x..	Block number: <ul style="list-style-type: none"> • 0x00 ... 0x3F for MIFARE 1k • 0x00 ... 0xFF for MIFARE 4k 	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x1D	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x1D	Command code	
2	0x02	Status transponder present with errors	

c) if the transponder is present and the command is successfully executed:

Byte	Value	Description	Notes
1	0x1D	Command code	
2	0x00	Status Ok	
2+1	0x..	Data of the block, 1 st byte	
2+i	0x..	Data of the block, i-th byte	i < 16
2+16	0x..	Data of the block, 16 th byte	



For devices with 2 antennas, the command code 0x1D is used to work with the antenna nr 1 while the command code 0x9D is used to work with antenna nr 2.

2.55 Write a Data Block of a MIFARE 1k/4k (UID 7) Transponder

This command is used to write a data block (16 bytes) of a known (UID) **MIFARE 1k/4k (UID 7)** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x1E	Command code	
1+1	0x..	transponder UID, 1 st byte	
1+i	0x..	transponder UID, i-th byte	$i < 7$
1+7	0x..	transponder UID, 7 th byte	
9	0x..	Key type: <ul style="list-style-type: none"> 0x00: Key A 0x01: Key B 	
9+1	0x..	Key code, 1 st byte	
9+j	0x..	Key code, j-th byte	$j < 6$
9+6	0x..	Key code, 6 th byte	
16	0x..	Block number: <ul style="list-style-type: none"> 0x00 ... 0x3F for MIFARE 1k 0x00 ... 0xFF for MIFARE 4k 	
16+1	0x..	Data to be written in the block, 1 st byte	
16+k	0x..	Data to be written in the block, k th byte	$k < 16$
16+16	0x..	Data to be written in the block, 16 th byte	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x1E	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x1E	Command code	
2	0x02	Status transponder present with errors	

c) if the transponder is present and the command is succesfully executed:

Byte	Value	Description	Notes
1	0x1E	Command code	
2	0x00	Status Ok	



For devices with 2 antennas, the command code 0x1E is used to work with the antenna nr 1 while the command code 0x9E is used to work with antenna nr 2.

2.56 Read a Data Page of a MIFARE Ultralight transponder

This command is used to get a data page (4 bytes) of a known (UID) **MIFARE Ultralight** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x1B	Command code	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i-th byte	$i < 7$
1+7	0x..	Transponder UID, 7 th byte	
9	0x..	Page number: • 0x00 ... 0x0F for MIFARE Ultralight	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x1B	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x1B	Command code	
2	0x02	Status page not supported or errors	

c) if the transponder is present and the command is succesfully executed:

Byte	Value	Description	Notes
1	0x1B	Command code	
2	0x00	Status Ok	
2+1	0x..	Data of the page, 1 st byte	
2+i	0x..	Data of the page, i-th byte	i < 4
2+4	0x..	Data of the page, 4 th byte	



For devices with 2 antennas, the command code 0x1B is used to work with the antenna nr 1 while the command code 0x9B is used to work with antenna nr 2.

2.57 Write a Data Page of a MIFARE Ultralight Transponder

This command is used to write a data page (4 bytes) of a known (UID) **MIFARE Ultralight** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x1C	Command code	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i-th byte	$i < 7$
1+7	0x..	Transponder UID, 7 th byte	
9	0x..	Page number: • 0x00 ... 0x0F for MIFARE Ultralight	
9+1	0x..	Data to be written in the page, 1 st byte	
9+j	0x..	Data to be written in the page, j-th byte	$j < 4$
9+4	0x..	Data to be written in the page, 4 th byte	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x1C	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x1C	Command code	
2	0x02	Status error	

c) if the transponder is present and the command is successfully executed:

Byte	Value	Description	Notes
1	0x1C	Command code	
2	0x00	Status Ok	



For devices with 2 antennas, the command code 0x1C is used to work with the antenna nr 1 while the command code 0x9C is used to work with antenna nr 2.

2.58 Read a Data Page of a NTAG213/215/216 transponder

This command is used to get a data page (4 bytes) of a known (UID) **NTAG213/215/216** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x1B	Command code	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i-th byte	$i < 7$
1+7	0x..	Transponder UID, 7 th byte	
9	0x..	Page number: <ul style="list-style-type: none"> • 0x00 ... 0x2C for NTAG213 • 0x00 ... 0x86 for NTAG215 • 0x00 ... 0xE6 for NTAG216 	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x1B	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x1B	Command code	
2	0x02	Status page not supported or errors	

c) if the transponder is present and the command is successfully executed:

Byte	Value	Description	Notes
1	0x1B	Command code	
2	0x00	Status Ok	
2+1	0x..	Data of the page, 1 st byte	
2+i	0x..	Data of the page, i-th byte	$i < 4$
2+4	0x..	Data of the page, 4 th byte	



For devices with 2 antennas, the command code 0x1B is used to work with the antenna nr 1 while the command code 0x9B is used to work with antenna nr 2.

2.59 Write a Data Page of a NTAG213/215/216 Transponder

This command is used to write a data page (4 bytes) of a known (UID) **NTAG213/215/216** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x1C	Command code	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i-th byte	$i < 7$
1+7	0x..	Transponder UID, 7 th byte	
9	0x..	Page number: <ul style="list-style-type: none"> • 0x00 ... 0x2C for NTAG213 • 0x00 ... 0x86 for NTAG215 • 0x00 ... 0xE6 for NTAG216 	
9+1	0x..	Data to be written in the page, 1 st byte	
9+j	0x..	Data to be written in the page, j-th byte	$j < 4$
9+4	0x..	Data to be written in the page, 4 th byte	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x1C	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x1C	Command code	
2	0x02	Status page not supported or errors	

c) if the transponder is present and the command is successfully executed:

Byte	Value	Description	Notes
1	0x1C	Command code	
2	0x00	Status Ok page successfully written	



For devices with 2 antennas, the command code 0x1C is used to work with the antenna nr 1 while the command code 0x9C is used to work with antenna nr 2.

2.60 ISO 14443A-4 Transponder 'RATS' Command

This command allows to select and send a RATS (Request for Answer To Select) command to a ISO 14443A transponder – **MIFARE Desfire, MIFARE PLUS 2k, MIFARE Plus 4k** – to switch, in case of successful operation, from ISO 14443A-3 level to ISO 14443A-4 level. If the 'continuous' mode is enabled, it will be suspended, and the RF field left on, by the execution of this command and will be suspended as long as this command is used; it will be resumed automatically when another type of command will be executed except of the generic ISO 14443A-4 command. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x40	Command code	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i-th byte	$i < n$
1+7	0x..	Transponder UID, n-th byte	n is the UID length

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x40	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x40	Command code	
2	0x02	Status error	

a) if the transponder is present and the command has been successfully executed:

Byte	Value	Description	Notes
1	0x40	Command code	
2	0x00	Status Ok	
3	0x..	RATS, 1 st byte	
...	0x..	RATS, i-th byte	
n+2	0x..	RATS, n-th byte	



For devices with 2 antennas, the command code 0x40 is used to work with the antenna nr 1 while the command code 0xC0 is used to work with antenna nr 2.

2.61 ISO 14443A-4 Transponder 'Generic Command'

This command allows to send any ISO 14443A-4 general format protocol command to a ISO 14443A-4 transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet.

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x41	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x41	Command code	
2	0x02	Status error	

c) if the transponder is present and the command is successfully executed:

Byte	Value	Description	Notes
1	0x41	Command code	
2	0x00	Status Ok	
2+1	0x..	Data received from the tag, 1 st byte	
2+i	0x..	Data received from the tag, i-th byte	i < n

Byte	Value	Description	Notes
2+n	0x..	Data received from the tag, n-th byte	n depends on the specific protocol command response



For devices with 2 antennas, the command code 0x41 is used to work with the antenna nr 1 while the command code 0xC1 is used to work with antenna nr 2.

2.62 ISO 14443B Transponder 'Inventory' Command

This command is used to get the UID code of a ISO 14443B transponder – SR176, SRI512 - that is present near the antenna.

Byte	Value	Description	Notes
1	0x20	Command code	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) case of absence of transponder:

Byte	Value	Description	Notes
1	0x20	Command code	
2	0x01	Status no transponder	

b) case of presence of a transponder:

Byte	Value	Description	Notes
1	0x20	Command code	
2	0x00	Status Ok	
2+1	0x..	Transponder 1 type. See Annex A for tag type and UID length table	
2+1+1	0x..	Transponder 1 UID, 1 st byte	
2+1+i	0x..	Transponder 1 UID, i-th byte	i < n

Byte	Value	Description	Notes
2+1+n	0x..	Transponder 1 UID, n-th byte	n = UID length (See Annex A for tag type and UID length table)
...	
...	0x..	Transponder k type. See Annex A for tag type and UID length table	
...	0x..	Transponder k UID, 1 st byte	
...	0x..	Transponder k UID, j-th byte	j < m
...	0x..	Transponder k UID, m-th byte	m = UID length (See Annex A for tag type and UID length table)



For devices with 2 antennas, the command code 0x20 is used to work with the antenna nr 1 while the command code 0xA0 is used to work with antenna nr 2.

2.63 Read a Data Block of a SR 176 Transponder

This command is used to get a data block (2 bytes) of a known (UID) SR 176 transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x21	Command code	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i th byte	i < 8
1+8	0x..	Transponder UID, 8 th byte	
10	0x..	Block number: • 0x00 ... 0x0F for SR 176	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x21	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x21	Command code	
2	0x02	Status transponder present with errors	

c) if the transponder is present and the command is succesfully executed:

Byte	Value	Description	Notes
1	0x21	Command code	
2	0x00	Status Ok	
3	0x..	Data of the block, 1 st byte	
4	0x..	Data of the block, 2 nd byte	



For devices with 2 antennas, the command code 0x21 is used to work with the antenna nr 1 while the command code 0xA1 is used to work with antenna nr 2.

2.64 Write a data block of a SR 176 transponder

This command is used to write a data block (2 bytes) of a known (UID) SR 176 transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x22	Command code	
1+1	0x..	Transponder UID, 1 st byte	
1+i	0x..	Transponder UID, i th byte	i < 8

Byte	Value	Description	Notes
1+8	0x..	Transponder UID, 8 th byte	
10	0x..	Block number: • 0x00 ... 0x0F for SR 176	
11	0x..	Data to be written in the block, 1 st byte	
12	0x..	Data to be written in the block, 2 nd byte	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) if the transponder is not present:

Byte	Value	Description	Notes
1	0x22	Command code	
2	0x01	Status no transponder	

b) if some error is occurred during the transaction:

Byte	Value	Description	Notes
1	0x22	Command code	
2	0x02	Status transponder present with errors	

c) if the transponder is present and the command is successfully executed:

Byte	Value	Description	Notes
1	0x22	Command code	
2	0x00	Status Ok	



For devices with 2 antennas, the command code 0x22 is used to work with the antenna nr 1 while the command code 0xA2 is used to work with antenna nr 2.

2.65 PicoPass Transponders 'Inventory' Command

This command is used to get the UID code of the identified PicoPass transponders that are present near the antenna/s.

Byte	Value	Description	Notes
1	0x48	Command code	

If the command is not successfully executed, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers,

a) case of absence of transponder:

Byte	Value	Description	Notes
1	0x48	Command code	
2	0x01	Status no transponder	

b) case of presence of transponder/s:

Byte	Value	Description	Notes
1	0x48	Command code	
2	0x00	Status Ok	
2+1	0x..	Transponder 1 UID, 1 st byte	
2+i	0x..	Transponder 1 UID, i-th byte	$i < 8$
2+8	0x..	Transponder 1 UID, 8 th byte	
...	
...	0x..	Transponder n UID, 1 st byte	
...	0x..	Transponder n UID, j-th byte	$j < 8$
2+n*8	0x..	Transponder n UID, 8 th byte	



For devices with 2 antennas, the command code 0x48 is used to work with the antenna nr 1 while the command code 0xC8 is used to work with antenna nr 2.

2.66 ISO 18000-63 Transponder 'Inventory' Command

This command is used to get the list of the ID (variable size) of the identified ISO 18000-63 tags that are present near the antennas. If the command can be executed, the response time is variable and depends upon the number of enabled antennas and the activation time of each one.

Byte	Value	Description	Notes
1	0x18	Command code.	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a1) if at least one tag (m tags) is present and flag for reading antenna disabled

Byte	Value	Description	Notes
1	0x18	Command code.	
2	0x00	Status Ok.	
3	0x..	1st transponder code, 1st byte.	
...	...		
n+2	0x..	1st transponder code, n-th byte.	
...	...		
...	0x..	m-th transponder code, 1st byte.	
...	...		
...	0x..	m-th transponder code, n-th byte.	

a2) if at least one tag is present and flag for reading antenna enabled

Byte	Value	Description	Notes
1	0x18	Command code.	
2	0x00	Status Ok.	
3	0x..	1st transponder code, 1st byte.	
...	...		
n+2	0x..	1st transponder code, n-th byte.	
n+3	0x..	Reading antenna of the 1st tag:	

Byte	Value	Description	Notes
		<ul style="list-style-type: none"> 0x01 -> Antenna 1. 0x02 -> Antenna 2. 0x03 -> Antenna 3. 0x04 -> Antenna 4. 	
...	...		
...	0x..	m-th transponder code, 1st byte.	
...	...		
...	0x..	m-th transponder code, n-th byte.	
...	0x..	Reading antenna of the m-th tag: <ul style="list-style-type: none"> 0x01 -> Antenna 1. 0x02 -> Antenna 2. 0x03 -> Antenna 3. 0x04 -> Antenna 4. 	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x18	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x18	Command code.	
2	0x01	No tag present.	

This command could also be used to get the RSSI of the read transponders:

Byte	Value	Description	Notes
1	0x18	Command code.	
2	0x01		

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a1) if at least one tag (m tags) is present and flag for reading antenna disabled

Byte	Value	Description	Notes
1	0x18	Command code.	
2	0x00	Status Ok.	
3	0x..	1st transponder code, 1st byte.	
...	...		
n+2	0x..	1st transponder code, n-th byte.	
n+3	0x..	The RSSI Q-Channel in dB of the 1st transponder.	
n+4	0x..	The RSSI I-Channel in dB of the 1st transponder.	
...	...		
...	0x..	m-th transponder code, 1st byte.	
...	...		
...	0x..	m-th transponder code, n-th byte.	
...	0x..	The RSSI Q-Channel in dB of the m-th transponder.	
...	0x..	The RSSI I-Channel in dB of the m-th transponder.	

a2) if at least one tag is present and flag for reading antenna enabled

Byte	Value	Description	Notes
1	0x18	Command code.	
2	0x00	Status Ok.	
3	0x..	1st transponder code, 1st byte.	
...	...		
n+2	0x..	1st transponder code, n-th byte.	
n+3	0x..	The RSSI Q-Channel in dB of the 1st transponder.	
n+4	0x..	The RSSI I-Channel in dB of the 1st transponder.	
n+5	0x..	Reading antenna of the 1st tag: <ul style="list-style-type: none"> • 0x01 -> Antenna 1. • 0x02 -> Antenna 2. 	

Byte	Value	Description	Notes
		<ul style="list-style-type: none"> 0x03 -> Antenna 3. 0x04 -> Antenna 4. 	
...	...		
...	0x..	m-th transponder code, 1st byte.	
...	...		
...	0x..	m-th transponder code, n-th byte.	
...	0x..	The RSSI Q-Channel in dB of the m-th transponder.	
...	0x..	The RSSI I-Channel in dB of the m-th transponder.	
...	0x..	Reading antenna of the m-th tag: <ul style="list-style-type: none"> 0x01 -> Antenna 1. 0x02 -> Antenna 2. 0x03 -> Antenna 3. 0x04 -> Antenna 4. 	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x18	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x18	Command code.	
2	0x01	No tag present.	

2.67 Program EPC of an ISO 18000-63 Transponder

This command is used to program the EPC on a known (ID) ISO 18000-63 tag.

Byte	Valore	Descrizione	Note
1	0x1E	Command code.	

Byte	Valore	Descrizione	Note
2	0x..	Transponder code, 1st byte.	
...	...		
n+1	0x..	Transponder code, n-th byte.	
n+2	0x..	Tag access password, 1° byte.	The tag access password is 4 bytes length.
n+3	0x..	1st data block, 1st byte.	Every data block is 2 bytes length.
n+4	0x..	1st data block, 2nd byte.	
...	...		
...	0x..	n-th data block, 1st byte.	
...	0x..	n-th data block, 2nd byte.	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a) if the addressed tag is present and the data bytes have been successfully written

Byte	Valore	Descrizione	Note
1	0x1E	Command code.	
2	0x00	Status Ok.	

b) if the addressed tag do not support the requested blocks or if some error is occurred during the transaction

Byte	Valore	Descrizione	Note
1	0x1E	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Valore	Descrizione	Note
1	0x1E	Command code.	
2	0x01	No tag present.	

2.68 Read Data of an ISO 18000-63 Transponder

This command is used to get data blocks (data block → 2 consecutive bytes) of a known (IDISO 18000-63 tag).

Byte	Valore	Descrizione	Note
1	0x19	Command code.	
2	0x..	Transponder code, 1st byte.	
...	...		
n+1	0x..	Transponder code, n-th byte.	
n+2	0x..	Tag access password, 1° byte.	The tag access password is 4 bytes length.
...	...		
n+5	0x..	Tag access password, 4° byte.	
n+6	0x..	Memory bank: <ul style="list-style-type: none"> • 0x00: Reserved. • 0x01: EPC. • 0x02: TID. • 0x03: User. 	
n+7	0x..	Memory address of the 1st byte of the 1st memory block to read, 1st byte.	The memory address is 4 bytes length.
...	...		
n+10	0x..	Memory address of the 1st byte of the 1st memory block to read, 4th byte.	
n+11	0x..	Number of blocks to read (1 ... 64).	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a) if the addressed tag is present

Byte	Valore	Descrizione	Note
1	0x19	Command code.	
2	0x00	Status Ok.	
3	0x..	1st data block, 1st byte.	Every data block is 2 bytes length.

Byte	Valore	Descrizione	Note
4	0x..	1st data block, 2nd byte.	
...	...		
...	0x..	n-th data block, 1st byte.	
...	0x..	n-th data block, 2nd byte.	

b) if the addressed tag do not support the requested blocks or if some error is occurred during the transaction

Byte	Valore	Descrizione	Note
1	0x19	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Valore	Descrizione	Note
1	0x19	Command code.	
2	0x01	No tag present.	

2.69 Write Data of an ISO 18000-63 Transponder

This command is used to write data on a known (ID) ISO 18000-63 tag.

Byte	Valore	Descrizione	Note
1	0x1A	Command code.	
2	0x..	Transponder code, 1st byte.	
...	...		
n+1	0x..	Transponder code, n-th byte.	
n+2	0x..	Tag access password, 1° byte.	The tag access password is 4 bytes length.
...	...		
n+5	0x..	Tag access password, 4° byte.	
n+6	0x..	Memory bank: <ul style="list-style-type: none"> 0x00: Reserved. 	

Byte	Valore	Descrizione	Note
		<ul style="list-style-type: none"> 0x01: EPC. 0x02: TID. 0x03: User. 	
n+7	0x..	Memory address of the 1st byte of the 1st memory block to write, 1st byte.	The memory address is 4 bytes length.
...	...		
n+10	0x..	Memory address of the 1st byte of the 1st memory block to write, 4th byte.	
n+11	0x..	Number of blocks to write (1 ... 64).	
n+12	0x..	1st data block, 1st byte.	Every data block is 2 bytes length.
n+13	0x..	1st data block, 2nd byte.	
...	...		
...	0x..	n-th data block, 1st byte.	
...	0x..	n-th data block, 2nd byte.	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a) if the addressed tag is present and the data bytes have been successfully written

Byte	Valore	Descrizione	Note
1	0x1A	Command code.	
2	0x00	Status Ok.	

b) if the addressed tag do not support the requested blocks or if some error is occurred during the transaction

Byte	Valore	Descrizione	Note
1	0x1A	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Valore	Descrizione	Note
1	0x1A	Command code.	
2	0x01	No tag present.	

A variation of this command allows to write data on a known (ID) type C tag using the BlockWrite command as defined in the EPC Class-1 Generation-2 standard and not only as a loop of Write commands:

Byte	Valore	Descrizione	Note
1	0x1D	Command code.	
2	0x..	Transponder code, 1st byte.	
...	...		
n+1	0x..	Transponder code, n-th byte.	
n+2	0x..	Tag access password, 1° byte.	The tag access password is 4 bytes length.
...	...		
n+5	0x..	Tag access password, 4° byte.	
n+6	0x..	Memory bank: <ul style="list-style-type: none"> • 0x00: Reserved. • 0x01: EPC. • 0x02: TID. • 0x03: User. 	
n+7	0x..	Memory address of the 1st byte of the 1st memory block to write, 1st byte.	The memory address is 4 bytes length.
...	...		
n+10	0x..	Memory address of the 1st byte of the 1st memory block to write, 4th byte.	
n+11	0x..	Number of blocks to write (1 ... 64).	
n+12	0x..	1st data block, 1st byte.	Every data block is 2 bytes length.
n+13	0x..	1st data block, 2nd byte.	
...	...		
...	0x..	n-th data block, 1st byte.	
...	0x..	n-th data block, 2nd byte.	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a) if the addressed tag is present and the data bytes have been successfully written

Byte	Valore	Descrizione	Note
1	0x1D	Command code.	
2	0x00	Status Ok.	

b) if the addressed tag do not support the requested blocks or if some error is occurred during the transaction

Byte	Valore	Descrizione	Note
1	0x1D	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Valore	Descrizione	Note
1	0x1D	Command code.	
2	0x01	No tag present.	

2.70 Lock Data of an ISO 18000-63 Transponder

This command is used to lock individual password and/or individual memory banks on a known (ID) ISO 18000-63 tag.

Byte	Valore	Descrizione	Note
1	0x1B	Command code.	
2	0x..	Transponder code, 1st byte.	
...	...		
n+1	0x..	Transponder code, n-th byte.	
n+2	0x..	Tag access password, 1° byte.	The tag access password is 4 bytes length.
...	...		

Byte	Valore	Descrizione	Note
n+5	0x..	Tag access password, 4° byte.	
n+6	0x..	Kill password lock property. ASCII character: <ul style="list-style-type: none"> • 0x00: Accessible from all states; • 0x01: Permanently accessible from all states and may never be locked; • 0x02: Accessible only from the secured state; • 0x03: Not accessible from any state; • 0x04: No change. 	
n+7	0x..	Access password lock property. ASCII character: <ul style="list-style-type: none"> • 0x00: Accessible from all states; • 0x01: Permanently accessible from all states and may never be locked; • 0x02: Accessible only from the secured state; • 0x03: Not accessible from any state; • 0x04: No change. 	
n+8	0x..	EPC memory bank lock property. ASCII character: <ul style="list-style-type: none"> • 0x00: Writable from all states; • 0x01: Permanently writable from all states and may never be locked; • 0x02: Writable only from the secured state; • 0x03: Not writable from any state; • 0x04: No change. 	
n+9	0x..	TID memory bank lock property. ASCII character: <ul style="list-style-type: none"> • 0x00: Writable from all states; • 0x01: Permanently writable from all states and may never be locked; • 0x02: Writable only from the secured state; • 0x03: Not writable from any state; • 0x04: No change. 	
n+10	0x..	User memory bank lock property. ASCII character: <ul style="list-style-type: none"> • 0x00: Writable from all states; 	

Byte	Valore	Descrizione	Note
		<ul style="list-style-type: none"> 0x01: Permanently writable from all states and may never be locked; 0x02: Writable only from the secured state; 0x03: Not writable from any state; 0x04: No change. 	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a) if the addressed tag is present and it has been successfully locked

Byte	Valore	Descrizione	Note
1	0x1B	Command code.	
2	0x00	Status Ok.	

b) if some error is occurred during the transaction

Byte	Valore	Descrizione	Note
1	0x1B	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Valore	Descrizione	Note
1	0x1B	Command code.	
2	0x01	No tag present.	

2.71 'Kill' Command of an ISO 18000-63 Transponder

This command is used to kill a known (ID) ISO 18000-63 tag.

Byte	Valore	Descrizione	Note
1	0x1C	Command code.	
2	0x..	Transponder code, 1st byte.	
...	...		

Byte	Valore	Descrizione	Note
n+1	0x..	Transponder code, n-th byte.	
n+2	0x..	Tag kill password, 1° byte.	The tag kill password is 4 bytes length.
...	...		
n+5	0x..	Tag kill password, 4° byte.	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a) if the addressed tag is present and it has been successfully killed

Byte	Valore	Descrizione	Note
1	0x1C	Command code.	
2	0x00	Status Ok.	

b) if some error is occurred during the transaction

Byte	Valore	Descrizione	Note
1	0x1C	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Valore	Descrizione	Note
1	0x1C	Command code.	
2	0x01	No tag present.	

2.72 'QT Read' Command of a Monza 4QT Transponder

This command allows to send a QT read command as described below to an **Impinj Monza 4QT** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x20	Command code	

Byte	Value	Description	Notes
2	0x..	Transponder code, 1st byte	
...	...		
n+1	0x..	Transponder code, n-th byte	
n+2	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
n+5	0x..	Tag access password, 4th byte	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a) if the addressed tag is present and the command has been successfully executed

Byte	Value	Description	Notes
1	0x20	Command code.	
2	0x00	Status success	
3	0x..	QT control field to be written. MSB.	The QT control field is 2 bytes length
4	0x..	QT control field to be written. LSB.	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x20	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x20	Command code.	
2	0x01	Status transponder not present	

2.73 'QT Write' Command of a Monza 4QT Transponder

This command allows to send a QT write command as described below to an **Impinj Monza 4QT** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x21	Command code	
2	0x..	Transponder code, 1st byte	
...	...		
n+1	0x..	Transponder code, n-th byte	
n+2	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
n+5	0x..	Tag access password, 4th byte	
n+6	0x..	The persistence. Indicates whether the QT control is written to non volatile (NVM) or volatile memory: <ul style="list-style-type: none"> 0x00: Write to volatile memory. 0x01: Write to NVM. 	
n+7	0x..	QT control field to be written. MSB.	The QT control field is 2 bytes length
n+8	0x..	QT control field to be written. LSB.	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a) if the addressed tag is present and the command has been successfully executed

Byte	Value	Description	Notes
1	0x21	Command code.	
2	0x00	Status success	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x21	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x21	Command code.	
2	0x01	Status transponder not present	

2.74 'Read Sensor Code' Command of a Magnus Sx Transponder

This command allows to read the sensor code of an **RFMicron Magnus S2** and **S3** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x22	Command code	
2	0x..	The chip code: <ul style="list-style-type: none"> 0x02: Magnus S2. 0x03: Magnus S3. 	
3	0x..	Transponder code, 1st byte	
...	...		
n+2	0x..	Transponder code, n-th byte	
n+3	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
n+6	0x..	Tag access password, 4th byte	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a) if the addressed tag is present and the command has been successfully executed

Byte	Value	Description	Notes
1	0x22	Command code.	
2	0x00	Status success	
3	0x..	The sensor code read from tag. MSB.	The sensor code is 2 bytes length
4	0x..	The sensor code read from tag. LSB.	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x22	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x22	Command code.	
2	0x01	Status transponder not present	

2.75 'Read On-Chip RSSI' Command of a Magnus Sx Transponder

This command allows to read the on-chip RSSI of an **RFMicron Magnus S2** and **S3** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x23	Command code	
2	0x..	The chip code: <ul style="list-style-type: none"> • 0x02: Magnus S2. • 0x03: Magnus S3. 	
3	0x..	Transponder code, 1 st byte	
...	...		
n+2	0x..	Transponder code, n-th byte	

Byte	Value	Description	Notes
n+3	0x..	Tag access password, 1 st byte	The tag access password is 4 bytes length
...	...		
n+6	0x..	Tag access password, 4 th byte	
n+7	0x..	The RSSI threshold match criteria: <ul style="list-style-type: none"> 0x00: Match if code is <= threshold. 0x01: Match if code is > threshold 	
n+8	0x..	The RSSI threshold in the range 0 ... 31.	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a) if the addressed tag is present and the command has been successfully executed

Byte	Value	Description	Notes
1	0x23	Command code.	
2	0x00	Status success	
3	0x..	The on-chip RSSI read from tag.	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x23	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x23	Command code.	
2	0x01	Status transponder not present	

2.76 'Read Temperature Code' Command of a Magnus S3 Transponder

This command allows to read the temperature code of an **RFMicron Magnus S3** transponder. For more details see the specific transponder data sheet.

Byte	Value	Description	Notes
1	0x24	Command code	
2	0x..	The chip code: • 0x03: Magnus S3.	
3	0x..	Transponder code, 1st byte	
...	...		
n+2	0x..	Transponder code, n-th byte	
n+3	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
n+6	0x..	Tag access password, 4th byte	

If the command fails, the answer is given in short form through the 'reply ack' / 'reply nak' flags; otherwise, the **BLUEBOX** answers with:

a) if the addressed tag is present and the command has been successfully executed

Byte	Value	Description	Notes
1	0x24	Command code.	
2	0x00	Status success	
3	0x..	The temperature code read from tag. MSB.	The temperature code is 2 bytes length
4	0x..	The temperature code read from tag. LSB.	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x24	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x24	Command code.	
2	0x01	Status transponder not present	

2.77 'Spontaneous' Message

In 'continuous' mode, in case of tag presence, the **BLUEBOX** shall update the output buffer with the tag code emulating a 'spontaneous' message behavior.

a) for UHF devices in case of at least one ISO 18000-63 (EPC Class-1 Generation-2) transponder present:

Byte	Value	Description	Notes
1	0x..	Transponder type: <ul style="list-style-type: none"> 0x02: ISO 18000-63 (EPC Class-1 Generation-2). 	Optional parameter present only if the tag type information flag in the general parameters is active. See the device user manual for more info.
2	0x..	Transponder code, 1st byte.	
...	...		
n+1	0x..	Transponder code, n-th byte.	
...	0x..	Last seen RSSI Q value in dB of the tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Last seen RSSI I value in dB of the tag.	Optional parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Reading antenna of the identified tag: <ul style="list-style-type: none"> 0x01 -> Antenna 1. 0x02 -> Antenna 2. 0x03 -> Antenna 3. 0x04 -> Antenna 4. 	Optional parameter present only if the reading antenna information flag in the general parameters is active. See the device user manual for more info.



The message is truncated to the max Profibus buffer size, in this case the flag 'more' shall be set to inform the user of the code truncation.

3 Examples

Hereinafter the firmware version reading, the buffer data request and the queue data request commands and 'spontaneous' message usage example.

3.1 Read Firmware Version

This command is used to get the firmware version of the **BLUEBOX**. For the example we suppose:

- IN/OUT buffer size: 16 bytes

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '0' because we suppose that no transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

[illegible]

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

[illegible]

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

[illegible][illegible]

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

At the end of the execution, the 'slave' will answer to the 'master' sending the firmware version.

The message consists of 2 data blocks because the code length is 16 bytes and the data buffer can only support at most 14 bytes.

For the first data packet the 'slave' loads the second byte of the header (length) and the data bytes of the first 14 data bytes of the answer and then sets to '1' the flag 'More' (another data packet will follow) and the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	0																		
T	0	0	0	0	1	1	0	0E	42	42	5F	54	57	4F	5F	55	20	20	20	33	2E	30			

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

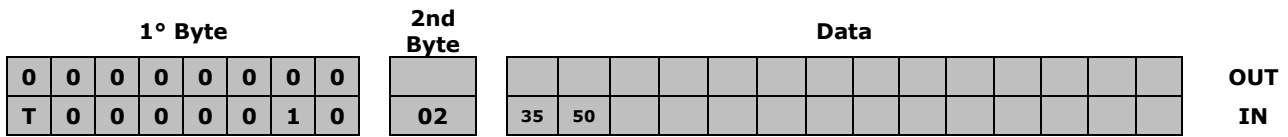
1° Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	1																		
T	0	0	0	0	1	1	0	0E	42	42	5F	54	57	4F	5F	55	20	20	20	33	2E	30			

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

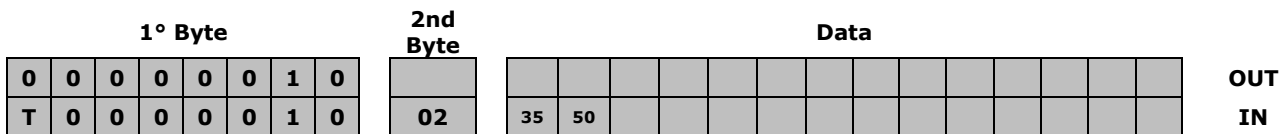
1st Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	0																		
T	0	0	0	0	0	0	1																		

1st Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	0																		
T	0	0	0	0	0	0	0																		

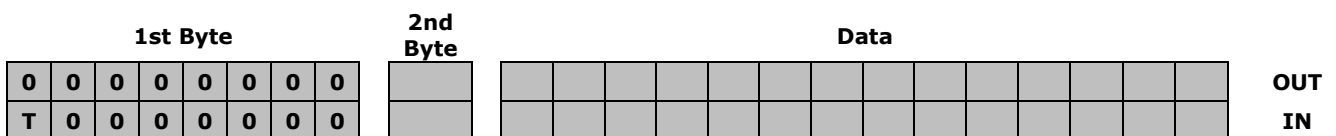
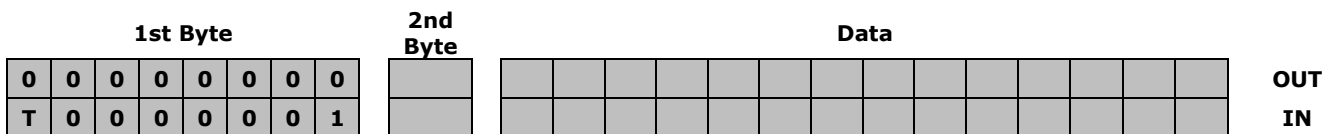
The 'slave' again loads the second byte of the header (length) and the data bytes of the last 6 data bytes of the answer and then resets to '0' the flag 'More' (no other data packet will follow) and the flag 'Req rx' for the 'master'.



The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.



The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').



3.2 Buffer Data Request

This command sends back the code of the eventual transponder that is present in the buffer. When 'continuous' mode is enabled, the reply is immediate because the **BLUEBOX** sends back the data hold in the buffer that is managed by the 'continuous' identification activity; otherwise, the **BLUEBOX** performs readily the identification task under time out protection and sends back the result of the operation. For the example we suppose:

- IN/OUT buffer size: 16 bytes

a) for LF devices with only 1 antenna,

a1) if the antenna have identified a transponder:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																		OUT IN
0	0	0	0	0	0	0	1	01	05																		
T	1	0	0	0	0	0	0																				

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																		OUT IN
0	0	0	0	0	0	0	1	01	05																		
T	1	0	0	0	0	0	1																				

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																		OUT IN
0	0	0	0	0	0	0	0																				
T	1	0	0	0	0	0	1																				

1st Byte								2nd Byte	Data																		OUT IN
0	0	0	0	0	0	0	0																				
T	1	0	0	0	0	0	0																				

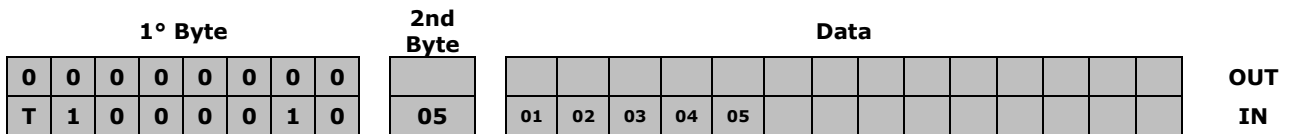
The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																		OUT IN
0	0	0	0	0	0	0	0																				
T	1	0	0	1	0	0	0																				

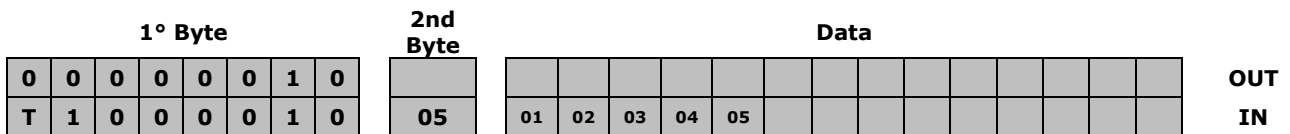
At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder code. For the example we suppose 0x01 0x02 0x03 0x04 0x05.

The message consists of 1 data blocks because the code length is 5 bytes.

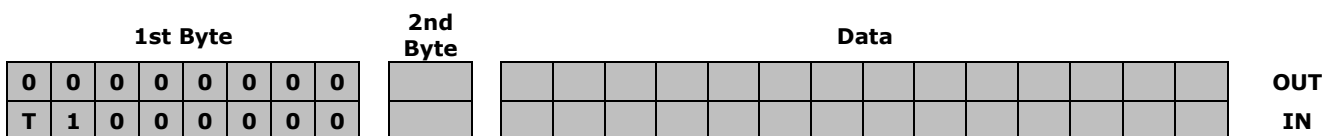
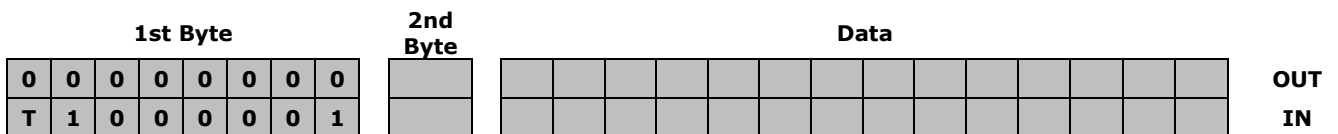
The 'slave' loads the second byte of the header (length) and the data bytes of the 5 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.



The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.



The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

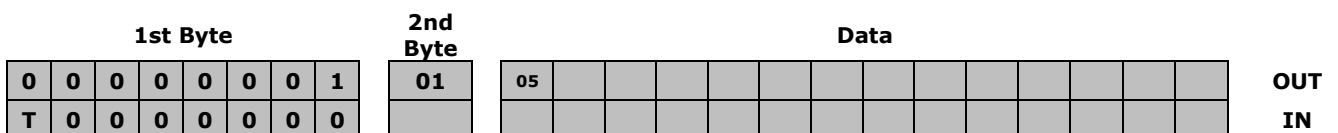


a2) if no transponder is identified by the antenna:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '0' because we suppose that no transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.



The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder code.

The message consists of 1 data blocks because the code length is 5 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 5 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

b) for LF devices with 2 antennas,

b1) if both antennas have identified a transponder:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	1	01	05																
T	1	0	0	0	0	0	0																		

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	1	01	05																
T	1	0	0	0	0	0	1																		

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data												OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder codes. For the example we suppose a tag with code 0x01 0x02 0x03 0x04 0x05 on antenna nr 1 and a tag with code 0x11 0x22 0x33 0x44 0x55 on antenna nr 2.

The message consists of 1 data blocks because the data length is 12 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 12 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data												OUT IN		
0	0	0	0	0	0	0	0																
T	1	0	0	0	0	1	0	0C	05	01	02	03	04	05	05	11	22	33	44	55			

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data												OUT IN			
0	0	0	0	0	0	1	0																	
T	1	0	0	0	0	1	0	0C	05	01	02	03	04	05	05	11	22	33	44	55				

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

1st Byte								2nd Byte	Data																OUT IN			
0	0	0	0	0	0	0	0																					
T	1	0	0	0	0	0	0																					

b2) if only antenna 1 have identified a transponder:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

[illegible]

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

[illegible]

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

[illegible][illegible]

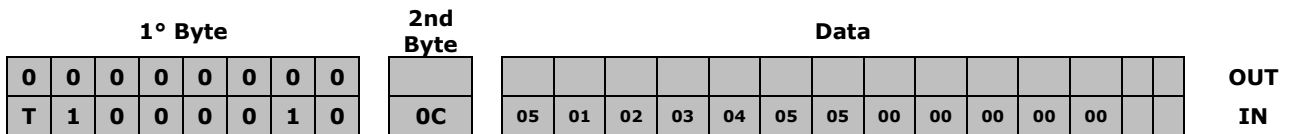
The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

[illegible]

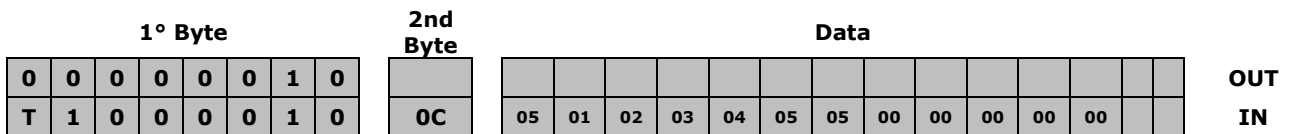
At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder codes. For the example we suppose a tag with code 0x01 0x02 0x03 0x04 0x05 on antenna nr 1.

The message consists of 1 data blocks because the data length is 12 bytes.

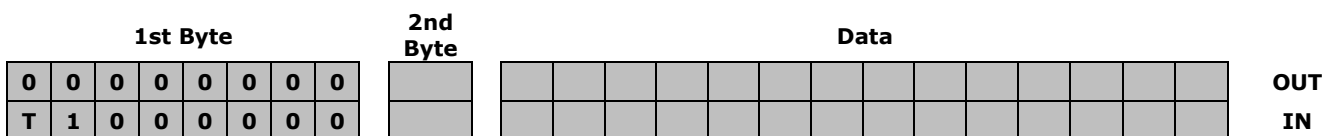
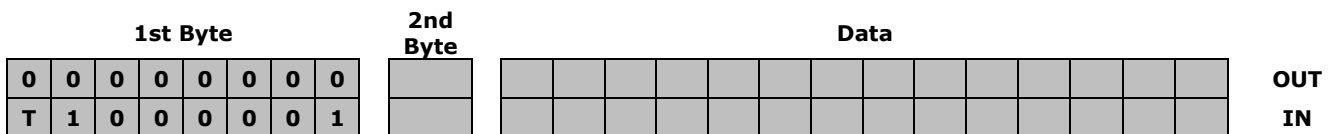
The 'slave' loads the second byte of the header (length) and the data bytes of the 12 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.



The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.



The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

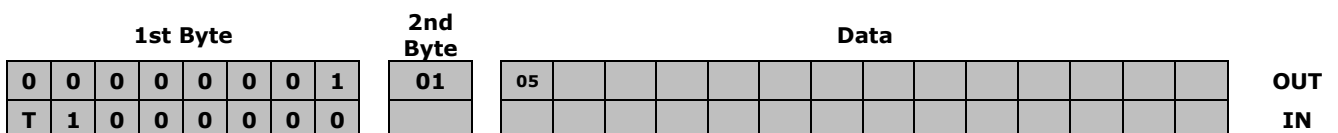


b3) if only antenna 2 have identified a transponder:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.



The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		</

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													</

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder codes. For the example we suppose a tag with code code 0x11 0x22 0x33 0x44 0x55 on antenna nr 2.

The message consists of 1 data blocks because the data length is 12 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 12 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																</

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
0	0	0	0	0	0	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						</

b4) if none of the antennas have identified a transponder:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '0' because we suppose that no transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					</

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					</

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							</

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder codes.

The message consists of 1 data blocks because the data length is 12 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 12 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data														OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										</

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data														OUT IN		
0	0	0	0	0	0	1	0																		
T	0	0	0	0	0	1	0	0C	05	00	00	00	00	00	05	00	00	00	00	00					

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														</

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										</

c) for HF devices with only 1 antenna,

c1) if the antenna have identified a transponder:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															</

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				</

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder code. For the example we suppose a MIFARE 1k (UID=4) with code 0x11 0x22 0x33 0x44.

The message consists of 1 data blocks because the code length is 5 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 5 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	0																		
T	1	0	0	0	0	1	0	05	11	11	22	33	44												

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																OUT IN	
0	0	0	0	0	0	1	0																			
T	1	0	0	0	0	1	0	05	11	11	22	33	44													

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								

c2) if no transponder is identified by the antenna:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '0' because we suppose that no transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																</

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder code.

The message consists of 1 data blocks because the code length is 5 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 5 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	0																		
T	0	0	0	0	0	1	0	05	00	00	00	00	00												

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																OUT IN	
0	0	0	0	0	0	1	0																			
T	0	0	0	0	0	1	0	05	00	00	00	00	00													

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

d) for HF devices with 2 antennas,

d1) if both antennas have identified a transponder:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				</

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										</

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder codes. For the example we suppose a MIFARE 1k (UID=4) tag with code 0x01 0x02 0x03 0x04 on antenna nr 1 and a MIFARE 1k (UID=4) tag with code 0x11 0x22 0x33 0x44 on antenna nr 2.

The message consists of 1 data blocks because the data length is 10 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 10 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																OUT IN	
0	0	0	0	0	0	0	0																			
T	1	0	0	0	0	1	0	0A	11	01	02	03	04	11	11	22	33	44								

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	1	0																		
T	1	0	0	0	0	1	0	0A	11	01	02	03	04	11	11	22	33	44							

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							</

d2) if only antenna 1 have identified a transponder:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			</

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder codes. For the example we suppose a MIFARE 1k (UID = 4) tag with code 0x01 0x02 0x03 0x04 on antenna nr 1.

The message consists of 1 data blocks because the data length is 10 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 10 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data												OUT IN				
0	0	0	0	0	0	0	0																		
T	1	0	0	0	0	1	0	0A	11	01	02	03	04	00	00	00	00	00							

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
0	0	0	0	0	0	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		</

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT IN			
0	0	0	0	0	0	0	0																					
T	1	0	0	0	0	0	1																					

1st Byte								2nd Byte	Data																OUT IN		
0	0	0	0	0	0	0	0																				
T	1	0	0	0	0	0	0																				

d3) if only antenna 2 have identified a transponder:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	1	01	05																
T	1	0	0	0	0	0	0																		

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			</

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder codes. For the example we suppose a MIFARE 1k (UID=4) tag with code code 0x11 0x22 0x33 0x44 on antenna nr 2.

The message consists of 1 data blocks because the data length is 10 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 10 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
0	0	0	0	0	0	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

1st Byte								2nd Byte	Data																OUT	
0	0	0	0	0	0	0	0																			
T	1	0	0	0	0	0	0																			IN

d4) if none of the antennas have identified a transponder:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '0' because we suppose that no transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																</

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									</

At the end of the execution, the 'slave' will answer to the 'master'.

The message consists of 1 data blocks because the data length is 10 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 10 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																OUT IN	
0	0	0	0	0	0	0	0																			
T	0	0	0	0	0	0	1	0	0A	00	00	00	00	00	00	00	00	00	00	00						

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	1	0																		
T	0	0	0	0	0	1	0	0A	00	00	00	00	00	00	00	00	00	00	00						

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										

e) for UHF devices with 1 antenna,

e1) if only one transponder was found, transponder info and antenna info are not active, and the device is not in 'gate' mode:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	1	01	05																
T	1	0	0	0	0	0	0																		

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
0	0	0	0	0	0	0	1	01	05																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									</

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										</

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder codes. For the example we suppose a tag with code 0x30 0x00 0xE2 0x00 0x40 0x80 0x67 0x10 0x01 0x28 0x25 0x30 0x14 0xBF 0x3A 0x61.

The message consists of 2 data blocks because the code length is 16 bytes and the data buffer can only support at most 14 bytes.

For the first data packet the 'slave' loads the second byte of the header (length) and the data bytes of the first 14 data bytes of the answer and then sets to '1' the flag 'More' (another data packet will follow) and the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																		OUT
0	0	0	0	0	0	0	0																				IN
T	0	0	0	0	0	1	1	0	0E	30	00	E2	00	40	80	67	10	01	28	25	30	14	BF				

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																		OUT
0	0	0	0	0	0	0	1	0																			IN
T	0	0	0	0	0	1	1	0	0E	30	00	E2	00	40	80	67	10	01	28	25	30	14	BF				

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

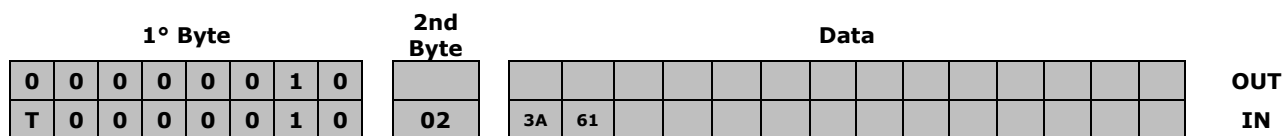
1st Byte								2nd Byte	Data																		OUT
0	0	0	0	0	0	0	0																				IN
T	0	0	0	0	0	0	0	1																			

1st Byte								2nd Byte	Data																		OUT
0	0	0	0	0	0	0	0																				IN
T	0	0	0	0	0	0	0	0																			

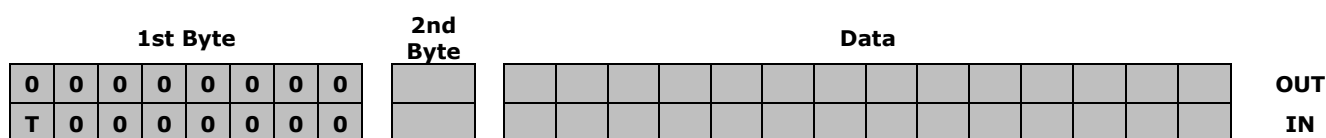
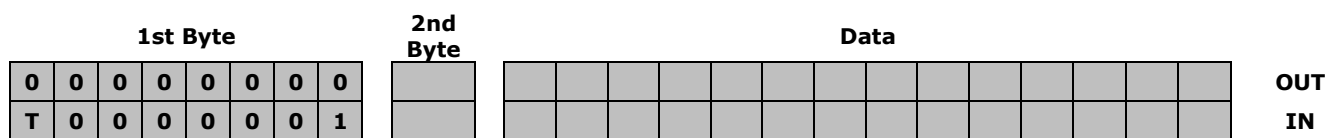
The 'slave' again loads the second byte of the header (length) and the data bytes of the last 2 data bytes of the answer and then resets to '0' the flag 'More' (no other data packet will follow) and the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																		OUT
0	0	0	0	0	0	0	0																				IN
T	0	0	0	0	0	0	1	0	02	3A	61																

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.



The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').



3.3 Queue Data Request

In 'continuous' mode, when the **BLUEBOX** finds a 'new' transponder, it inserts its code in the FIFO queue. This command sends back the first present code in the queue. For the example we suppose:

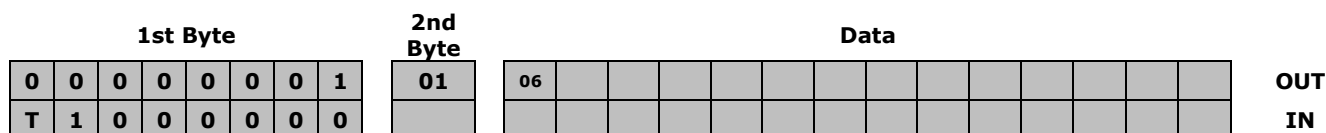
- IN/OUT buffer size: 16 bytes

a) for LF devices with only 1 antenna:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.



The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.



0	0	0	0	0	0	0	1
T	1	0	0	0	0	0	1

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte							
0	0	0	0	0	0	0	0
T	1	0	0	0	0	0	1

1st Byte							
0	0	0	0	0	0	0	0
T	1	0	0	0	0	0	0

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte							
0	0	0	0	0	0	0	0
T	1	0	0	1	0	0	0

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder code. For the example we suppose a tag with code 0x01 0x02 0x03 0x04 0x05.

The message consists of 1 data blocks because the code length is 5 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 5 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte							
0	0	0	0	0	0	0	0
T	1	0	0	0	0	1	0

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte							
0	0	0	0	0	0	1	0
T	1	0	0	0	0	1	0

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte

0	0	0	0	0	0	0	0											OUT IN
T	1	0	0	0	0	0	1											

[illegible]

b) for LF devices with 2 antennas:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

[illegible]

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

[illegible]

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

[illegible][illegible]

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										</

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder code. For the example we suppose a tag with code 0x01 0x02 0x03 0x04 0x05 identified by antenna nr 1.

The message consists of 1 data blocks because the code length is 6 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 6 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
0	0	0	0	0	0	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										</

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											</

c) for HF devices with only 1 antenna:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
0	0	0	0	0	0	0	1	01	06																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										</

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	1	01	06																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										</

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder code. For the example we suppose a MIFARE 1k (UID=4) tag with code 0x01 0x02 0x03 0x04.

The message consists of 1 data blocks because the code length is 5 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 5 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	0																		
T	1	0	0	0	0	1	0	05	11	01	02	03	04												

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data												OUT IN			
0	0	0	0	0	0	1	0																	
T	1	0	0	0	0	1	0	05	11	01	02	03	04											

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

[illegible][illegible]

d) for HF devices with 2 antennas:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

[illegible]

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	1	01	06																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

1st Byte								2nd Byte	Data																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder code. For the example we suppose a tag with code 0x01 0x02 0x03 0x04 identified by antenna nr 1.

The message consists of 1 data blocks because the code length is 6 bytes.

The 'slave' loads the second byte of the header (length) and the data bytes of the 6 data bytes of the answer and then sets to '1' the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																OUT IN
0	0	0	0	0	0	0	0																		
T	1	0	0	0	0	1	0	06	11	01	02	03	04	01											

The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																OUT IN		
0	0	0	0	0	0	1	0																				
T	1	0	0	0	0	1	0	06	11	01	02	03	04	01													

The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										

e) for UHF devices with 1 antenna,

e1) if transponder info and antenna info are not active, and the device is not in 'gate' mode:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
0	0	0	0	0	0	0	1	01	06																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										</

The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0	0	0	0	0	0	0	1	01	06																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

1st Byte								2nd Byte	Data																OUT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													</

1st Byte								2nd Byte	Data																OUT
0	0	0	0	0	0	0	0																		

[illegible]

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

[illegible]

At the end of the execution, the 'slave' will answer to the 'master' sending the read transponder codes. For the example we suppose a tag with code 0x30 0x00 0xE2 0x00 0x40 0x80 0x67 0x10 0x01 0x28 0x25 0x30 0x14 0xBF 0x3A 0x61.

The message consists of 2 data blocks because the code length is 16 bytes and the data buffer can only support at most 14 bytes.

For the first data packet the 'slave' loads the second byte of the header (length) and the data bytes of the first 14 data bytes of the answer and then sets to '1' the flag 'More' (another data packet will follow) and the flag 'Req rx' for the 'master'.

1° Byte								2nd Byte	Data																	
0	0	0	0	0	0	0	0																			OUT
T	0	0	0	0	0	1	1	0E	30	00	E2	00	40	80	67	10	01	28	25	30	14	BF			IN	

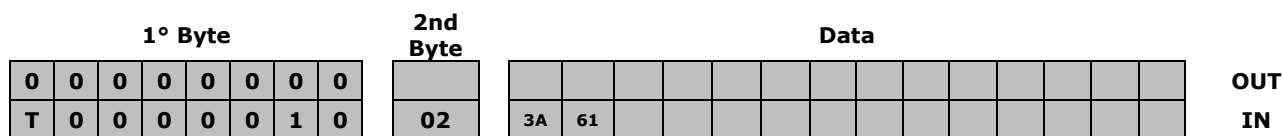
The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.

1° Byte								2nd Byte	Data																	
0	0	0	0	0	0	1	0																			OUT
T	0	0	0	0	1	1	0	0E	30	00	E2	00	40	80	67	10	01	28	25	30	14	BF				IN

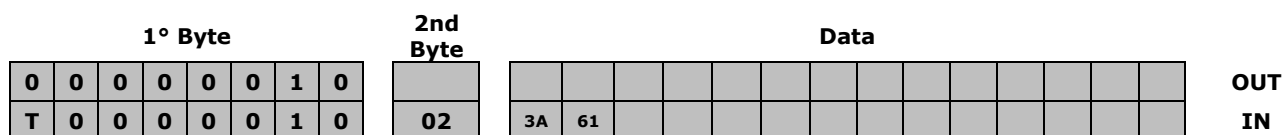
The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

[illegible][illegible]

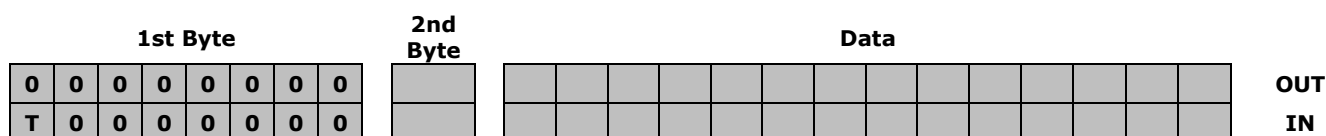
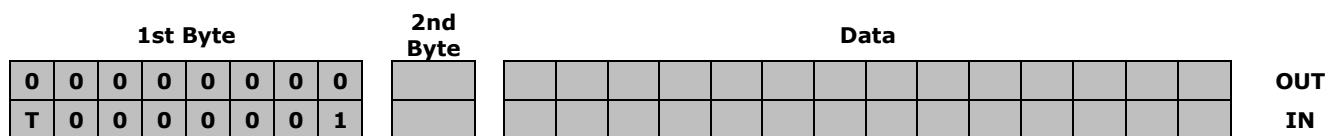
The 'slave' again loads the second byte of the header (length) and the data bytes of the last 2 data bytes of the answer and then resets to '0' the flag 'More' (no other data packet will follow) and the flag 'Req rx' for the 'master'.



The 'master' confirms the reception of the data packet setting to '1' the flag 'Ack rx'.



The handshake will terminate by resetting to '0' the flag 'Req rx' (by the 'slave') and the flag 'Ack rx' (by the 'master').

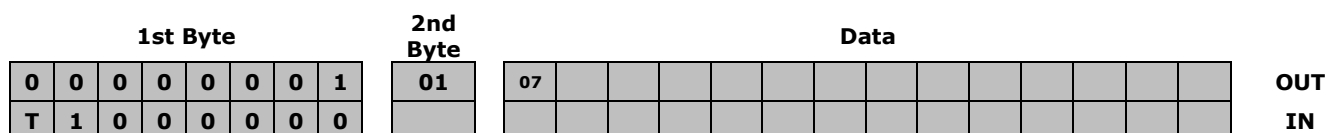


To delete the received code from the queue, the 'master' reply to the **BLUEBOX** with:

The bit 7 of the first byte of the header of the incoming buffer of the 'master' device will be indicate as 'T' (toggle) because we suppose that the 'slave' is running correctly, the bit 6 of the same byte is set to '1' because we suppose that a transponder is present near the antenna.

In the following the sequence of the composition of the outgoing and incoming buffer of the 'master' device will be graphically depicted.

For the transmission of this first data package the 'master' loads the second byte of the header (length) and the data bytes of the command (in this case only one) and then sets to '1' the flag 'Req tx' for the 'slave'.



The 'slave' confirms the reception of the data packet setting to '1' the flag 'Ack tx'.



[illegible]

Consequently the 'master' will reset to '0' the flag 'Req tx' and then also the 'slave' will reset to '0' the flag 'Ack tx'.

[illegible][illegible]

The 'slave' will set to '1' the flag 'Busy' during the execution of the received command.

[illegible]

At the end of the execution, the 'slave' will answer to the 'master' in a short form using ack/nak flags. For the example we suppose an ack reply.

[illegible]

4 Document Revision History

Date	Revision	Description
30/08/16	1.00	Initial release.
01/09/16	1.01	<p>Added 5223L, 5233L and 5243L readers support to this manual.</p> <p>Added 5223U-S, 5239U-S and 5233U readers support to this manual.</p> <p>Added the 'Reading Test Activation/Deactivation' command description (section 2.1).</p> <p>Added LF readers support to 'Data Request' and 'Queue Data Request' commands (sections 2.6 and 2.7).</p> <p>Added LF transponders management commands (sections 2.8 to 2.25).</p> <p>Added LF readers support to 'Spontaneous' message description (section 2.50).</p> <p>Added LF readers support to commands examples (sections 3.2, 3.3 and 3.4).</p> <p>Added UHF readers support to 'Data Request' and 'Queue Data Request' commands (sections 2.6 and 2.7).</p> <p>Added UHF transponders management commands (sections 2.44 to 2.49).</p> <p>Added UHF readers support to 'Spontaneous' message description (section 2.50).</p> <p>Added UHF readers support to commands examples (sections 3.2, 3.3 and 3.4).</p> <p>Added LF supported transponders as appendix A.</p> <p>Added UHF supported transponders as appendix C.</p> <p>Added LF transponders nibble coding descriptions as appendix D.</p> <p>Updated the supported commands table in appendix E.</p>
30/09/16	1.02	<p>Updated the 5223U-S, 5239U-S and 5233U firmware versions object of this manual.</p> <p>Added 'Antennas Auto-Tuning' (section 2.13).</p> <p>Added 'RF Power Test' command (section 2.18).</p>
23/12/16	1.03	Deleted supported tags appendixes.
08/02/18	1.04	<p>Updated reader's firmware versions object of this manual.</p> <p>Corrected all the read and write commands titles.</p> <p>Added the variable size (max 240 bytes) null terminated string parameters management in 'Read Configuration</p>

Date	Revision	Description
		<p>Parameters' and 'Write Configuration Parameters' commands.</p> <p>Added 'Read ID Code of a HITAG 2 Transponder', 'Read a Page of a HITAG 2 Transponder' and 'Write a Page of a HITAG 2 Transponder' commands.</p> <p>Added 'ISO 14443A-4 Transponder RATS Command' and 'ISO 14443A-4 Transponder Generic Command' commands.</p> <p>Added the PicoPass transponder 'Inventory' command.</p> <p>Added the examples section.</p> <p>Updated the supported commands tables.</p>
01/08/18	1.05	<p>Updated reader's firmware versions object of this manual.</p> <p>Added the 5239U-C reader support to this manual.</p> <p>Added the RSSI Q and I channel info in 'Buffer Data Request' and 'Queue Data Request' commands and 'Spontaneous Message'.</p> <p>Corrected tag type, antenna info and gate crossing direction values in 'Buffer Data Request' and 'Queue Data Request' commands and 'Spontaneous Message'.</p> <p>Corrected antenna info values in 'Inventory ISO 18000-63' commands.</p> <p>Added the 'RF Sensitivity' Test, Read Reflected Power and Read RSSI Power commands.</p> <p>Added 'ISO 14443A-4 Transponder RATS Command' and 'ISO 14443A-4 Transponder Generic Command' commands.</p> <p>Added the 'QT Read' and 'QT Write' commands of an Impinj Monza 4QT transponder.</p> <p>Added the 'Read Sensor Code' and 'Read On-Chip RSSI' commands of an RFMicron Magnus S2 and S3 transponders.</p> <p>Added the 'Read Temperature Code' of an RFMicron Magnus S3.</p>
04/05/20	1.06	<p>Updated the company name/logo and BLUEBOX logo.</p> <p>Updated the reader's description object of this manual.</p> <p>Replaced ISO 18000-6C with ISO 18000-63. They are the same standard, 18000-6C became 18000-63 in 2012 due to ISO naming rules that do not allow letters in standards names.</p> <p>Format changes and document fixes in all sections.</p>

A. Supported Commands Table

	5223L, 5233L, 5243L	5223H, 5233H, 5243H	5223U-S, 5239U-S, 5233U
Device Reset	✓	✓	✓
Read Device Serial Number	✓	✓	✓
Read Firmware Version	✓	✓	✓
Read Temperature	✓	✓	✓
Read Date/Time	✓	✓	✓
Write Date/Time	✓	✓	✓
Write General Parameters	✓	✓	✓
Write Configuration Parameters	✓	✓	✓
Set Default Parameters	✓	✓	✓
Read General Parameters	✓	✓	✓
Read Configuration Parameters	✓	✓	✓
'RF Reading' Test			✓
'RF Power' Test			✓
'RF Sensitivity' Test			✓
Read Reflected Power			✓
Read RSSI Power			✓
Digital Output Activation	✓	✓	✓
Read Device Status	✓	✓	✓
RF Deactivation	✓	✓	✓
RF Activation	✓	✓	✓
Antennas Auto-Tuning			✓
Buffer Data Request	✓	✓	✓
Queue Data Request	✓	✓	✓
Write Data to an EM4305 Transponder	✓		
Read ID Code of an EM4305 Transponder	✓		
Write Data to a T5557 Transponder	✓		
Read ID Code of a T5557 Transponder	✓		
Write Data to a Q5 Transponder	✓		
Read ID Code of a Q5 Transponder	✓		
Write Data to an HITAG S Transponder	✓		
Read ID Code of an HITAG 1 / HITAG S Transponder	✓		
Read a Page of an HITAG 1 / HITAG S Transponder	✓		
Write a Page of an HITAG 1 / HITAG S Transponder	✓		
Read ID Code of an HITAG 2 Transponder	✓		

	5223L, 5233L, 5243L	5223H, 5233H, 5243H	5223U-S, 5239U-C, 5233U
Read a Page of an HITAG 2 Transponder	✓		
Write a Page of an HITAG 2 Transponder	✓		
'Reset' Command for a TITAN Transponder	✓		
'Login' Command for a TITAN Transponder	✓		
'Write Password' Command for a TITAN Transponder	✓		
'Standard Read' Command for a TITAN Transponder	✓		
'Selective Read' Command for a TITAN Transponder	✓		
'Write Word' Command for a TITAN Transponder	✓		
'Write Several Words' Command for a TITAN Transponder	✓		
'Read After Write Word' Command for a TITAN Transponder	✓		
ISO 15693 Transponders 'Inventory' Command		✓	
Read a Data Block of an ISO 15693 Transponder		✓	
Write a Data Block of an ISO 15693 Transponder		✓	
Lock a Data Block of an ISO 15693 Transponder		✓	
ISO 15693 Transponder 'Get System Info' Command		✓	
ISO 15693 Transponder 'General Protocol' Command		✓	
ISO 14443A Transponders 'Inventory' Command		✓	
Read a Data Block of a MIFARE 1k/4k (UID 4) Transponder		✓	
Write a Data Block of a MIFARE 1k/4k (UID 4) Transponder		✓	
Read a Data Block of a MIFARE 1k/4k (UID 7) Transponder		✓	
Write a Data Block of a MIFARE 1k/4k (UID 7) Transponder		✓	
Read a Data Page of a MIFARE Ultralight Transponder		✓	
Write a Data Page of a MIFARE Ultralight Transponder		✓	
Read a Data Page of a NTAG213/215/216 Transponder		✓	
Write a Data Page of a NTAG213/215/216 Transponder		✓	
ISO 14443A-4 Transponder RATS Command		✓	
ISO 14443A-4 Transponder Generic Command		✓	
ISO 14443B Transponders 'Inventory' Command		✓	
Read a Data Block of a SR 176 Transponder		✓	
Write a Data Block of a SR176 Transponder		✓	
PicoPass Transponders 'Inventory' Command		✓	
ISO 18000-63 Transponder 'Inventory' Command			✓
Program EPC of an ISO 18000-63 Transponder			✓
Read Data of an ISO 18000-63 Transponder			✓
Write Data of an ISO 18000-63 Transponder			✓
Lock Data of an ISO 18000-63 Transponder			✓
'Kill' Command of an ISO 18000-63 Transponder			✓

	5223L, 5233L, 5243L	5223H, 5233H, 5243H	5223U-S, 5239U-C, 5233U
'QT Read' Command of an Impinj Monza 4QT Transponder			✓
'QT Write' Command of an Impinj Monza 4QT Transponder			✓
'Read Sensor Code' Command of an RFMicron Magnus S2 / S3 Transponder			✓
'Read On-Chip RSSI' Command of an RFMicron Magnus S2 / S3 Transponder			✓
'Read Temperature Code' Command of an RFMicron Magnus S3 Transponder			✓
'Spontaneous' Message			✓