



Bluebox LF, HF, UHF Communication Protocol Serial / Ethernet

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

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

Issue 1.29
– 05. August 2025 –

Subject to alteration without prior notice.
© Copyright iDTRONIC GmbH 2025
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 always kept available 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, omissions in the manual, 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 with firmware version as of

OEM Reader LF		
Order Number	Hardware Version	Firmware Version
1021L	3	3.17d
1031L	3	3.17d
1041L	3	3.17d
1021L	4	4.06d
1031L	4	4.06d

OEM Reader HF		
Order Number	Hardware Version	Firmware Version
1021H	1	1.19
1031H	1	1.19
1041H	1	1.19
1051H	1	1.19
1021H	2	2.07
1031H	2	2.07

OEM Reader NFC		
Order Number	Hardware Version	Firmware Version
1021N	1	1.11
1031N	1	1.11

OEM Reader UHF		
Order Number	Hardware Version	Firmware Version
1021U	1	1.54
1021U-FCC	1	1.54
1021U-BRA	1	1.54
1021U-S	1	1.54
1021U-S-FCC	1	1.54
1021U-S-BRA	1	1.54
1031U	1	1.54
1031U-FCC	1	1.54

OEM Reader UHF

Order Number	Hardware Version	Firmware Version
1031U-BRA	1	1.54
1031U-S	1	1.54
1031U-S-FCC	1	1.54
1031U-S-BRA	1	1.54
1061U	1	1.54M
1061U-FCC	1	1.54M
1061U-BRA	1	1.54M
1061U-S	1	1.54M
1061U-S-FCC	1	1.54M
1061U-S-BRA	1	1.54M
1071U	1	1.54M
1071U-FCC	1	1.54M
1071U-BRA	1	1.54M
1071U-S	1	1.54M
1071U-S-FCC	1	1.54M
1071U-S-BRA	1	1.54M
1041U	2	2.54Q
1041U-FCC	2	2.54Q
1051U	2	2.54Q
1051U-FCC	2	2.54Q
1051U-BRA	2	2.54Q

Desktop Reader LF

Order Number	Hardware Version	Firmware Version
3122L	2 + 3	2.31a + 3.17d
3122L	2 + 4	2.31a + 4.06d

Desktop Reader HF

Order Number	Hardware Version	Firmware Version
3122H	1 + 2	1.23 + 2.31a
3122H	2	2.31a + 2.07

Desktop Reader NFC

Order Number	Hardware Version	Firmware Version
3122N	1 + 2	1.11 + 2.31a

Desktop Reader UHF

Order Number	Hardware Version	Firmware Version
3122U	1 + 2	1.54 + 2.31a

Panel Reader LF

Order Number	Hardware Version	Firmware Version
3221L	3	3.17d
3121L	4	4.06d
3222L	3	3.17d
3222L	4	4.06d

Panel Reader LF

Order Number	Hardware Version	Firmware Version
3223L	1 + 3	1.00 + 3.17d
3223L	1 + 4	1.00 + 4.06d

Panel Reader NFC

Order Number	Hardware Version	Firmware Version
3221N	1	1.11
3221N	2	2.07
3222N	1	1.11
3222N	2	2.07
3223N	1	1.00 + 1.11
3223N	1 + 2	1.00 + 2.07

Basic Reader LF

Order Number	Hardware Version	Firmware Version
5121L	2 + 3	2.21 + 3.17d
5121L	2 + 4	2.21 + 4.06d
5131L	2 + 3	2.21 + 3.17d
5131L	2 + 4	2.21 + 4.06d

Basic Reader HF

Order Number	Hardware Version	Firmware Version
5121H	1 + 2	1.23 + 2.21
5121H	2	2.07 + 2.21
5131H	1 + 2	1.23 + 2.21
5131H	2	2.07 + 2.21

Basic Reader UHF

Order Number	Hardware Version	Firmware Version
5121U	1 + 2	1.54 + 2.21
5131U	1 + 2	1.54 + 2.21

Advant Reader LF

Order Number	Hardware Version	Firmware Version
5221L	1 + 3	2.01 + 3.17d
5222L	1 + 3	2.01 + 3.17d
5231L	1 + 3	2.01 + 3.17d
5232L	1 + 3	2.01 + 3.17d
5241L	1 + 3	2.01 + 3.17d
5242L	1 + 3	2.01 + 3.17d

Advant Reader HF

Order Number	Hardware Version	Firmware Version
5221H	1	1.23 + 2.10
5222H	1	1.23 + 2.10
5231H	1	1.23 + 2.10

Advant Reader HF

Order Number	Hardware Version	Firmware Version
5232H	1	1.23 + 2.10
5241H	1	1.23 + 2.10
5242H	1	1.23 + 2.10

Advant Reader UHF

Order Number	Hardware Version	Firmware Version
5221U-S	1 + 2	1.45M + 3.10
5222U-S	1 + 2	1.45M + 3.10
5237U-S	1 + 2	1.45M + 3.10
5238U-S	1 + 2	1.45M + 3.10
5231U	2	2.10 + 2.45Q
5232U	2	2.10 + 2.45Q
5224U	1	1.54
5224U	2	2.54
5225U	1	1.54
5225U	2	2.54

CX Reader UHF

Order Number	Hardware Version	Firmware Version
5325U	2	2.67B
5325U-RTC	2	2.67T
5325U-FCC	2	2.67B
5325U-RTC-FCC	2	2.67T
5335U	2	2.67A
5335U-RTC	2	2.67S
5345U	2	2.67
5345U-RTC	2	2.67R
5326U	2	2.67B
5326U-RTC	2	2.67T
5326U-FCC	2	2.67B
5326U-RTC-FCC	2	2.67T
5336U	2	2.67A
5336U-RTC	2	2.67S
5346U	2	2.67
5346U-RTC	2	2.67R
5346U-BL	2	2.67L
5327U	2	2.67B
5327U-FCC	2	2.67B
5337U	2	2.67A
5347U	2	2.67
5328U	2	2.67B
5328U-RTC	2	2.67T
5328U-FCC	2	2.67B
5328U-RTC-FCC	2	2.67T
5338U	2	2.67A
5338U-RTC	2	2.67S

CX Reader UHF

Order Number	Hardware Version	Firmware Version
5348U	2	2.67
5348U-RTC	2	2.67R

CX E Reader UHF

Order Number	Hardware Version	Firmware Version
5426U	2	2.67E
5426U-RTC	2	2.67U
5427U	2	2.67E
5428U	2	2.67E
5428U-RTC	2	2.67U
5426U-G	2	2.67E
5426U-RTC-G	2	2.67U
5427U-G	2	2.67E
5428U-G	2	2.67E
5428U-RTC-G	2	2.67U
5526U	2	2.67E
5526U-RTC	2	2.67U
5527U	2	2.67E
5528U	2	2.67E
5528U-RTC	2	2.67U
5526U-G	2	2.67E
5526U-RTC-G	2	2.67U
5527U-G	2	2.67E
5528U-G	2	2.67E
5528U-RTC-G	2	2.67U

RTU Reader UHF

Order Number	Hardware Version	Firmware Version
5721U	1	1.54M

Contents

	Safety Instructions / Warning – Read before start-up.....	3
1	Introduction	11
2	Protocol Specifications	12
2.1	Device Reset	13
2.2	Read Device Serial Number	13
2.3	Read Ethernet MAC Address	13
2.4	Read Bluetooth MAC Address	14
2.5	Read Firmware Version	14
2.6	Firmware Upgrade	15
2.7	Read Temperature	15
2.8	Read Date/Time	16
2.9	Write Date/Time	16
2.10	Write ROM General Parameters.....	17
2.11	Write RAM Configuration Parameters.....	17
2.12	Write ROM Configuration Parameters	18
2.13	Write ROM Default Parameters	18
2.14	Read RAM General Parameters.....	19
2.15	Read RAM Configuration Parameters.....	19
2.16	Read ROM Configuration Parameters	19
2.17	Sleep Mode	20
2.18	“RF Reading” Test	20
2.19	“RF Power” Test	21
2.20	“RF Sensitivity” Test	21
2.21	Read Reflected Power	22
2.22	Read RSSI Power	23
2.23	Digital Output Activation	23
2.24	Status Reading	24
2.25	RF Deactivation	24
2.26	RF Activation	24
2.27	Antennas Auto-Tuning	25
2.28	Buffer Data Request.....	25
2.29	Queue Data Request	29
2.30	Read Number of Records	31
2.31	Reset Record Database	31
2.32	Read Current Record	31
2.33	Dequeue Current Record.....	32
2.34	Re-Read an Dequeued Record.....	32
2.35	Start Continuous Read Records	33
2.36	Stop Continuous Read Records	34
2.37	Write Data to an EM4305 Transponder	34
2.38	Read ID Code of an EM4305 Transponder	35
2.39	Write Data to a T5557 Transponder	35
2.40	Read ID Code of a T5557 Transponder	36
2.41	Write Data to a Q5 Transponder	36
2.42	Read ID Code of a Q5 Transponder	37
2.43	Write Data to a HITAG S Transponder	37
2.44	Read ID Code of a HITAG 1 / HITAG S Transponder	38
2.45	Read a Page of a HITAG 1 / HITAG S Transponder	38

2.46	Write a Page of a HITAG 1 / HITAG S Transponder.....	39
2.47	Read ID Code of a HITAG 2 Transponder.....	40
2.48	Read a Page of a HITAG 2 Transponder	40
2.49	Write a Page of a HITAG 2 Transponder	41
2.50	“Reset” Command for TITAN Transponder	41
2.51	“Login” Command for TITAN Transponder	42
2.52	“Write Password” Command for TITAN Transponder	42
2.53	“Standard Read” Command for TITAN Transponder	43
2.54	“Selective Read” Command for TITAN Transponder	44
2.55	“Write Word” Command for TITAN Transponder	44
2.56	“Write Several Words” Command for TITAN Transponder	45
2.57	“Read After Write Word” Command for TITAN Transponder.....	46
2.58	ISO 15693 Transponders “Inventory” Command	46
2.59	Read a Data Block of an ISO 15693 Transponder	47
2.60	Write a Data Block of an ISO 15693 Transponder	48
2.61	Lock a Data Block of an ISO 15693 Transponder	48
2.62	ISO 15693 Transponder “Get System Info” Command.....	49
2.63	ISO 15693 Transponder “General Protocol” Command.....	50
2.64	ISO 14443A Transponder “Inventory” Command.....	51
2.65	Read a Data Block of a MIFARE 1k/4k (UID 4) Transponder	51
2.66	Write a Data Block of a MIFARE 1k/4k (UID 4) Transponder	52
2.67	Read a Data Block of a MIFARE 1k/4k (UID 7) Transponder	53
2.68	Write a Data Block of a MIFARE 1k/4k (UID 7) Transponder	54
2.69	Read a Data Page of a MIFARE Ultralight Transponder	54
2.70	Write a Data Page of a MIFARE Ultralight Transponder	55
2.71	Read a Data Page of a NTAG213/215/216 Transponder	56
2.72	Write a Data Page of a NTAG213/215/216 Transponder	56
2.73	ISO 14443A-4 Transponder “RATS” Command.....	57
2.74	ISO 14443A-4 Transponder “Generic Command”	58
2.75	MIFARE DESFire Transponder “Generic Command”	58
2.75.1	MIFARE DESFire Transponder “Authenticate” Command	59
2.75.2	MIFARE DESFire Transponder “AuthenticateISO” Command	60
2.75.3	MIFARE DESFire Transponder “AuthenticateAES” Command	61
2.75.4	MIFARE DESFire Transponder “FreeMem” Command	61
2.75.5	MIFARE DESFire Transponder “Format” Command	62
2.75.6	MIFARE DESFire Transponder “GetVersion” Command	62
2.75.7	MIFARE DESFire Transponder “ChangeKey” Command	63
2.75.8	MIFARE DESFire Transponder “ChangeKeySettings” Command	64
2.75.9	MIFARE DESFire Transponder “CreateApplication” Command	64
2.75.10	MIFARE DESFire Transponder “DeleteApplication” Command	65
2.75.11	MIFARE DESFire Transponder “SelectApplication” Command	66
2.75.12	MIFARE DESFire Transponder “CreateStdDataFile” Command	66
2.75.13	MIFARE DESFire Transponder “CreateBackupDataFile” Command.....	67
2.75.14	MIFARE DESFire Transponder “CreateValueFile” Command.....	68
2.75.15	MIFARE DESFire Transponder “DeleteFile” Command	68
2.75.16	MIFARE DESFire Transponder “ReadData” Command	69
2.75.17	MIFARE DESFire Transponder “WriteData” Command	70
2.75.18	MIFARE DESFire Transponder “GetValue” Command	71
2.75.19	MIFARE DESFire Transponder “Credit” Command	71
2.75.20	MIFARE DESFire Transponder “LimitedCredit” Command	72

2.75.21	MIFARE DESFire Transponder “Debit” Command	73
2.75.22	MIFARE DESFire Transponder “CommitTransaction” Command	73
2.75.23	MIFARE DESFire Transponder “AbortTransaction” Command	74
2.76	ISO 14443B Transponder “Inventory” Command	74
2.77	Read a Data Block of a SR 176 Transponder	75
2.78	Write a Data Block of a SR 176 Transponder	75
2.79	PicoPass Transponders “Inventory” Command	76
2.80	ISO 18000-63 Transponder “Inventory” Command	77
2.81	Program EPC of an ISO 18000-63 Transponder	78
2.82	Read Data of an ISO 18000-63 Transponder	79
2.83	Write Data of an ISO 18000-63 Transponder	80
2.84	Lock Data of an ISO 18000-63 Transponder	81
2.85	“Kill” Command of an ISO 18000-63 Transponder	83
2.86	“QT Read” Command of a Monza 4QT Transponder	83
2.87	“QT Write” Command of a Monza 4QT Transponder	84
2.88	“Read Sensor Code” Command of a Magnus Sx Transponder	85
2.89	“Read On-Chip RSSI” Command of a Magnus Sx Transponder	85
2.90	“Read Temperature Code” Command of a Magnus S3 Transponder	86
2.91	“Spontaneous” Message	87
3	Examples	90
3.1	Read Firmware Version	90
3.2	Buffer Data Request	91
3.3	Queue Data Request	96
3.4	“Spontaneous” Message	99
4	Getting Started with C	102
4.1	Command / Reply Checksum	102
4.2	“Spontaneous” Message Checksum	102
4.3	Command / Reply Management	102
5	Document Revision History	107
6	Supported Commands Table	111
6.1	OEM Devices	111
6.2	Desktop Devices	114
6.3	Industrial Devices	117

1 Introduction

This document describes the message format of the serial / ethernet communication protocol used by the host and the Bluebox to issue commands and reply with responses.

2 Protocol Specifications

The “master/slave” serial / ethernet protocol expects that the Bluebox (as “slave”) after the reception of a message sent to it by the host (as “master”), sends back an answer message after a minimum time of about 10 ms. For the communication through the serial line interface, by default, the Bluebox will apply the following parameters:

- “master/slave” protocol address 255,
- baud rate 19200 bps,
- 8 data bits,
- parity none and
- 1 stop bit.

These parameters can be modified as specified in the “General Parameters Programming” protocol command. For the communication through the ethernet interface, by default, the Bluebox will apply the following parameters:

- “master/slave” protocol address 255,
- IP address 192.168.4.200,
- port 3000,
- subnet mask 255.255.255.0 and
- gateway address 0.0.0.0.

These parameters can be modified as specified in the “General Parameters Programming” protocol command for the “master/slave” protocol address and in the “Configuration Parameters Programming” protocol command.

To simplify the explanations, the following conventions will be used:

SOH	Character 0x01
STX	Character 0x02
ETX	Character 0x03
EOT	Character 0x04
ENQ	Character 0x05
ACK	Character 0x06
NAK	Character 0x15
SYN	Character 0x16
CR	Character 0x0D
“0”..”9”	Character 0x30 .. 0x39
“A”..”F”	Character 0x41..0x46
<..>	Character 0x30..0x39, 0x41..0x46
<bcc>	Checksum

This is the general structure of a message: SOH <add h> <add l> .. <bcc> CR

SOH is the opening character, CR is the final character, <bcc> is the checksum and is calculated as “xor” of the previous characters starting from SOH and applying the following rule: if <bcc> = SOH or <bcc> = CR or <bcc> = EOT, then <bcc> := <bcc> + 1 (must be incremented by 1).

The Bluebox address is expressed with a byte (0..255 in decimal, 0x00..0xFF in hexadecimal) transformed into two ASCII characters: the first ASCII character <add h> corresponds to the ASCII coding of the high nibble of the byte, while the second ASCII character <add l> corresponds to the ASCII coding of the low nibble of the byte.

Example: 255 → 0xFF → “F” “F”.

This rule is also valid for coding a generic byte value.

For instance, the “data request” command message for a Bluebox device with address 1 will be:

SOH “0” “1” ENQ ENQ CR (in hexadecimal: 0x01, 0x30, 0x31, 0x05, 0x05, 0x0D).

2.1 Device Reset

This command is used to restart the Bluebox (the device has the same behaviour as if powered up).

The “master” sends the following command:

```
SOH <add h> <add l> STX “3” “0” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

```
SOH <add h> <add l> ACK <bcc> CR
```

2.2 Read Device Serial Number

This command is used to get the serial number of the Bluebox (unique for each device and assigned during the production process). The serial number is constituted by 6 bytes.

The “master” sends the following command:

```
SOH <add h> <add l> STX “2” “A” “0” “1” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

```
SOH <add h> <add l> STX “2” “A” “0” “1” <sn1 h> <sn1 l> .. <sn1 h> <sn1 l> .. <sn6 h>
<sn6 l> ETX <bcc> CR
```

Where:

i	1..6
<sn1 h> <sn1 l>	i-th byte of the serial number of the device. ASCII encoded byte

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

2.3 Read Ethernet MAC Address

This command is used to get the MAC address of the Ethernet interface of the Bluebox (unique for each device), the MAC address is constituted by 6 bytes.

The “master” sends the following command:

```
SOH <add h> <add l> STX “2” “A” “0” “2” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

```
SOH <add h> <add l> STX “2” “A” “0” “2” <mac1 h> <mac1 l> .. <mac1 h> <mac1 l> ..
<mac6 h> <mac6 l> ETX <bcc> CR
```

Where:

i	1..6
<mac1 h> <mac1 l>	i-th byte of the MAC address of the device. ASCII encoded byte

2.4 Read Bluetooth MAC Address

This command is used to get the MAC address of the Bluetooth interface of the Bluebox (unique for each device), the MAC address is constituted by 6 bytes.

The “master” sends the following command:

```
SOH <add h> <add l> STX “2” “A” “0” “6” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

```
SOH <add h> <add l> STX “2” “A” “0” “6” <mac1 h> <mac1 l> .. <maci h> <maci l> ..  
<mac6 h> <mac6 l> ETX <bcc> CR
```

Where:

i	1..6
<maci h> <maci l>	i-th byte of the MAC address of the device. ASCII encoded byte

2.5 Read Firmware Version

This command is used to get the firmware version of the Bluebox.

The “master” sends the following command:

```
SOH <add h> <add l> STX “3” “4” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

```
SOH <add h> <add l> STX “3” “4” <vf01 h> <vf01 l> .. <vf16 h> <vf16 l> ETX <bcc> CR
```

Where:

<vf01 h> <vf01 l>	ASCII coding of the first byte of the string
...	
<vf16 h> <vf16 l>	ASCII coding of the last byte of the string

In this case the 16 bytes are a string of 16 ASCII characters that define the version. It is also possible to get the firmware version of the reader module/s mounted in the Bluebox.

The “master” sends the following command to get the firmware version of module 1 (to get the firmware version of module 2, replace “1” with “2”):

```
SOH <add h> <add l> STX “3” “4” “0” “1” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

```
SOH <add h> <add l> STX “3” “4” “0” “1” <vf01 h> <vf01 l> .. <vf64 h> <vf64 l> ETX  
<bcc> CR
```

Where:

<vf01 h> <vf01 l>	ASCII coding of the first byte of the string
...	

<vf64 h> <vf64 l>	ASCII coding of the last byte of the string
-------------------	---

In this case, the 64 bytes are a string of 64 ASCII characters that define the version.

2.6 Firmware Upgrade

This command is used to start the firmware upgrade procedure of the Bluebox. Once the firmware upgrade is started and the Bluebox can receive the firmware image file, the file transfer takes place through the XMODEM protocol.

The “master” sends the following command:

```
SOH <add h> <add l> STX “3” “3” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

```
SOH <add h> <add l> ACK <bcc> CR
```

It is also possible to upgrade the firmware of the reader module/s mounted in the Bluebox.

The “master” sends the following command to start the firmware upgrade of module 1 (to start the firmware upgrade of module 2, replace “1” with “2”):

```
SOH <add h> <add l> STX “3” “3” “0” “1” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

```
SOH <add h> <add l> ACK <bcc> CR
```

With a successful reply, the firmware upgrade procedure is started. Once the Bluebox can receive the image file, it sends the following string “Upload Firmware?”.

Then the file transfer takes place through the XMODEM protocol. At the end of the file transfer the Bluebox checks the firmware image file it just received and copies it in the internal flash memory and reboots itself. If something goes wrong during the file transfer (communication error, power supply error, ...) the Bluebox reboots itself with the old firmware or with a boot firmware waiting for a file transfer through the XMODEM protocol.

Attention: Do not power off the Bluebox during the firmware upgrade procedure, it shall reboot itself at the end of the firmware upgrade.

2.7 Read Temperature

This command sends back the internal temperature of the Bluebox measured by the on-board temperature sensor.

The “master” sends the following command:

```
SOH <add h> <add l> STX “3” “A” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) for 52xxL, 52xxH and 52xxU devices:

SOH <add h> <add l> STX “3” “A” <temp1 h> <temp1 l> <temp2 h> <temp2 l> ETX <bcc> CR

Where:

<temp1 h> <temp1 l>	Integer value of the temperature in °C. ASCII encoded byte
<temp2 h> <temp2 l>	Fractional value of the temperature. ASCII encoded byte. Bits 7 to 5 encode the fractional value in steps of 0.125 °C: <ul style="list-style-type: none"> • 0b00000000 → 0.000 °C • 0b00100000 → 0.125 °C • ... • 0b11100000 → 0.875 °C

Note: 0x28, 0xE0 → 40.875 °C

b) for other devices:

SOH <add h> <add l> STX “3” “A” <temp h> <temp l> ETX <bcc> CR

Where:

<temp h> <temp l>	Integer value of the temperature in °C. ASCII encoded byte
----------------------	--

2.8 Read Date/Time

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

The “master” sends the following command:

SOH <add h> <add l> STX “2” “8” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> STX “2” “8” <year1 h> <year1 l> <year2 h> <year2 l> <mon h> <mon l> <day h> <day l> <hou h> <hou l> <min h> <min l> <sec h> <sec l> ETX <bcc> CR

Where:

<year1 h> <year1 l>	Year value thousands and hundreds. ASCII encoded byte. BCD encoded byte
<year2 h> <year2 l>	Year value tens and units. ASCII encoded byte. BCD encoded byte.
<mon h> <mon l>	Month value tens and units. ASCII encoded byte. BCD encoded byte.
<day h> <day l>	Day value tens and units. ASCII encoded byte. BCD encoded byte.
<hou h> <hou l>	Hour value tens and units. ASCII encoded byte. BCD encoded byte.
<min h> <min l>	Minute value tens and units. ASCII encoded byte. BCD encoded byte.
<sec h> <sec l>	Second value tens and units. ASCII encoded byte. BCD encoded byte.

2.9 Write Date/Time

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

The “master” sends the following command:

SOH <add h> <add l> STX “2” “9” <year1 h> <year1 l> <year2 h> <year2 l> <mon h> <mon l> <day h> <day l> <hou h> <hou l> <min h> <min l> <sec h> <sec l> ETX <bcc> CR

Where:

<year1 h> <year1 l>	Year value thousands and hundreds. ASCII encoded byte. BCD encoded byte
<year2 h> <year2 l>	Year value tens and units. ASCII encoded byte. BCD encoded byte.
<mon h> <mon l>	Month value tens and units. ASCII encoded byte. BCD encoded byte.
<day h> <day l>	Day value tens and units. ASCII encoded byte. BCD encoded byte.
<hou h> <hou l>	Hour value tens and units. ASCII encoded byte. BCD encoded byte.
<min h> <min l>	Minute value tens and units. ASCII encoded byte. BCD encoded byte.
<sec h> <sec l>	Second value tens and units. ASCII encoded byte. BCD encoded byte.

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

2.10 Write ROM General Parameters

This command is used to write the general parameters of the Bluebox to ROM.

The “master” sends the following command:

SOH <add h> <add l> STX “2” “F” <param1 h> <param1 l> .. <parami h> <parami l> ..
<param7 h> <param7 l> ETX <bcc> CR

Where:

i	1..7
<parami h> <parami l>	i-th byte of the general parameters (see the specific user manual for details). ASCII encoded byte.

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

Attention: After the command execution, the Bluebox performs a device reset to apply the change in RAM as well.

2.11 Write RAM Configuration Parameters

This command is used to write to RAM the configuration parameters of the Bluebox.

The “master” sends the following command:

SOH <add h> <add l> STX “3” “F” <page h> <page l> <param1 h> <param1 l> .. <parami h>
<parami l> .. <paramn h> <paramn l> ETX <bcc> CR

Where:

<page h> <page l>	The configuration page (0x00..0x0F, 0x80..0x87, 0xC0..0xCF). ASCII encoded byte.
i	1..n
n	The configuration parameters array size in bytes: <ul style="list-style-type: none"> • 7 if configuration page is 0x00..0x0F; • 14 if configuration page is 0x80..0x87; • Variable size (max. 240 bytes) null terminated string if configuration page is 0xC0..0xCF.

<parami h> <parami l>	i-th byte of the configuration parameters (see the reader user manual for details). ASCII encoded byte.
--------------------------	---

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

Attention: After the command execution, the Bluebox applies the change in RAM only for those parameters that can be changed at runtime. See the Bluebox user manual for a list of the parameters that can be changed at runtime.

2.12 Write ROM Configuration Parameters

This command is used to write to ROM the configuration parameters of the Bluebox.

The “master” sends the following command:

SOH <add h> <add l> STX “3” “D” <page h> <page l> <param1 h> <param1 l> .. <parami h> <parami l> .. <paramn h> <paramn l> ETX <bcc> CR

Where:

<page h> <page l>	The configuration page (0x00..0x0F, 0x80..0x87, 0xC0..0xCF). ASCII encoded byte.
i	1..n
n	The configuration parameters array size in bytes: <ul style="list-style-type: none"> • 7 if configuration page is 0x00..0x0F; • 14 if configuration page is 0x80..0x87; • Variable size (max. 240 bytes) null terminated string if configuration page is 0xC0..0xCF.
<parami h> <parami l>	i-th byte of the configuration parameters (see the reader user manual for details). ASCII encoded byte.

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

Attention: After the command execution, the Bluebox applies the change in RAM as well only for those parameters that can be changed at runtime. See the Bluebox user manual for a list of the parameters that can be changed at runtime.

2.13 Write ROM Default Parameters

This command is used to write in ROM the default configuration parameters of the Bluebox.

The “master” sends the following command:

SOH <add h> <add l> STX “3” “1” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

Attention: After the command execution, the Bluebox performs a device reset to apply the change in RAM as well.

2.14 Read RAM General Parameters

This command is used to read the values of the general parameters of the Bluebox from RAM.

The “master” sends the following command:

SOH <add h> <add l> STX “2” “A” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> STX “2” “A” <param1 h> <param1 l> .. <parami h> <parami l> ..
<param7 h> <param7 l> ETX <bcc> CR

Where:

i	1..7
<parami h> <parami l>	i-th byte of the general parameters (see the specific user manual for details). ASCII encoded byte.

2.15 Read RAM Configuration Parameters

This command is used to read the values of the configuration parameters of the Bluebox from RAM.

The “master” sends the following command:

SOH <add h> <add l> STX “3” “C” <page h> <page l> ETX <bcc> CR

Where:

<page h> <page l>	The configuration page (0x00..0x0F, 0x80..0x87, 0xC0..0xCF). ASCII encoded byte.
-------------------	--

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> STX “3” “C” <param1 h> <param1 l> .. <parami h> <parami l> ..
<paramn h> <paramn l> ETX <bcc> CR

Where:

i	1..n
n	The configuration parameters array size in bytes: <ul style="list-style-type: none"> 7 if configuration page is 0x00..0x0F; 14 if configuration page is 0x80..0x87; Variable size (max. 240 bytes) null terminated string if configuration page is 0xC0..0xCF.
<parami h> <parami l>	i-th byte of the configuration parameters (see the reader user manual for details). ASCII encoded byte.

2.16 Read ROM Configuration Parameters

This command is used to read the values of the configuration parameters of the Bluebox from ROM.

The “master” sends the following command:

SOH <add h> <add l> STX “3” “E” <page h> <page l> ETX <bcc> CR

Where:

<page h> <page l>	The configuration page (0x00..0x0F, 0x80..0x87, 0xC0..0xCF). ASCII encoded byte.
-------------------	--

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> STX “3” “E” <param1 h> <param1 l> .. <parami h> <parami l> .. <paramn h> <paramn l> ETX <bcc> CR

Where:

i	1..n
n	The configuration parameters array size in bytes: <ul style="list-style-type: none"> • 7 if configuration page is 0x00..0x0F; • 14 if configuration page is 0x80..0x87; Variable size (max. 240 bytes) null terminated string if configuration page is 0xC0..0xCF.
<parami h> <parami l>	i-th byte of the configuration parameters (see the reader user manual for details). ASCII encoded byte.

2.17 Sleep Mode

This command is used to manage the sleep mode of the Bluebox.

The “master” sends the following command:

SOH <add h> <add l> STX “3” “B” <mode h> <mode l> ETX <bcc> CR

Where:

<mode h> <mode l>	The sleep mode. ASCII encoded byte: <ul style="list-style-type: none"> • 0x00: To exit sleep mode • 0x01: To enter sleep mode. Any command on the serial interface wakes up the device; after the reply, the device stays active • 0x02: To enter sleep mode. Any command on the serial interface wakes up the device; after the reply, the device enters sleep mode again
----------------------	---

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

2.18 “RF Reading” Test

In “continuous” mode, this command is used to activate/deactivate the “reading” test mode. It allows the user to easily and quickly test the read range of the reader with fast beeping (100 ms) the buzzer for every identified tag.

The “master” sends the following command:

SOH <add h> <add l> STX “D” “7” <onoff h> <onoff l> ETX <bcc> CR

Where:

<onoff h> <onoff l>	To activate/deactivate the “RF reading” test mode. ASCII encoded byte: <ul style="list-style-type: none"> • 0x00: To deactivate “RF reading” test mode • 0x01: To activate “RF reading” test mode.
------------------------	--

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

Attention: The “RF reading” test mode setting is stored in non-volatile memory and its status is kept at every restart of the Bluebox.

2.19 “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 the maximum RF output power or until it finds a tag, increasing the RF power by 1 dB every 500 ms with fixed Q selection algorithm and Q=0.

The “master” sends the following command:

SOH <add h> <add l> STX “D” “A” “0” <antenna> <channel h> <channel l> ETX <bcc> CR

Where:

<antenna>	Antenna to use for the test. ASCII character: <ul style="list-style-type: none"> • “1”: Antenna 1 • “2”: Antenna 2 • “3”: Antenna 3 • “4”: Antenna 4
<channel h> <channel l>	RF channel to use for the test. ASCII encoded byte: <ul style="list-style-type: none"> • 0x01..0x0A if ETSI region is selected • 0x01..0x32 if FCC region is selected

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a tag has been identified

SOH <add h> <add l> STX “D” “A” “0” “0” <power h> <power l> ETX <bcc> CR

Where:

<power h> <power l>	Minimum RF output power needed to read the tag. ASCII encoded byte.
------------------------	---

- b) if no tag has been found

SOH <add h> <add l> STX “D” “A” “0” “1” ETC <bcc> CR

2.20 “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 the maximum RF input sensitivity or until it finds a tag, increasing the RF sensitivity by 1 dB every 500 ms with fixed Q selection algorithm and Q=0.

The “master” sends the following command:

SOH <add h> <add l> STX “D” “B” “0” <antenna> <channel h> <channel l> ETX <bcc> CR

Where:

<antenna>	Antenna to use for the test. ASCII character: <ul style="list-style-type: none"> • “1”: Antenna 1 • “2”: Antenna 2 • “3”: Antenna 3
-----------	--

	<ul style="list-style-type: none"> • “4”: Antenna 4
<channel h> <channel l>	RF channel to use for the test. ASCII encoded byte: <ul style="list-style-type: none"> • 0x01..0x0A if ETSI region is selected • 0x01..0x32 if FCC region is selected

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

a) if a tag has been identified

SOH <add h> <add l> STX “D” “B” “0” “0” <sens h> <sens l> ETX <bcc> CR

Where:

<sens h> <sens l>	Absolut value of the minimum RF sensitivity needed to read the tag. ASCII encoded byte.
-------------------	---

b) if no tag has been found

SOH <add h> <add l> STX “D” “B” “0” “1” ETC <bcc> CR

2.21 Read Reflected Power

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

The “master” sends the following command:

SOH <add h> <add l> STX “F” “E” “0” <antenna> <freq1 h> <freq1 l> <freq2 h> <freq2 l>
<freq3 h> <freq3 l> ETX <bcc> CR

Where:

<antenna>	Antenna to use for the test. ASCII character: <ul style="list-style-type: none"> • “1”: Antenna 1 • “2”: Antenna 2 • “3”: Antenna 3 • “4”: Antenna 4
<freq1 h> <freq1 l> <freq2 h> <freq2 l> <freq3 h> <freq3 l>	The frequency to test in MHz in the range 840 – 960 MHz. 3-bytes ASCII encoded value.

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> STX “F” “E” <Ich h> <Ich l> <Qch h> <Qch l> <G h> <G l> <bcc> CR

Where:

<Ich h> <Ich l>	The I-channel RSSI value. ASCII encoded byte.
<Qch h> <Qch l>	The Q-channel RSSI value. ASCII encoded byte.
<G h> <G l>	The G value used to calculate the reflected power as defined below. ASCII encoded value.

The reflected power is calculated as follows:

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

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

2.22 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.

The “master” sends the following command:

```
SOH <add h> <add l> STX “F” “D” “0” <antenna> <freq1 h> <freq1 l> <freq2 h> <freq2 l>
<freq3 h> <freq3 l> ETX <bcc> CR
```

Where:

<antenna>	Antenna to use for the test. ASCII character: <ul style="list-style-type: none"> • “1”: Antenna 1 • “2”: Antenna 2 • “3”: Antenna 3 • “4”: Antenna 4
<freq1 h> <freq1 l> <freq2 h> <freq2 l> <freq3 h> <freq3 l>	The frequency to test in MHz in the range 840 – 960 MHz. 3-bytes ASCII encoded value.

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

```
SOH <add h> <add l> STX “F” “D” <Ich h> <Ich l> <Qch h> <Qch l> <G h> <G l> <bcc> CR
```

Where:

<Ich h> <Ich l>	The I-channel RSSI value. ASCII encoded byte.
<Qch h> <Qch l>	The Q-channel RSSI value. ASCII encoded byte.
<G h> <G l>	The G value used to calculate the reflected power as defined below. ASCII encoded value.

The RSSI power is calculated as follows:

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

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

2.23 Digital Output Activation

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

The “master” sends the following command:

```
SOH <add h> <add l> STX “3” “7” <chn h> <chn l> <dur h> <dur l> ETX <bcc> CR
```

Where:

<chn h> <chn l>	Output to activate. ASCII encoded byte <ul style="list-style-type: none"> • 0x01: Output 1 • 0x02: Output 2
<dur h> <dur l>	Activation time. ASCII encoded byte <ul style="list-style-type: none"> • 0x01..0x63 (1..99 seconds) → “Impulsive” output activation • 0x81 → “Continuous” activation • 0x80 → Deactivation

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

2.24 Status Reading

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

The “master” sends the following command:

SOH <add h> <add l> STX “3” “6” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> STX “3” “6” <status hh> <status hl> <status lh> <status ll> ETX <bcc> CR

Where:

<status hh> <status hl> <status lh> <status ll>	Bluebox status, ASCII encoded word. See the device user manual for details.
--	---

2.25 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.

The “master” sends the following command:

SOH <add h> <add l> STX “3” “8” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

2.26 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.

The “master” sends the following command:

SOH <add h> <add l> STX “3” “9” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

2.27 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.

The “master” sends the following command:

```
SOH <add h> <add l> STX “D” “4” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

```
SOH <add h> <add l> ACK <bcc> CR
```

2.28 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 held 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.

The “master” sends the following command:

```
SOH <add h> <add l> ENQ <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) for LF devices with only 1 antenna and an identified transponder

```
SOH <add h> <add l> STX <code1 h> <code1 l> .. <codei h> <codei l> .. <coden h> <coden l> ETX <bcc> CR
```

Where:

i	1..n
n	Number of bytes of the tag code identified by the antenna: <ul style="list-style-type: none"> 5: UNIQUE, Bluebox Short 10: Bluebox Medium 20: Bluebox Large
<codei h> <codei l>	i-th byte of the code of the tag identified by the antenna. ASCII encoded byte

- b) for LF devices with only 1 antenna and no identified transponder

```
SOH <add h> <add l> STX “0” “0” “0” “0” “0” “0” “0” “0” “0” “0” ETX <bcc> CR
```

- c) for LF devices with 2 antennas and both antennas have identified a transponder

```
SOH <add h> <add l> STX <codeant11 h> <codeant11 l> .. <codeant1i h> <codeant1i l> .. <codeant1n h> <codeant1n l> “-” <codeant21 h> <codeant21 l> .. <codeant2j h> <codeant2j l> .. <codeant2m h> <codeant2m l> ETX <bcc> CR
```

Where:

i	1..n
n	Number of bytes of the tag code identified by antenna 2: <ul style="list-style-type: none"> 5: UNIQUE, Bluebox Short

	<ul style="list-style-type: none"> 10: Bluebox Medium 20: Bluebox Large
<codeant1i h> <codeant1i l>	i-th byte of the code of the tag identified by antenna 1. ASCII encoded byte
j	1..m
m	Number of bytes of the tag code identified by antenna 2: <ul style="list-style-type: none"> 5: UNIQUE, Bluebox Short 10: Bluebox Medium 20: Bluebox Large
<codeant2i h> <codeant2i l>	j-th byte of the code of the tag identified by antenna 2. ASCII encoded byte

- d) for LF devices with 2 antennas and only antenna 1 has identified a transponder

SOH <add h> <add l> STX <codeant11 h> <codeant11 l> .. <codeant1i h> <codeant1i l>
 .. <codeant1n h> <codeant1n l> “-” “0” “0” “0” “0” “0” “0” “0” “0” “0” “0” ETX <bcc>
 CR

Where:

i	1..n
n	Number of bytes of the tag code identified by antenna 2: <ul style="list-style-type: none"> 5: UNIQUE, Bluebox Short 10: Bluebox Medium 20: Bluebox Large
<codeant1i h> <codeant1i l>	i-th byte of the code of the tag identified by antenna 1. ASCII encoded byte

- e) for LF devices with 2 antennas and only antenna 2 has identified a transponder

SOH <add h> <add l> STX “0” “0” “0” “0” “0” “0” “0” “0” “0” “0” “-” <codeant21 h>
 <codeant21 l> .. <codeant2j h> <codeant2j l> .. <codeant2m h> <codeant2m l> ETX <bcc>
 CR

Where:

j	1..m
m	Number of bytes of the tag code identified by antenna 2: <ul style="list-style-type: none"> 5: UNIQUE, Bluebox Short 10: Bluebox Medium 20: Bluebox Large
<codeant2i h> <codeant2i l>	j-th byte of the code of the tag identified by antenna 2. ASCII encoded byte

- f) for LF devices with 2 antennas and none of the antennas have identified a transponder

SOH <add h> <add l> STX “0” “0” “0” “0” “0” “0” “0” “0” “0” “0” “-” “0” “0” “0” “0”
 “0” “0” “0” “0” “0” “0” ETX <bcc> CR

- g) for HF devices with only 1 antenna and an identified transponder

SOH <add h> <add l> STX <type h> <type l> <UID1 h> <UID1 l> .. <UIDi h> <UIDi l> ..
 <UIDn h> <UIDn l> ETX <bcc> CR

Where:

<type h> <type l>	Transponder type
i	1..n (the UID length)
<UIDi h> <UIDi l>	i-th byte of the UID code of the identified tag. ASCII encoded byte

- h) for HF devices with only 1 antenna and no identified transponder
 SOH <add h> <add l> STX "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" ETX <bcc> CR
- i) for HF devices with 2 antennas and both antennas identified a transponder
 SOH <add h> <add l> STX <typeant1 h> <typeant1 l> <UIDant11 h> <UIDant11 l> ..
 <UIDant1i h> <UIDant1i l> .. <UIDant1n h> <UIDant1n l> "-" <typeant2 h> <typeant2 l>
 <UIDant21 h> <UIDant21 l> .. <UIDant2j h> <UIDant2j l> .. <UIDant2m h> <UIDant2m l>
 ETX <bcc> CR

Where:

<typeant11 h> <typeant11 l>	Transponder type identified by antenna 1
i	1..n (the UID length)
<UIDant1i h> <UIDant1i l>	i-th byte of the UID code of the tag identified by antenna 1. ASCII encoded byte
<typeant21 h> <typeant21 l>	Transponder type identified by antenna 2
j	1..m (the UID length)
<UIDant2j h> <UIDant2j l>	j-th byte of the UID code of the tag identified by antenna 2. ASCII encoded byte

- j) for HF devices with 2 antennas and only antenna 1 identified a transponder
 SOH <add h> <add l> STX <typeant1 h> <typeant1 l> <UIDant11 h> <UIDant11 l> ..
 <UIDant1i h> <UIDant1i l> .. <UIDant1n h> <UIDant1n l> "-" "0" "0" "0" "0" "0" "0"
 "0" "0" "0" "0" ETX <bcc> CR

Where:

<typeant11 h> <typeant11 l>	Transponder type identified by antenna 1
i	1..n (the UID length)
<UIDant1i h> <UIDant1i l>	i-th byte of the UID code of the tag identified by antenna 1. ASCII encoded byte

- k) for HF devices with 2 antennas and only antenna 2 identified a transponder
 SOH <add h> <add l> STX "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "-" <typeant2 h>
 <typeant2 l> <UIDant21 h> <UIDant21 l> .. <UIDant2j h> <UIDant2j l> .. <UIDant2m h>
 <UIDant2m l> ETX <bcc> CR

Where:

<typeant21 h> <typeant21 l>	Transponder type identified by antenna 2
j	1..m (the UID length)
<UIDant2j h> <UIDant2j l>	j-th byte of the UID code of the tag identified by antenna 2. ASCII encoded byte

- l) for HF devices with 2 antennas and none of the antennas have identified a transponder
 SOH <add h> <add l> STX "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "-" "0" "0" "0" "0"
 "0" "0" "0" "0" "0" "0" ETX <bcc> CR

- m) for UHF devices

SOH <add h> <add l> STX
 <tag1 h> <tag1 l> <ID11 h> <ID11 l> .. <ID1i h> <ID1i l> .. <ID1m h> <ID1m l>
 <1st RSSIQ1 h> <1st RSSIQ1 l> <1st RSSII1 h> <1st RSSII1 l>

```

<last RSSIQ1 h> <last RSSIQ1 l> <last RSSII1 h> <last RSSII1 l>
<max RSSIQ1 h> <max RSSIQ1 l> <max RSSII1 h> <max RSSII1 l>
<ant1 h> <ant1 l> <dir1 h> <dir1 l>
<1st tm11 h> <1st tm11 l> .. <1st tm17 h> <1st tm17 l>
<last tm11 h> <last tm11 l> .. <last tm17 h> <last tm17 l>
<rdcount1 hh> <rdcount1 hl> <rdcount1 lh> <rdcount1 ll>
“_” .. “_”
<tagj h> <tagj l> <IDj1 h> <IDj1 l> .. <IDji h> <IDji l> .. <IDjm h> <IDjm l>
<1st RSSIQj h> <1st RSSIQj l> <1st RSSIIj h> <1st RSSIIj l>
<last RSSIQj h> <last RSSIQj l> <last RSSIIj h> <last RSSIIj l>
<max RSSIQj h> <max RSSIQj l> <max RSSIIj h> <max RSSIIj l>
<antj h> <antj l> <dirj h> <dirj l>
<1st tmj1 h> <1st tmj1 l> .. <1st tmj7 h> <1st tmj7 l>
<last tmj1 h> <last tmj1 l> .. <last tmj7 h> <last tmj7 l>
<rdcountj hh> <rdcountj hl> <rdcountj lh> <rdcountj ll>
“_” .. “_”
<tagn h> <tagn l> <IDn1 h> <IDn1 l> .. <IDni h> <IDni l> .. <IDnm h> <IDnm l>
<1st RSSIQn h> <1st RSSIQn l> <1st RSSIIIn h> <1st RSSIIIn l>
<last RSSIQn h> <last RSSIQn l> <last RSSIIIn h> <last RSSIIIn l>
<max RSSIQn h> <max RSSIQn l> <max RSSIIIn h> <max RSSIIIn l>
<antn h> <antn l> <dirn h> <dirn l>
<1st tmn1 h> <1st tmn1 l> .. <1st tmn7 h> <1st tmn7 l>
<last tmn1 h> <last tmn1 l> .. <last tmn7 h> <last tmn7 l>
<rdcountn hh> <rdcountn hl> <rdcountn lh> <rdcountn ll>
ETX <bcc> CR

```

Where:

i	1..m
m	ID length
j	1..n
n	Number of identified tags
<tagj h> <tagj l>	Transponder type for the j-th identified tag (optional parameter present only if the tag type information flag in the general parameters is active, see the user manual for mode info) ASCII encoded byte: 0x02: ISO 18000-63 (EPC Class-1 Generation-2)
<IDji h> <IDji l>	i-th byte of the ID of the j-th identified tag. ASCII encoded byte
<1st RSSIQj h> <1st RSSIQj l> <1st RSSIIj h> <1st RSSIIj l>	First seen RSSI Q and I values in dB for the j-th identified tag (optional parameter present only if the RSSI information flag in the RF configuration parameters is active, see the reader user manual for more info). ASCII encoded bytes.
<last RSSIQj h> <last RSSIQj l> <last RSSIIj h> <last RSSIIj l>	Last seen RSSI Q and I values in dB for the j-th identified tag (optional parameter present only if the RSSI information flag in the RF configuration parameters is active, see the reader user manual for more info). ASCII encoded bytes.
<max RSSIQj h> <max RSSIQj l> <max RSSIIj h> <max RSSIIj l>	Max seen RSSI Q and I values in dB for the j-th identified tag (optional parameter present only if the max RSSI information flag in the RF configuration parameters is active, see the reader user manual for more info). ASCII encoded bytes.
<antj h> <antj l>	Reading antenna for the j-th identified tag (optional parameter present only if the reading antenna information flag in the general parameters is active, see the reader user manual for more info). ASCII character: <ul style="list-style-type: none"> “1”: Antenna 1

	<ul style="list-style-type: none"> • “2”: Antenna 2 • “3”: Antenna 3 • “4”: Antenna 4
<dirj h> <dirj l>	Gate crossing direction for the j-th identified tag (optional parameter present only if “gate” mode is active, see the reader user manual for more info). ASCII character: <ul style="list-style-type: none"> • 0x01 → Crossing from input 1 to input 2 • 0x02 → Crossing from input 2 to input 1
<1st tmj1 h> <1st tmj1 l> ... <1st tmj7 h> <1st tmj7 l>	First seen timestamp for the j-th identified tag (optional parameter present only if the reading timestamp information flag in the general parameters is active, see the reader user manual for more info). ASCII encoded byte array of the BCD encoded timestamp with the format yyyyMMddhhmmss.
<last tmj1 h> <last tmj1 l> ... <last tmj7 h> <last tmj7 l>	Last seen timestamp for the j-th identified tag (optional parameter present only if the reading timestamp information flag in the general parameters is active, see the reader user manual for more info). ASCII encoded byte array of the BCD encoded timestamp with the format yyyyMMddhhmmss.
<rdcountj hh> <rdcountj hl> <rdcountj lh> <rdcountj ll>	The tag read count for the j-th identified tag (optional parameter present only if the tag read count information flag in the RF parameters is active, see the reader user manual for more info). ASCII encoded word.
“_”	Separator 0x5F

2.29 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.

The “master” sends the following command:

SOH <add h> <add l> SYN <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

a) for LF devices with only 1 antenna

SOH <add h> <add l> STX <code1 h> <code1 l> .. <codei h> <codei l> .. <coden h>
<coden l> ETX <bcc> CR

Where:

i	1..n
n	Number of bytes of the tag code identified by antenna 2: <ul style="list-style-type: none"> • 5: UNIQUE, Bluebox Short • 10: Bluebox Medium • 20: Bluebox Large
<codei h> <codei l>	i-th byte of the code of the tag identified by antenna 1. ASCII encoded byte

b) for LF devices with 2 antennas

SOH <add h> <add l> STX <code1 h> <code1 l> .. <codei h> <codei l> .. <coden h>
<coden l> <ant h> <ant l> < ETX <bcc> CR

Where:

i	1..n
---	------

n	Number of bytes of the tag code identified by antenna 2: <ul style="list-style-type: none"> • 5: UNIQUE, Bluebox Short • 10: Bluebox Medium • 20: Bluebox Large
<codei h> <codei l>	i-th byte of the code of the tag identified by antenna 1. ASCII encoded byte
<ant h> <ant l>	The antenna number which identified the tag; ASCII encoded byte: <ul style="list-style-type: none"> • 0x01: Antenna 1 • 0x02: Antenna 2

c) for HF devices with only 1 antenna

SOH <add h> <add l> STX <type h> <type l> <UID1 h> <UID1 l> .. <UIDi h> <UIDi l> .. <UIDn h> <UIDn l> ETX <bcc> CR

Where:

<type h> <type l>	Transponder type
i	1..n (the UID length)
<UIDi h> <UIDi l>	i-th byte of the UID code of the identified tag. ASCII encoded byte

d) for HF devices with 2 antennas

SOH <add h> <add l> STX <type h> <type l> <UID1 h> <UID1 l> .. <UIDi h> <UIDi l> .. <UIDn h> <UIDn l> <ant h> <ant l> ETX <bcc> CR

Where:

<type h> <type l>	Transponder type
i	1..n (the UID length)
<UIDi h> <UIDi l>	i-th byte of the UID code of the identified tag. ASCII encoded byte
<ant h> <ant l>	The antenna number which identified the tag; ASCII encoded byte: <ul style="list-style-type: none"> • 0x01: Antenna 1 • 0x02: Antenna 2

e) for UHF devices

SOH <add h> <add l> STX <tag h> <tag l> <ID1 h> <ID1 l> .. <IDi h> <IDi l> .. <IDm h> <IDm l> <RSSIQ h> <RSSIQ l> <RSSII h> <RSSII l> <ant h> <ant l> <dir h> <dir l> <tm1 h> <tm1 l> .. <tm7 h> <tm7 l> ETX <bcc> CR

Where:

i	1..m
m	ID length
<tag h> <tag l>	Transponder type for the identified tag (optional parameter present only if the tag type information flag in the general parameters is active, see the user manual for mode info) ASCII encoded byte: 0x02: ISO 18000-63 (EPC Class-1 Generation-2)
<IDi h> <IDi l>	i-th byte of the ID of the identified tag. ASCII encoded byte
<RSSIQ h> <RSSIQ l> <RSSII h> <RSSII l>	RSSI Q and I channel info in dB of the identified tag. (optional parameter present only if the RSSI information flag in the RF configuration parameters is active, see the reader user manual for more info). ASCII encoded bytes.
<ant h> <ant l>	Reading antenna for the identified tag (optional parameter present only if the reading antenna information flag in the general parameters is active, see the reader user manual for more info). ASCII character: <ul style="list-style-type: none"> • "1": Antenna 1 • "2": Antenna 2 • "3": Antenna 3 • "4": Antenna 4

<dir h> <dir l>	Gate crossing direction for the identified tag (optional parameter present only if “gate” mode is active, see the reader user manual for more info). ASCII character: <ul style="list-style-type: none"> 0x01 → Crossing from input 1 to input 2 0x02 → Crossing from input 2 to input 1
<tm1 h> <tm1 l> ... <tm7 h> <tm7 l>	Timestamp for the identified tag (optional parameter present only if the reading timestamp information flag in the general parameters is active, see the reader user manual for more info). ASCII encoded byte array of the BCD encoded timestamp with the format yyyyMMddhhmmss.

If the queue is empty, the Bluebox will answer with:

SOH <add h> <add l> STX “0” “0” “0” “0” “0” “0” “0” “0” “0” “0” ETX <bcc> CR

To delete the received code from the queue, the “master” replies to the Bluebox with

SOH <add h> <add l> ACK <bcc> CR

2.30 Read Number of Records

This command is used to get the number of unread records in the database of the Bluebox.

The “master” sends the following command:

SOH <add h> <add l> STX “0” “0” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> STX “0” “0” <num hh> <num hl> <num lh> <num ll> ETX <bcc> CR

Where:

<num hh> <num hl> <num lh> <num ll>	Number of unread records in the database. ASCII encoded word
--	--

2.31 Reset Record Database

This command is used to reset all the records stored in the Bluebox.

The “master” sends the following command:

SOH <add h> <add l> STX “3” “5” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

2.32 Read Current Record

This command is used to get the first unread record from the database of the Bluebox.

The “master” sends the following command:

SOH <add h> <add l> STX “0” “1” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> STX "0" "1" <idt hh> <idt hl> <idt lh> <idt ll> <record1 h>
<record1 l> .. <recordn h> <recordn l> ETX <bcc> CR

Where:

<idt hh> <idt hl> <idt lh> <idt ll>	Index of the record in the database. ASCII encoded word
n	The record size in bytes. Must be a power of 2 minus 1 bytes of header and checksum (for example 15 or 31 bytes)
<record1 h> <record1 l>	1st byte of the record. ASCII encoded byte
...	
<recordn h> <recordn l>	Last byte of the record. ASCII encoded byte

Attention: The field <idt> in the answer must be used to dequeue the record from the database.

2.33 Dequeue Current Record

This command is used to dequeue the first unread record in the database of the Bluebox.

The "master" sends the following command:

SOH <add h> <add l> STX "0" "2" <idt hh> <idt hl> <idt lh> <idt ll> ETX <bcc> CR

Where:

<idt hh> <idt hl> <idt lh> <idt ll>	Index of the record to dequeue. ASCII encoded word
--	--

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

Attention: The field <idt> in the command is the index of the record to dequeue received with the command "Read Current Record".

2.34 Re-Read an Dequeued Record

This command is used to get an already read and dequeued record from the database of the Bluebox.

The "master" sends the following command:

SOH <add h> <add l> STX "0" "3" <num hh> <num hl> <num lh> <num ll> ETX <bcc> CR

Where:

<num hh> <num hl> <num lh> <num ll>	The index of the already read record starting from the first unread record in the database. ASCII encoded word. 0001 to read the last read record, ..., 0000 not allowed.
--	---

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> STX "0" "3" <num hh> <num hl> <num lh> <num ll> <record1 h>
<record1 l> .. <recordn h> <recordn l> ETX <bcc> CR

Where:

<num hh> <num hl> <num lh> <num ll>	The index of the already read record starting from the first unread record in the database. ASCII encoded word. 0001 to read the last read record, ..., 0000 not allowed.
n	The record size in bytes. Must be a power of 2 minus 1 bytes of header and checksum (for example 15 or 31 bytes)
<record1 h> <record1 l>	1st byte of the record. ASCII encoded byte
...	
<recordn h> <recordn l>	Last byte of the record. ASCII encoded byte

2.35 Start Continuous Read Records

This command is used to start the reception of the unread records from the database of the Bluebox.

The "master" sends the following command:

SOH <add h> <add l> STX "5" "0" ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> STX "5" "0" <idt hh> <idt hl> <idt lh> <idt ll> <num hh> <num hl>
<num lh> <num ll> ETX <bcc> CR

SOH <add h> <add l> STX "5" "0" <id hh> <id hl> <id lh> <id ll> <record1 h>
<record1 l> .. <recordn h> <recordn l> ETX <bcc> CR

<..>

SOH <add h> <add l> STX "5" "0" ETX <bcc> CR

Where:

<idt hh> <idt hl> <idt lh> <idt ll>	Index of the first unread record in the database. ASCII encoded word
<num hh> <num hl> <num lh> <num ll>	Number of unread records in the database. ASCII encoded word
n	The record size in bytes. Must be a power of 2 minus 1 bytes of header and checksum (for example 15 or 31 bytes)
<record1 h> <record1 l>	1st byte of the record. ASCII encoded byte
...	
<recordn h> <recordn l>	Last byte of the record. ASCII encoded byte
<..>	The records to read

In case of no unread records, it answers with

SOH <add h> <add l> STX "5" "0" <idt hh> <idt hl> <idt lh> <idt ll> "0" "0" "0" "0"
 ETX <bcc> CR

Attention: The field <idt> in the answer must be used to dequeue the records from the database.

2.36 Stop Continuous Read Records

This command is used to stop the continuous reception of the records and to dequeue the records from the database of the Bluebox.

The "master" sends the following command:

SOH <add h> <add l> STX "5" "2" <idt hh> <idt hl> <idt lh> <idt ll> <idu hh> <idu hl>
 <idu lh> <idu ll> ETX <bcc> CR

Where:

<idt hh> <idt hl> <idt lh> <idt ll>	Index of the first unread record in the database. ASCII encoded word
<idu hh> <idu hl> <idu lh> <idu ll>	Index of the last unread record in the database. ASCII encoded word

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

2.37 Write Data to an EM4305 Transponder

This command is used to write data to an 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

The "master" sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace "1" "9" with "6" "9"):

SOH <add h> <add l> STX "1" "9" <code1 h> .. <code1 l> .. <codei h> <codei l> ..
 <coden h> <coden l> ETX <bcc> CR

Where:

i	1..n
n	Number of bytes of the tag code identified by antenna 1: <ul style="list-style-type: none"> • 5: UNIQUE, Bluebox Short • 10: Bluebox Medium • 20: Bluebox Large
<codei h> <codei l>	i-th byte of the code of the tag identified by antenna 1. ASCII encoded byte

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

2.38 Read ID Code of an EM4305 Transponder

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

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “1” “8” with “6” “8”):

```
SOH <add h> <add l> STX “1” “8” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

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

```
SOH <add h> <add l> STX “1” “8” “0” “0” <code1 h> <code1 l>..  
<codei h> <codei l>..  
<code4 h> <code4 l> ETX <bcc> CR
```

Where:

i	1..4
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.

- b) if errors occurred:

```
SOH <add h> <add l> STX “1” “8” “0” “2” ETX <bcc> CR
```

- c) if no transponder is present:

```
SOH <add h> <add l> STX “1” “8” “0” “1” ETX <bcc> CR
```

2.39 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 5 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

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “1” “D” with “6” “D”):

```
SOH <add h> <add l> STX “1” “D” <code1 h> <code1 l>..  
<codei h> <codei l>..  
<coden h> <coden l> ETX <bcc> CR
```

Where:

i	1..n
n	Number of bytes of the tag code identified by antenna 1: <ul style="list-style-type: none"> • 5: UNIQUE, Bluebox Short • 10: Bluebox Medium • 20: Bluebox Large
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

```
SOH <add h> <add l> ACK <bcc> CR
```

2.40 Read ID Code of a T5557 Transponder

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

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “1” “C” with “6” “C”):

SOH <add h> <add l> STX “1” “C” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “1” “C” “0” “0” <code1 h> <code1 l>..<code8 h> <code8 l> ETX <bcc> CR

Where:

i	1..8
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.

- b) if errors occurred:

SOH <add h> <add l> STX “1” “C” “0” “2” ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX “1” “C” “0” “1” ETX <bcc> CR

2.41 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

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “2” “1” with “7” “1”):

SOH <add h> <add l> STX “2” “1” <code1 h> <code1 l>..<code n l> ETX <bcc> CR

Where:

i	1..n
n	Number of bytes of the tag code identified by antenna 1: <ul style="list-style-type: none"> • 5: UNIQUE, Bluebox Short • 10: Bluebox Medium • 20: Bluebox Large
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

SOH <add h> <add l> ACK <bcc> CR

2.42 Read ID Code of a Q5 Transponder

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

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “2” “0” with “7” “0”):

SOH <add h> <add l> STX “2” “0” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “2” “0” “0” “0” <code1 h> <code1 l>..<code5 h> <code5 l> ETX <bcc> CR

Where:

i	1..5
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.

- b) if errors occurred:

SOH <add h> <add l> STX “2” “0” “0” “2” ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX “2” “0” “0” “1” ETX <bcc> CR

2.43 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.

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “2” “3” with “7” “3”):

SOH <add h> <add l> STX “2” “3” <code1 h> <code1 l>..<code n l> ETX <bcc> CR

Where:

i	1..n
n	Number of bytes of the tag code identified by antenna 1: <ul style="list-style-type: none"> 5: UNIQUE, Bluebox Short 10: Bluebox Medium
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> ACK <bcc> CR

- b) if errors occurred:

SOH <add h> <add l> STX “2” “3” “0” “2” ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "2" "3" "0" "1" ETX <bcc> CR

2.44 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.

The "master" sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace "2" "2" with "7" "2"):

SOH <add h> <add l> STX "2" "2" ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "2" "2" "0" "0" <type h> <type l> <code1 h> <code1 l>..

Where:

i	1..4
<type h> <type l>	The tag type. ASCII encoded byte: <ul style="list-style-type: none"> • 0x01: HITAG S 256 • 0x02: HITAG S 2048 • 0x03: HITAG 1
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.

- b) if errors occurred:

SOH <add h> <add l> STX "2" "2" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "2" "2" "0" "1" ETX <bcc> CR

2.45 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.

The "master" sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace "2" "4" with "7" "4"):

SOH <add h> <add l> STX "2" "4" <code1 h> <code1 l>..

Where:

i	1..4
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.
<pag h> <pag l>	Page to be read. ASCII encoded byte: <ul style="list-style-type: none"> • 0x00..0x3F for HITAG 1 transponders • 0x00..0x07 for HITAG S 256 transponders • 0x00..0x3F for HITAG S 2048 transponders

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "2" "4" "0" "0" <data1 h> <data1 l>..
<datai h> <datai l>..
<data4 h> <data4 l> ETX <bcc> CR

Where:

i	1..4
<datai h> <datai l>	i-th byte of the tag page. ASCII encoded byte.

- b) if errors occurred:

SOH <add h> <add l> STX "2" "4" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "2" "4" "0" "1" ETX <bcc> CR

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

This command is used to write a data page of the HITAG1 / HITAG S transponder, constituted by 32 bits (4 bytes). Note that it is necessary to know the ID code of the transponder.

The "master" sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace "2" "5" with "7" "5"):

SOH <add h> <add l> STX "2" "5" <code1 h> <code1 l>..
<codei h> <codei l>..
<code4 h> <code4 l> <pag h> <pag l> <data1 h> <data1 l>..
<dataj h> <dataj l>..
<data4 h> <data4 l> ETX <bcc> CR

Where:

i	1..4
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.
<pag h> <pag l>	Page to be read. ASCII encoded byte: <ul style="list-style-type: none"> • 0x00..0x3F for HITAG 1 transponders • 0x00..0x07 for HITAG S 256 transponders • 0x00..0x3F for HITAG S 2048 transponders
j	1..4
<dataj h> <dataj l>	j-th byte of the tag page. ASCII encoded byte

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "2" "5" "0" "0" ETX <bcc> CR

- b) if errors occurred:

SOH <add h> <add l> STX "2" "5" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "2" "5" "0" "1" ETX <bcc> CR

2.47 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.

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “4” “8” with “9” “8”):

SOH <add h> <add l> STX “4” “8” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the ID has been successfully read:
- SOH <add h> <add l> STX “4” “8” “0” “0” <code1 h> <code1 l>..`i h> <codei l>..`

Where:

i	1..4
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.

- b) if errors occurred:
- SOH <add h> <add l> STX “4” “8” “0” “2” ETX <bcc> CR
- c) if no transponder is present:
- SOH <add h> <add l> STX “4” “8” “0” “1” ETX <bcc> CR

2.48 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.

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “4” “A” with “9” “A”):

SOH <add h> <add l> STX “4” “A” <code1 h> <code1 l>..`i h> <codei l>..`

Where:

i	1..4
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.
j	1..4
<pjdj h> <pjdj l>	j-th byte of the password of the tag. ASCII encoded byte
<pag h> <pag l>	Page to be read. ASCII encoded byte (0x00..0x07)

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the page has been successfully read:
- SOH <add h> <add l> STX “4” “A” “0” “0” <data1 h> <data1 l>..

Where:

i	1..4
---	------

<datai h> <datai l>	i-th byte of the tag page. ASCII encoded byte.
---------------------	--

b) if errors occurred:
SOH <add h> <add l> STX "4" "A" "0" "2" ETX <bcc> CR

c) if no transponder is present:
SOH <add h> <add l> STX "4" "A" "0" "1" ETX <bcc> CR

2.49 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.

The "master" sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace "4" "B" with "9" "B"):

```
SOH <add h> <add l> STX "4" "B" <code1 h> <code1 l>..  
<codei h> <codei l>..  
<code4 h> <code4 l> <pwd1 h> <pwd1 l>..  
<pwdj h> <pwdj l>..  
<pwd4 h> <pwd4 l> <pag h> <pag l>  
<data1 h> <data1 l>..  
<datak h> <datak l>..  
<data4 h> <data4 l> ETX <bcc> CR
```

Where:

i	1..4
<codei h> <codei l>	i-th byte of the code of the tag. ASCII encoded byte.
j	1..4
<pwdj h> <pwdj l>	j-th byte of the password of the tag. ASCII encoded byte
<pag h> <pag l>	Page to be written. ASCII encoded byte (0x00..0x07)
k	1..4
<datak h> <datak l>	k-th byte of the tag page. ASCII encoded byte

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

a) if a transponder is present and the page has been successfully written:
SOH <add h> <add l> STX "4" "B" "0" "0" ETX <bcc> CR

b) if errors occurred:
SOH <add h> <add l> STX "4" "B" "0" "2" ETX <bcc> CR

c) if no transponder is present:
SOH <add h> <add l> STX "4" "B" "0" "1" ETX <bcc> CR

2.50 "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.

The "master" sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace "4" "0" with "6" "0"):

```
SOH <add h> <add l> STX "4" "0" ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "4" "0" "0" "0" ETX <bcc> CR

- b) if errors occurred:

SOH <add h> <add l> STX "4" "0" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "4" "0" "0" "1" ETX <bcc> CR

2.51 "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.

The "master" sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace "4" "1" with "6" "1"):

SOH <add h> <add l> STX "4" "1" <pwd1 h> <pwd1 l>..<pwd4 l> ETX <bcc> CR

Where:

i	1..4
<pwdi h> <pwdi l>	i-th byte of the password to use. ASCII encoded byte

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "4" "1" "0" "0" ETX <bcc> CR

- b) if errors occurred:

SOH <add h> <add l> STX "4" "1" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "4" "1" "0" "1" ETX <bcc> CR

2.52 "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.

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “4” “2” with “6” “2”):

SOH <add h> <add l> STX “4” “2” <pwdoi h> <pwdoi l>..
 <pwdoi h> <pwdoi l>..
 <pwdn1 h> <pwdn1 l>..
 <pwdnj h> <pwdnj l>..
 <pwdn4 h> <pwdn4 l> ETX <bcc> CR

Where:

i	1..4
<pwdoi h> <pwdoi l>	i-th byte of the password to use. ASCII encoded byte
j	1..4
<pwdnj h> <pwdnj l>	j-th byte of the password to set. ASCII encoded byte

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “4” “2” “0” “0” ETX <bcc> CR

- b) if errors occurred:

SOH <add h> <add l> STX “4” “2” “0” “2” ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX “4” “2” “0” “1” ETX <bcc> CR

2.53 “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 resumed automatically when another type of command will be executed.

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “4” “3” with “6” “3”):

SOH <add h> <add l> STX “4” “3” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “4” “3” “0” “0” <data11 h> <data11 l>..
 ..<data11 h> <data11 l>..
 <data14 h> <data14 l>..
 <dataj1 h> <dataj1 l>..
 <dataj1 h> <dataj1 l>..
 <dataj4 h> <dataj4 l>..
 <datan1 h> <datan1 l>..
 <datani h> <datani l>..
 <datan4 h> <datan4 l> ETX
 <bcc> CR

Where:

i	1..4
j	1..n
n	Number of pages read

<dataj h> <dataj l>	i-th byte of the j-th page read. ASCII encoded byte
------------------------	---

- b) if errors occurred:
SOH <add h> <add l> STX "4" "3" "0" "2" ETX <bcc> CR
- c) if no transponder is present:
SOH <add h> <add l> STX "4" "3" "0" "1" ETX <bcc> CR

2.54 “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.

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “4” “4” with “6” “4”):

SOH <add h> <add l> STX "4" "4" <addu h> <addu l> <addp h> <addp l> ETX <bcc> CR

Where:

<addu h> <addu l>	Address of the first page to read. ASCII encoded byte
<addp h> <addp l>	Address of the last page to read. ASCII encoded byte

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:
- ```
SOH <add h> <add l> STX "4" "4" "0" "0" <data11 h> <data11 l>..
..<data14 h> <data14 l>..
..<dataj1 h> <dataj1 l>..
..<dataji h> <dataji l>..
..<dataj4 h> <dataj4 l>..
..<datan1 h> <datan1 l>..
..<datani h> <datani l>..
..<datan4 h> <datan4 l> ETX
<bcc> CR
```

Where:

|                                                |                                                     |
|------------------------------------------------|-----------------------------------------------------|
| i                                              | 1..4                                                |
| j                                              | 1..n                                                |
| n                                              | Number of pages read                                |
| <data <i>j</i> <i>i</i> ><br><data <i>j</i>  > | i-th byte of the j-th page read. ASCII encoded byte |

- b) if errors occurred:  
SOH <add h> <add l> STX "4" "4" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "4" "0" "1" ETX <bcc> CR

## 2.55 “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.

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “4” “5” with “6” “5”):

```
SOH <add h> <add l> STX “4” “5” <addw h> <addw l> <data1 h> <data1 l>..
<datai h> <datai l>..
<data4 h> <data4 l> ETX <bcc> CR
```

Where:

|                     |                                                         |
|---------------------|---------------------------------------------------------|
| <addw h> <addw l>   | Address of the page to be written. ASCII encoded byte   |
| i                   | 1..4                                                    |
| <datai h> <datai l> | i-th byte of the page to be written. ASCII encoded byte |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

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

```
SOH <add h> <add l> STX “4” “5” “0” “0” ETX <bcc> CR
```

- b) if errors occurred:

```
SOH <add h> <add l> STX “4” “5” “0” “2” ETX <bcc> CR
```

- c) if no transponder is present:

```
SOH <add h> <add l> STX “4” “5” “0” “1” ETX <bcc> CR
```

## 2.56 “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 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.

The “master” sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace “4” “6” with “6” “6”):

```
SOH <add h> <add l> STX “4” “6” <addw h> <addw l> <data11 h> <data11 l>..
<data1i h> <data1i l>..
<data14 h> <data14 l>..
<dataj1 h> <dataj1 l>..
<dataji h> <dataji l>..
<dataj4 h> <dataj4 l>..
<datan1 h> <datan1 l>..
<datani h> <datani l>..
<datan4 h> <datan4 l> ETX <bcc> CR
```

Where:

|                          |                                                              |
|--------------------------|--------------------------------------------------------------|
| <addw h> <addw l>        | Address of the page to be written. ASCII encoded byte        |
| i                        | 1..4                                                         |
| j                        | 1..n                                                         |
| n                        | Number of pages to be written                                |
| <dataji h><br><dataji l> | i-th byte of the j-th page to be written. ASCII encoded byte |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "6" "0" "0" ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX "4" "6" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "6" "0" "1" ETX <bcc> CR

## 2.57 "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 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.

The "master" sends the following command (for devices with 2 antennas, this command is valid to work with antenna no. 1; to work with antenna no. 2, replace "4" "7" with "6" "7"):

```
SOH <add h> <add l> STX "4" "7" <addw h> <addw l> <data1 h> <data1 l>..
<datai h> <datai l>..
<data4 h> <data4 l> ETX <bcc> CR
```

Where:

|                     |                                                         |
|---------------------|---------------------------------------------------------|
| <addw h> <addw l>   | Address of the page to be written. ASCII encoded byte   |
| i                   | 1..4                                                    |
| <datai h> <datai l> | i-th byte of the page to be written. ASCII encoded byte |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "7" "0" "0" <data1 h> <data1 l>..  
..  
<datai h> <datai l>..  
<data4 h> <data4 l> ETX <bcc> CR

Where:

|                     |                                                |
|---------------------|------------------------------------------------|
| i                   | 1..4                                           |
| <datai h> <datai l> | i-th byte of the read page. ASCII encoded byte |

- b) if errors occurred:  
SOH <add h> <add l> STX "4" "7" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "7" "0" "1" ETX <bcc> CR

## 2.58 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. For devices with 2 antennas, the command code 0x10 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x90 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “1” “0” with “9” “0”):

SOH <add h> <add l> STX “1” “0” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “1” “0” “0” “0” <UID11 h> <UID11 l>..<<UID1i h> <UID1i l>  
..<<UID18 h> <UID18 l>..<<UIDj1 h> <UIDj1 l>..<<UIDji h> <UIDji l> ..<UIDj8 h> <UIDj8  
l>..<<UIDn1 h> <UIDn1 l>..<<UIDni h> <UIDni l> ..<UIDn8 h> <UIDn8 l> ETX <bcc> CR

Where:

|                     |                                                                     |
|---------------------|---------------------------------------------------------------------|
| i                   | 1..8                                                                |
| j                   | 1..n                                                                |
| n                   | Number of identified tags                                           |
| <UIDji h> <UIDji l> | i-th byte of the UID of the j-th identified tag. ASCII encoded byte |

- b) if errors occurred:

SOH <add h> <add l> STX “1” “0” “0” “2” ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX “1” “0” “0” “1” ETX <bcc> CR

## 2.59 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 while the Fujitsu MB89R118 transponder is organized in blocks of 8 bytes. For more details see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x11 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x91 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “1” “1” with “9” “1”):

SOH <add h> <add l> STX “1” “1” <UID1 h> <UID1 l>..<<UIDi h> <UIDi l>..<<UID8 h>  
<UID8 l> <blk h> <blk l> ETX <bcc> CR

Where:

|                   |                                                      |
|-------------------|------------------------------------------------------|
| i                 | 1..8                                                 |
| <UIDi h> <UIDi l> | i-th byte of the UID of the tag. ASCII encoded byte. |
| <blk h> <blk l>   | Address of the block to read. ASCII encoded byte     |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “1” “1” “0” “0” <data1 h> <data1 l>..<<datai h> <datai l>..  
<datan h> <datan l> ETX <bcc> CR

Where:

|   |                                   |
|---|-----------------------------------|
| i | 1..n                              |
| n | Number of bytes in the read block |

|                     |                                                            |
|---------------------|------------------------------------------------------------|
| <datai h> <datai l> | i-th byte of the read block of the tag. ASCII encoded byte |
|---------------------|------------------------------------------------------------|

- b) if errors occurred:  
SOH <add h> <add l> STX "1" "1" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "1" "1" "0" "1" ETX <bcc> CR

## 2.60 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 while the Fujitsu MB89R118 transponder is organized in blocks of 8 bytes. For more details see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x12 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x92 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "1" "2" with "9" "2"):

```
SOH <add h> <add l> STX "1" "2" <UIDi h> <UIDi l>..

```

Where:

|                     |                                                      |
|---------------------|------------------------------------------------------|
| i                   | 1..8                                                 |
| <UIDi h> <UIDi l>   | i-th byte of the UID of the tag. ASCII encoded byte. |
| <blk h> <blk l>     | Address of the block to write. ASCII encoded byte    |
| j                   | 1..n                                                 |
| n                   | Number of bytes of the block to write                |
| <dataj h> <dataj l> | j-th byte of the page to write. ASCII encoded byte.  |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "1" "2" "0" "0" ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX "1" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "1" "2" "0" "1" ETX <bcc> CR

## 2.61 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 datasheet.

For devices with 2 antennas, the command code 0x13 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x93 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "1" "3" with "9" "3"):

```
SOH <add h> <add l> STX "1" "3" <UIDi h> <UIDi l>..

```



<UID8 l> <blk h> <blk l> ETX <bcc> CR

Where:

|                   |                                                      |
|-------------------|------------------------------------------------------|
| i                 | 1..8                                                 |
| <UIDi h> <UIDi l> | i-th byte of the UID of the tag. ASCII encoded byte. |
| <blk h> <blk l>   | Address of the block to write. ASCII encoded byte    |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "1" "3" "0" "0" ETX <bcc> CR

- b) if errors occurred:

SOH <add h> <add l> STX "1" "3" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "1" "3" "0" "1" ETX <bcc> CR

## 2.62 ISO 15693 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 datasheet.

For devices with 2 antennas, the command code 0x14 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x94 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "1" "4" with "9" "4"):

SOH <add h> <add l> STX "1" "4" <UID1 h> <UID1 l>..<<UIDi h> <UIDi l>..<<UID8 h> <UID8 l> ETX <bcc> CR

Where:

|                   |                                                      |
|-------------------|------------------------------------------------------|
| i                 | 1..8                                                 |
| <UIDi h> <UIDi l> | i-th byte of the UID of the tag. ASCII encoded byte. |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "1" "4" "0" "0" <flg h> <flg l> <UID1 h> <UID1 l>..<<UIDi h> <UIDi l>..<<UID8 h> <UID8 l> <dsf h> <dsf l> <afi h> <afi l> <msbs h> <msbs l> <msnb h> <msnb l> <icr h> <icr l> ETX <bcc> CR

Where:

|                 |                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <flg h> <flg l> | Info flags of the tag. ASCII encoded byte. Single bits are dedicated to specify the presence of the following fields (0 = absent, 1 = present): <ul style="list-style-type: none"> <li>• Bit 7..4: Not used</li> <li>• Bit 3: IC Reference (1 byte)</li> <li>• Bit 2: Memory Size (2 bytes)</li> <li>• Bit 1: AFI (1 byte)</li> <li>• Bit 0: DSFID (1 byte)</li> </ul> |
| i               | 1..8                                                                                                                                                                                                                                                                                                                                                                   |

|                   |                                                                                                                                             |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <UIDi h> <UIDi l> | i-th byte of the UID of the tag. ASCII encoded byte.                                                                                        |
| <dsf h> <dsf l>   | DSFID of the tag. ASCII encoded byte. Field present only if bit 0 of Info Flags is set                                                      |
| <afi h> <afi l>   | AFI of the tag. ASCII encoded byte. Field present only if bit 1 of Info Flags is set                                                        |
| <msbs h> <msbs l> | Memory Size – Block Size in bytes. ASCII encoded byte; 0x00 (1 byte) .. 0x1F (32 bytes)<br>Field present only if bit 2 of Info Flags is set |
| <msnb h> <msnb l> | Memory Size – Number of blocks. ASCII encoded byte; 0x00 (1 block) .. 0xFF (256 blocks)<br>Field present only if bit 2 of Info Flags is set |
| <icr h> <icr l>   | IC Reference. ASCII encoded byte. Field present only if bit 3 of Info Flags is set                                                          |

b) if errors occurred:

SOH <add h> <add l> STX “1” “4” “0” “2” ETX <bcc> CR

c) if no transponder is present:

SOH <add h> <add l> STX “1” “4” “0” “1” ETX <bcc> CR

## 2.63 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 an 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.

For devices with 2 antennas, the command code 0x15 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x95 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “1” “5” with “9” “5”):

SOH <add h> <add l> STX “1” “5” <data1 h> <data1 l>..  
<datai h> <datai l>..  
<datan h> <datan l> ETX <bcc> CR

Where:

|                     |                                                   |
|---------------------|---------------------------------------------------|
| i                   | 1..n                                              |
| n                   | Number of bytes to send to the tag                |
| <datai h> <datai l> | i-th byte to send to the tag. ASCII encoded byte. |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “1” “5” “0” “0” <data1 h> <data1 l>..  
<datai h> <datai l>..  
<datan h> <datan l> ETX <bcc> CR

Where:

|                     |                                                      |
|---------------------|------------------------------------------------------|
| i                   | 1..n                                                 |
| n                   | Number of bytes received from the tag                |
| <datai h> <datai l> | i-th byte received from the tag. ASCII encoded byte. |

b) if errors occurred:

SOH <add h> <add l> STX “1” “5” “0” “2” ETX <bcc> CR

c) if no transponder is present:

SOH <add h> <add l> STX “1” “5” “0” “1” ETX <bcc> CR

## 2.64 ISO 14443A Transponder “Inventory” Command

This command is used to get the UID code of an ISO 14443A transponder that is present near the antenna. For devices with 2 antennas, the command code 0x18 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x98 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “1” “8” with “9” “8”):

SOH <add h> <add l> STX “1” “8” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “1” “8” “0” “0” <type1 h> <type1 l> <UID11 h> <UID11 l>..  
<UID1i h> <UID1i l>......

Where:

|                     |                                                                     |
|---------------------|---------------------------------------------------------------------|
| i                   | 1..m (UID length)                                                   |
| j                   | 1..n                                                                |
| n                   | Number of identified tags                                           |
| <typej h> <typej l> | j-th transponder type                                               |
| <UIDji h> <UIDji l> | i-th byte of the UID of the j-th identified tag. ASCII encoded byte |

- b) if errors occurred:

SOH <add h> <add l> STX “1” “8” “0” “2” ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX “1” “8” “0” “1” ETX <bcc> CR

## 2.65 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.

For devices with 2 antennas, the command code 0x19 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x99 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “1” “9” with “9” “9”):

SOH <add h> <add l> STX “1” “9” <UID1 h> <UID1 l>..<UID4 l> <kt h> <kt l> <key1 h> <key1 l>..<blk l> ETX <bcc> CR

Where:

|                   |                                                                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| i                 | 1..4                                                                                                                                       |
| <UIDi h> <UIDi l> | i-th byte of the UID of the tag. ASCII encoded byte                                                                                        |
| <kt h> <kt l>     | Key type to access the tag’s memory. ASCII encoded byte <ul style="list-style-type: none"> <li>0x00: Key A</li> <li>0x01: Key B</li> </ul> |
| j                 | 1..6                                                                                                                                       |

|                   |                                                                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <keyj h> <keyj l> | j-th byte of the key. ASCII encoded byte                                                                                                                  |
| <blk h> <blk l>   | Memory block to read. ASCII encoded byte <ul style="list-style-type: none"> <li>• 0x00..0x3F for MIFARE 1k</li> <li>• 0x00..0xFF for MIFARE 4k</li> </ul> |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "1" "9" "0" "0" <data1 h> <data1 l>..<>datai h> <datai l>  
 ..<data16 h> <data16 l> ETX <bcc> CR

Where:

|                     |                                                 |
|---------------------|-------------------------------------------------|
| i                   | 1..16                                           |
| <datai h> <datai l> | i-th byte read from the tag. ASCII encoded byte |

- b) if errors occurred:

SOH <add h> <add l> STX "1" "9" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "1" "9" "0" "1" ETX <bcc> CR

## 2.66 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.

For devices with 2 antennas, the command code 0x1A is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x9A is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "1" "A" with "9" "A"):

SOH <add h> <add l> STX "1" "A" <UIDi h> <UIDi l>..<>UIDi h> <UIDi l>..<>UID4 h>  
 <UID4 l> <kt h> <kt l> <key1 h> <key1 l>..<>keyj h> <keyj l>..<>key6 h> <key6 l> <blk h>  
 <blk l> <data1 h> <data1 l>..<>datak h> <datak l>..<>data16 h> <data16 l> ETX <bcc> CR

Where:

|                     |                                                                                                                                                           |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                   | 1..4                                                                                                                                                      |
| <UIDi h> <UIDi l>   | i-th byte of the UID of the tag. ASCII encoded byte                                                                                                       |
| <kt h> <kt l>       | Key type to access the tag's memory. ASCII encoded byte <ul style="list-style-type: none"> <li>• 0x00: Key A</li> <li>• 0x01: Key B</li> </ul>            |
| j                   | 1..6                                                                                                                                                      |
| <keyj h> <keyj l>   | j-th byte of the key. ASCII encoded byte                                                                                                                  |
| <blk h> <blk l>     | Memory block to read. ASCII encoded byte <ul style="list-style-type: none"> <li>• 0x00..0x3F for MIFARE 1k</li> <li>• 0x00..0xFF for MIFARE 4k</li> </ul> |
| k                   | 1..16                                                                                                                                                     |
| <datak h> <datak l> | k-th byte read from the tag. ASCII encoded byte                                                                                                           |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "1" "A" "0" "0" ETX <bcc> CR

- b) if errors occurred:

SOH <add h> <add l> STX "1" "A" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "1" "A" "0" "1" ETX <bcc> CR

## 2.67 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.

For devices with 2 antennas, the command code 0x1D is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x9D is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "1" "D" with "9" "D"):

SOH <add h> <add l> STX "1" "D" <UID1 h> <UID1 l>..

Where:

|                   |                                                                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                 | 1..7                                                                                                                                                  |
| <UIDi h> <UIDi l> | i-th byte of the UID of the tag. ASCII encoded byte                                                                                                   |
| <kt h> <kt l>     | Key type to access the tag's memory. ASCII encoded byte <ul style="list-style-type: none"> <li>0x00: Key A</li> <li>0x01: Key B</li> </ul>            |
| j                 | 1..6                                                                                                                                                  |
| <keyj h> <keyj l> | j-th byte of the key. ASCII encoded byte                                                                                                              |
| <blk h> <blk l>   | Memory block to read. ASCII encoded byte <ul style="list-style-type: none"> <li>0x00..0x3F for MIFARE 1k</li> <li>0x00..0xFF for MIFARE 4k</li> </ul> |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "1" "D" "0" "0" <data1 h> <data1 l>..

Where:

|                     |                                                 |
|---------------------|-------------------------------------------------|
| i                   | 1..16                                           |
| <datai h> <datai l> | i-th byte read from the tag. ASCII encoded byte |

- b) if errors occurred:

SOH <add h> <add l> STX "1" "D" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "1" "D" "0" "1" ETX <bcc> CR

## 2.68 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.

For devices with 2 antennas, the command code 0x1E is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x9E is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “1” “E” with “9” “E”):

```
SOH <add h> <add l> STX “1” “A” <UID1 h> <UID1 l>..

```

Where:

|                     |                                                                                                                                                           |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                   | 1..7                                                                                                                                                      |
| <UIDi h> <UIDi l>   | i-th byte of the UID of the tag. ASCII encoded byte                                                                                                       |
| <kt h> <kt l>       | Key type to access the tag’s memory. ASCII encoded byte <ul style="list-style-type: none"> <li>• 0x00: Key A</li> <li>• 0x01: Key B</li> </ul>            |
| j                   | 1..6                                                                                                                                                      |
| <keyj h> <keyj l>   | j-th byte of the key. ASCII encoded byte                                                                                                                  |
| <blk h> <blk l>     | Memory block to read. ASCII encoded byte <ul style="list-style-type: none"> <li>• 0x00..0x3F for MIFARE 1k</li> <li>• 0x00..0xFF for MIFARE 4k</li> </ul> |
| k                   | 1..16                                                                                                                                                     |
| <datak h> <datak l> | k-th byte read from the tag. ASCII encoded byte                                                                                                           |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

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

```
SOH <add h> <add l> STX “1” “E” “0” “0” ETX <bcc> CR
```

- b) if errors occurred:

```
SOH <add h> <add l> STX “1” “E” “0” “2” ETX <bcc> CR
```

- c) if no transponder is present:

```
SOH <add h> <add l> STX “1” “E” “0” “1” ETX <bcc> CR
```

## 2.69 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.

For devices with 2 antennas, the command code 0x1B is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x9B is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “1” “B” with “9” “B”):

```
SOH <add h> <add l> STX “1” “B” <UID1 h> <UID1 l>..

```

Where:

|                   |                                                     |
|-------------------|-----------------------------------------------------|
| i                 | 1..7                                                |
| <UIDi h> <UIDi l> | i-th byte of the UID of the tag. ASCII encoded byte |

|                   |                                                    |
|-------------------|----------------------------------------------------|
| <page h> <page l> | Data page to read. ASCII encoded byte (0x00..0x0F) |
|-------------------|----------------------------------------------------|

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "1" "B" "0" "0" <data1 h> <data1 l>..  
 <datai h> <datai l>..  
 <data4 h> <data4 l> ETX <bcc> CR

Where:

|                     |                                                 |
|---------------------|-------------------------------------------------|
| i                   | 1..4                                            |
| <datai h> <datai l> | i-th byte read from the tag. ASCII encoded byte |

- b) if errors occurred:

SOH <add h> <add l> STX "1" "B" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "1" "B" "0" "1" ETX <bcc> CR

## 2.70 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.

For devices with 2 antennas, the command code 0x1C is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x9C is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "1" "C" with "9" "C"):

SOH <add h> <add l> STX "1" "C" <UID1 h> <UID1 l>..  
 <UIDi h> <UIDi l>..  
 <UID7 h> <UID7 l> <page h> <page l> <data1 h> <data1 l>..  
 <dataj h> <dataj l>..  
 <data4 h> <data4 l> ETX <bcc> CR

Where:

|                     |                                                     |
|---------------------|-----------------------------------------------------|
| i                   | 1..7                                                |
| <UIDi h> <UIDi l>   | i-th byte of the UID of the tag. ASCII encoded byte |
| <page h> <page l>   | Data page to write. ASCII encoded byte (0x00..0x0F) |
| j                   | 1..4                                                |
| <dataj h> <dataj l> | j-th byte read from the tag. ASCII encoded byte     |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "1" "C" "0" "0" ETX <bcc> CR

- e) if errors occurred:

SOH <add h> <add l> STX "1" "C" "0" "2" ETX <bcc> CR

- f) if no transponder is present:

SOH <add h> <add l> STX "1" "C" "0" "1" ETX <bcc> CR

## 2.71 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.

For devices with 2 antennas, the command code 0x1B is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x9B is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “1” “B” with “9” “B”):

```
SOH <add h> <add l> STX “1” “B” <UID1 h> <UID1 l>..

```

Where:

|                   |                                                                                                                                                                                      |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                 | 1..7                                                                                                                                                                                 |
| <UIDi h> <UIDi l> | i-th byte of the UID of the tag. ASCII encoded byte                                                                                                                                  |
| <page h> <page l> | Data page to read. ASCII encoded byte <ul style="list-style-type: none"> <li>• 0x00..0x2C for NTAG213</li> <li>• 0x00..0x86 for NTAG215</li> <li>• 0x00..0xE6 for NTAG216</li> </ul> |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

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

```
SOH <add h> <add l> STX “1” “B” “0” “0” <data1 h> <data1 l>..

```

Where:

|                     |                                                 |
|---------------------|-------------------------------------------------|
| i                   | 1..4                                            |
| <datai h> <datai l> | i-th byte read from the tag. ASCII encoded byte |

- b) if errors occurred:

```
SOH <add h> <add l> STX “1” “B” “0” “2” ETX <bcc> CR
```

- c) if no transponder is present:

```
SOH <add h> <add l> STX “1” “B” “0” “1” ETX <bcc> CR
```

## 2.72 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.

For devices with 2 antennas, the command code 0x1C is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0x9C is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “1” “C” with “9” “C”):

```
SOH <add h> <add l> STX “1” “C” <UID1 h> <UID1 l>..

```

Where:

|                   |                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------|
| i                 | 1..7                                                                                                              |
| <UIDi h> <UIDi l> | i-th byte of the UID of the tag. ASCII encoded byte                                                               |
| <page h> <page l> | Data page to write. ASCII encoded byte <ul style="list-style-type: none"> <li>• 0x00..0x2C for NTAG213</li> </ul> |



|                     |                                                                                                              |
|---------------------|--------------------------------------------------------------------------------------------------------------|
|                     | <ul style="list-style-type: none"> <li>• 0x00..0x86 for NTAG215</li> <li>• 0x00..0xE6 for NTAG216</li> </ul> |
| j                   | 1..4                                                                                                         |
| <dataj h> <dataj l> | j-th byte read from the tag. ASCII encoded byte                                                              |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "1" "C" "0" "0" ETX <bcc> CR

- b) if errors occurred:

SOH <add h> <add l> STX "1" "C" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "1" "C" "0" "1" ETX <bcc> CR

## 2.73 ISO 14443A-4 Transponder "RATS" Command

This command allows to select and send a RATS (Request for Answer To Select) command to an ISO 14443A-4 transponder, to switch in case of successful operation from ISO 14443A-3 level to ISO 14443A-4 level. If the "continuous" mode is enabled, the execution of this command will suspend "continuous" mode and the RF field left on. It will be suspended for as long as this command is used and 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 datasheet.

For devices with 2 antennas, the command code 0x40 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC0 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "0" with "C" "0"):

SOH <add h> <add l> STX "4" "0" <UID1 h> <UID1 l>..<UIDn l> ETX <bcc> CR

Where:

|                   |                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------|
| i                 | 1..n                                                                                                                     |
| n                 | UID length <ul style="list-style-type: none"> <li>• n = 7: MIFARE Desfire</li> <li>• n = 7: MIFARE Plus 2k/4k</li> </ul> |
| <UIDi h> <UIDi l> | i-th byte of the UID of the tag. ASCII encoded byte                                                                      |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "4" "0" "0" "0" <rats1 h> <rats1 l>....

Where:

|                     |                                                      |
|---------------------|------------------------------------------------------|
| i                   | 1..n                                                 |
| n                   | Number of bytes received from the tag                |
| <ratsi h> <ratsi l> | i-th byte received from the tag. ASCII encoded byte. |

- b) if errors occurred:  
SOH <add h> <add l> STX “4” “0” “0” “2” ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX “4” “0” “0” “1” ETX <bcc> CR

## 2.74 ISO 14443A-4 Transponder “Generic Command”

This command allows to send any ISO 14443A-4 general format protocol command to an 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 beforehand, using the RATS command described above. For more details, see the specific transponder datasheet. For devices with 2 antennas, the command code 0x41 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC1 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “1” with “C” “1”):

```
SOH <add h> <add l> STX “4” “1” <data1 h> <data1 l>..

```

Where:

|                     |                                                  |
|---------------------|--------------------------------------------------|
| i                   | 1..n                                             |
| n                   | Number of bytes to send to the tag               |
| <datai h> <datai l> | i-th byte to send to the tag. ASCII encoded byte |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX “4” “1” “0” “0” <data1 h> <data1 l>..

Where:

|                     |                                                      |
|---------------------|------------------------------------------------------|
| i                   | 1..n                                                 |
| n                   | Number of bytes received from the tag                |
| <datai h> <datai l> | i-th byte received from the tag. ASCII encoded byte. |

- b) if errors occurred:  
SOH <add h> <add l> STX “4” “1” “0” “2” ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX “4” “1” “0” “1” ETX <bcc> CR

## 2.75 MIFARE DESFire Transponder “Generic Command”

This command allows to send a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command described above. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” <data1 h> <data1 l>..

```

Where:

|                     |                                                  |
|---------------------|--------------------------------------------------|
| i                   | 1..n                                             |
| n                   | Number of bytes to send to the tag               |
| <datai h> <datai l> | i-th byte to send to the tag. ASCII encoded byte |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "4" "2" "0" "0" <data1 h> <data1 l>..  
<datai h> <datai l>..  
<datan h> <datan l> ETX <bcc> CR

Where:

|                     |                                                      |
|---------------------|------------------------------------------------------|
| i                   | 1..n                                                 |
| n                   | Number of bytes received from the tag                |
| <datai h> <datai l> | i-th byte received from the tag. ASCII encoded byte. |

- b) if errors occurred:

SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.1 MIFARE DESFire Transponder "Authenticate" Command

This command allows to send an Authenticate command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand using the RATS command. For more details see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "2" with "C" "2"):

SOH <add h> <add l> STX "4" "2" "0" "A" <keyno h> <keyno l> <keyver h> <keyver l>  
<keytype h> <keytype l> <key1 h> <key1 l>..  
<keyi h> <keyi l>..  
<keyn h> <keyn l>  
ETX <bcc> CR

Where:

|                            |                                                                                                                                    |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <keyno h><br><keyno l>     | The key number; depends on the currently selected application. See the transponder data sheet for more details. ASCII encoded byte |
| <keyver h><br><keyver l>   | The key version. ASCII encoded byte                                                                                                |
| <keytype h><br><keytype l> | The key type. ASCII encoded byte <ul style="list-style-type: none"> <li>0x03: Single DES</li> <li>0x04: 2 Key TDES</li> </ul>      |
| i                          | 1..n                                                                                                                               |
| n                          | Number of bytes of the key to use                                                                                                  |
| <keyi h> <keyi l>          | i-th byte of the key to use. ASCII encoded byte                                                                                    |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

## 2.75.2 MIFARE DESFire Transponder "AuthenticateISO" Command

This command allows to send an AuthenticateISO command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "2" with "C" "2"):

```
SOH <add h> <add l> STX "4" "2" "1" "A" <keyno h> <keyno l> <keyver h> <keyver l>
<keytype h> <keytype l> <key1 h> <key1 l>..<keyi h> <keyi l>..<keyn h> <keyn l>
ETX <bcc> CR
```

Where:

|                            |                                                                                                                                                               |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <keyno h><br><keyno l>     | The key number; depends on the currently selected application. See the transponder data sheet for more details. ASCII encoded byte                            |
| <keyver h><br><keyver l>   | The key version. ASCII encoded byte                                                                                                                           |
| <keytype h><br><keytype l> | The key type. ASCII encoded byte <ul style="list-style-type: none"> <li>• 0x03: Single DES</li> <li>• 0x04: 2 Key TDES</li> <li>• 0x05: 3 Key TDES</li> </ul> |
| i                          | 1..n                                                                                                                                                          |
| n                          | Number of bytes of the key to use                                                                                                                             |
| <keyi h> <keyi l>          | i-th byte of the key to use. ASCII encoded byte                                                                                                               |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.3 MIFARE DESFire Transponder “AuthenticateAES” Command

This command allows to send an AuthenticateAES command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “A” “A” <keyno h> <keyno l> <keyver h> <keyver l>
<keytype h> <keytype l> <key1 h> <key1 l>..

```

Where:

|                            |                                                                                                                                    |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <keyno h><br><keyno l>     | The key number; depends on the currently selected application. See the transponder data sheet for more details. ASCII encoded byte |
| <keyver h><br><keyver l>   | The key version. ASCII encoded byte                                                                                                |
| <keytype h><br><keytype l> | The key type. ASCII encoded byte <ul style="list-style-type: none"> <li>0x00: AES 128</li> </ul>                                   |
| i                          | 1..n                                                                                                                               |
| n                          | Number of bytes of the key to use                                                                                                  |
| <keyi h> <keyi l>          | i-th byte of the key to use. ASCII encoded byte                                                                                    |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

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

```
SOH <add h> <add l> STX “4” “2” “0” “0” “0” “0” ETX <bcc> CR
```

- b) if errors occurred:

```
SOH <add h> <add l> STX “4” “2” “0” “2” ETX <bcc> CR
```

- c) if no transponder is present:

```
SOH <add h> <add l> STX “4” “2” “0” “1” ETX <bcc> CR
```

### 2.75.4 MIFARE DESFire Transponder “FreeMem” Command

This command allows to send a FreeMem command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “6” “E” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
 SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" <memsize1 h> <memsize1 l>  
 <memsize2 h> <memsize2 l> <memsize3 h> <memsize3 l> ETX <bcc> CR

Where:

|                                |                                                         |
|--------------------------------|---------------------------------------------------------|
| <memsize1 h><br>..<memsize3 l> | Size of the free memory, LSB first. ASCII encoded bytes |
|--------------------------------|---------------------------------------------------------|

- b) if errors occurred:  
 SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
 SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.5 MIFARE DESFire Transponder "Format" Command

This command allows to send a Format command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "2" with "C" "2"):

SOH <add h> <add l> STX "4" "2" "F" "C" ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
 SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR
- b) if errors occurred:  
 SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
 SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.6 MIFARE DESFire Transponder "GetVersion" Command

This command allows to send a GetVersion command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "2" with "C" "2"):

SOH <add h> <add l> STX "4" "2" "6" "0" ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" <vinfo1 h> <vinfo1 l>..  
<vinfoi h> <vinfoi l>..  
<vinfon h> <vinfon l> ETX <bcc> CR

Where:

|                       |                                                   |
|-----------------------|---------------------------------------------------|
| i                     | 1..n                                              |
| n                     | Number of bytes of the version info               |
| <vinfoi h> <vinfoi l> | i-th byte of the version info. ASCII encoded byte |

- b) if errors occurred:

SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.7 MIFARE DESFire Transponder "ChangeKey" Command

This command allows to send a ChangeKey command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "2" with "C" "2"):

SOH <add h> <add l> STX "4" "2" "C" "4" <keyno h> <keyno l> <oldkver h> <oldkver l>  
<oldktype h> <oldktype l> <oldkey1 h> <oldkey1 l>..  
<oldkeyi h> <oldkeyi l>..  
<oldkeyn h> <oldkeyn l> <newkver h> <newkver l> <newktype h> <newktype l> <newkey1 h>  
<newkey1 l>..  
<newkeyj h> <newkeyj l>..  
<newkeym h> <newkeym l> ETX <bcc> CR

Where:

|                              |                                                                                                                                                                                    |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <keyno h><br><keyno l>       | The key number; depends on the currently selected application. See the transponder data sheet for more details. ASCII encoded byte                                                 |
| <oldkver h><br><oldkver l>   | The old key version. ASCII encoded byte                                                                                                                                            |
| <oldktype h><br><oldktype l> | The old key type. ASCII encoded byte <ul style="list-style-type: none"> <li>0x00: AES 128</li> <li>0x03: Single DES</li> <li>0x04: 2 Key TDES</li> <li>0x05: 3 Key TDES</li> </ul> |
| i                            | 1..n                                                                                                                                                                               |
| n                            | Number of bytes of the old key                                                                                                                                                     |
| <oldkeyi h><br><oldkeyi l>   | i-th byte of the old key. ASCII encoded byte                                                                                                                                       |
| <newkver h><br><newkver l>   | The new key version. ASCII encoded byte                                                                                                                                            |
| <newktype h><br><newktype l> | The new key type. ASCII encoded byte <ul style="list-style-type: none"> <li>0x00: AES 128</li> <li>0x03: Single DES</li> <li>0x04: 2 Key TDES</li> <li>0x05: 3 Key TDES</li> </ul> |
| j                            | 1..m                                                                                                                                                                               |

|                            |                                              |
|----------------------------|----------------------------------------------|
| m                          | Number of bytes of the new key               |
| <newkeyj h><br><newkeyj l> | i-th byte of the new key. ASCII encoded byte |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.8 MIFARE DESFire Transponder "ChangeKeySettings" Command

This command allows to send a ChangeKeySettings command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "2" with "C" "2"):

SOH <add h> <add l> STX "4" "2" "5" "4" <keyset h> <keyset l> ETX <bcc> CR

Where:

|                          |                                                                                                                          |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <keyset h><br><keyset l> | The key setting; depends on the selected application. See the transponder datasheet for more details. ASCII encoded byte |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------|

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.9 MIFARE DESFire Transponder "CreateApplication" Command

This command allows to send a CreateApplication command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.



The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “C” “A” <aid1 h> <aid1 l> <aid2 h> <aid2 l> <aid3 h>
<aid3 l> <keyset1 h> <keyset1 l> <keyset2 h> <keyset2 l> ETX <bcc> CR
```

Where:

|                            |                                                                                       |
|----------------------------|---------------------------------------------------------------------------------------|
| <aid1 h>..<<aid3 l>        | The application identifier, LSB first. ASCII encoded bytes                            |
| <keyset1 h><br><keyset1 l> | The key setting 1. See the transponder datasheet for more details. ASCII encoded byte |
| <keyset2 h><br><keyset2 l> | The key setting 2. See the transponder datasheet for more details. ASCII encoded byte |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX “4” “2” “0” “0” “0” “0” ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX “4” “2” “0” “2” ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX “4” “2” “0” “1” ETX <bcc> CR

#### 2.75.10 MIFARE DESFire Transponder “DeleteApplication” Command

This command allows to send a DeleteApplication command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “D” “A” <aid1 h> <aid1 l> <aid2 h> <aid2 l> <aid3 h>
<aid3 l> ETX <bcc> CR
```

Where:

|                     |                                                            |
|---------------------|------------------------------------------------------------|
| <aid1 h>..<<aid3 l> | The application identifier, LSB first. ASCII encoded bytes |
|---------------------|------------------------------------------------------------|

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX “4” “2” “0” “0” “0” “0” ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX “4” “2” “0” “2” ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX “4” “2” “0” “1” ETX <bcc> CR

### 2.75.11 MIFARE DESFire Transponder “SelectApplication” Command

This command allows to send a SelectApplication command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “5” “A” <aid1 h> <aid1 l> <aid2 h> <aid2 l> <aid3 h>
<aid3 l> ETX <bcc> CR
```

Where:

|                     |                                                            |
|---------------------|------------------------------------------------------------|
| <aid1 h>..<<aid3 l> | The application identifier, LSB first. ASCII encoded bytes |
|---------------------|------------------------------------------------------------|

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX “4” “2” “0” “0” “0” “0” ETX <bcc> CR
- if errors occurred:  
SOH <add h> <add l> STX “4” “2” “0” “2” ETX <bcc> CR
- if no transponder is present:  
SOH <add h> <add l> STX “4” “2” “0” “1” ETX <bcc> CR

### 2.75.12 MIFARE DESFire Transponder “CreateStdDataFile” Command

This command allows to send a CreateStdDataFile command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “C” “D” <fileid h> <fileid l> <comm h> <comm l>
<rights1 h> <rights1 l> <rights2 h> <rights2 l> <fsize1 h> <fsize1 l> <fsize2 h>
<fsize2 l> <fsize3 h> <fsize3 l> ETX <bcc> CR
```

Where:

|                               |                                                                                                                                                        |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <fileid h> <fileid l>         | The file identifier. ASCII encoded byte                                                                                                                |
| <comm h><br><comm l>          | The communication mode. ASCII encoded bytes: <ul style="list-style-type: none"> <li>• 0x00: Plain</li> <li>• 0x10: MAC</li> <li>• 0x30: ENC</li> </ul> |
| <rights1 h><br>.. <rights2 l> | The file access rights, LSB first. See the set access conditions table in the transponder datasheet for details. ASCII encoded bytes.                  |
| <fsize1 h><br>.. <fsize3 l>   | The file size in bytes, LSB first. ASCII encoded bytes.                                                                                                |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.13 MIFARE DESFire Transponder "CreateBackupDataFile" Command

This command allows to send a CreateBackupDataFile command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "2" with "C" "2"):

SOH <add h> <add l> STX "4" "2" "C" "B" <fileid h> <fileid l> <comm h> <comm l>  
<rights1 h> <rights1 l> <rights2 h> <rights2 l> <fsize1 h> <fsize1 l> <fsize2 h>  
<fsize2 l> <fsize3 h> <fsize3 l> ETX <bcc> CR

Where:

|                               |                                                                                                                                                        |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <fileid h> <fileid l>         | The file identifier. ASCII encoded byte                                                                                                                |
| <comm h><br><comm l>          | The communication mode. ASCII encoded bytes: <ul style="list-style-type: none"> <li>• 0x00: Plain</li> <li>• 0x10: MAC</li> <li>• 0x30: ENC</li> </ul> |
| <rights1 h><br>.. <rights2 l> | The file access rights, LSB first. See the set access conditions table in the transponder datasheet for details. ASCII encoded bytes.                  |
| <fsize1 h><br>.. <fsize3 l>   | The file size in bytes, LSB first. ASCII encoded bytes.                                                                                                |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.14 MIFARE DESFire Transponder “CreateValueFile” Command

This command allows to send a CreateValueFile command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “C” “C” <fileid h> <fileid l> <comm h> <comm l>
<rights1 h> <rights1 l> <rights2 h> <rights2 l> <low1 h> <low1 l>..<lowi h> <lowi l>..
<low4 h> <low4 l> <up1 h> <up1 l>..<upi h> <upi l>..<up4 h> <up4 l> <val1 h> <val1 l>
..<vali h> <vali l>..<val4 h> <val4 l> <lim h> <lim l> ETX <bcc> CR
```

Where:

|                               |                                                                                                                                                        |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <fileid h> <fileid l>         | The file identifier. ASCII encoded byte                                                                                                                |
| <comm h><br><comm l>          | The communication mode. ASCII encoded bytes: <ul style="list-style-type: none"> <li>• 0x00: Plain</li> <li>• 0x10: MAC</li> <li>• 0x30: ENC</li> </ul> |
| <rights1 h><br>.. <rights2 l> | The file access rights, LSB first. See the set access conditions table in the transponder datasheet for details. ASCII encoded bytes.                  |
| i                             | 1..4                                                                                                                                                   |
| <lowi h> <lowi l>             | Lower limit of the file, LSB first. ASCII encoded bytes                                                                                                |
| <upi h> <upi l>               | Upper limit of the file, LSB first. ASCII encoded bytes                                                                                                |
| <vali h> <vali l>             | Current value of the file, LSB first. ASCII encoded bytes                                                                                              |
| <lim h> <lim l>               | LimitedCredit and GetValue commands enable. See the transponder datasheet for details. ASCII encoded bytes                                             |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

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

```
SOH <add h> <add l> STX “4” “2” “0” “0” “0” “0” ETX <bcc> CR
```

- b) if errors occurred:

```
SOH <add h> <add l> STX “4” “2” “0” “2” ETX <bcc> CR
```

- c) if no transponder is present:

```
SOH <add h> <add l> STX “4” “2” “0” “1” ETX <bcc> CR
```

### 2.75.15 MIFARE DESFire Transponder “DeleteFile” Command

This command allows to send a DeleteFile command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “D” “F” <fileid h> <fileid l> ETX <bcc> CR
```

Where:

|                       |                                         |
|-----------------------|-----------------------------------------|
| <fileid h> <fileid l> | The file identifier. ASCII encoded byte |
|-----------------------|-----------------------------------------|

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR

- b) if errors occurred:

SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.16 MIFARE DESFire Transponder "ReadData" Command

This command allows to send a ReadData command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "2" with "C" "2"):

SOH <add h> <add l> STX "4" "2" "B" "D" <fileid h> <fileid l> <comm h> <comm l>  
 <offset1 h> <offset1 l> <offset2 h> <offset2 l> <offset3 h> <offset3 l> <length1 h>  
 <length1 l> <length2 h> <length2 l> <length3 h> <length3 l> ETX <bcc> CR

Where:

where:

|                               |                                                                                                                                                    |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <fileid h> <fileid l>         | The file identifier. ASCII encoded byte                                                                                                            |
| <comm h><br><comm l>          | The communication mode. ASCII encoded bytes: <ul style="list-style-type: none"><li>• 0x00: Plain</li><li>• 0x10: MAC</li><li>• 0x30: ENC</li></ul> |
| <offset1 h><br>..<<offset3 l> | The starting position for the read operation, LSB first. ASCII encoded bytes                                                                       |
| <length1 h><br>..<<length3 l> | The number of bytes to read, LSB first. ASCII encoded bytes                                                                                        |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" <data1 h> <data1 l>..
 <datai l>..

Where:

|                     |                                    |
|---------------------|------------------------------------|
| i                   | 1..n                               |
| n                   | Number of bytes read               |
| <datai h> <datai l> | i-th byte read. ASCII encoded byte |

- b) if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.17 MIFARE DESFire Transponder "WriteData" Command

This command allows to send a WriteData command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "2" with "C" "2"):

```
SOH <add h> <add l> STX "4" "2" "3" "D" <fileid h> <fileid l> <comm h> <comm l>
<offset1 h> <offset1 l> <offset2 h> <offset2 l> <offset3 h> <offset3 l> <length1 h>
<length1 l> <length2 h> <length2 l> <length3 h> <length3 l> <data1 h> <data1 l>..
<datai h> <datai l>..<datan h> <datan l> ETX <bcc> CR
```

Where:

|                               |                                                                                                                                                        |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <fileid h> <fileid l>         | The file identifier. ASCII encoded byte                                                                                                                |
| <comm h><br><comm l>          | The communication mode. ASCII encoded bytes: <ul style="list-style-type: none"> <li>• 0x00: Plain</li> <li>• 0x10: MAC</li> <li>• 0x30: ENC</li> </ul> |
| <offset1 h><br>..<<offset3 l> | The starting position for the read operation, LSB first. ASCII encoded bytes                                                                           |
| <length1 h><br>..<<length3 l> | The number of bytes to read, LSB first. ASCII encoded bytes                                                                                            |
| i                             | 1..n                                                                                                                                                   |
| n                             | Number of bytes read                                                                                                                                   |
| <datai h> <datai l>           | i-th byte read. ASCII encoded byte                                                                                                                     |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.18 MIFARE DESFire Transponder “GetValue” Command

This command allows to send a GetValue command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “6” “C” <fileid h> <fileid l> <comm h> <comm l>
ETX <bcc> CR
```

Where:

|                       |                                                                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <fileid h> <fileid l> | The file identifier. ASCII encoded byte                                                                                                                |
| <comm h><br><comm l>  | The communication mode. ASCII encoded bytes: <ul style="list-style-type: none"> <li>• 0x00: Plain</li> <li>• 0x10: MAC</li> <li>• 0x30: ENC</li> </ul> |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

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

```
SOH <add h> <add l> STX “4” “2” “0” “0” “0” “0” <val1 h> <val1 l>..
<vali h> <vali l>..
<val4 h> <val4 l> ETX <bcc> CR
```

Where:

|                   |                                                           |
|-------------------|-----------------------------------------------------------|
| i                 | 1..4                                                      |
| <vali h> <vali l> | Current value of the file, LSB first. ASCII encoded bytes |

- b) if errors occurred:

```
SOH <add h> <add l> STX “4” “2” “0” “2” ETX <bcc> CR
```

- c) if no transponder is present:

```
SOH <add h> <add l> STX “4” “2” “0” “1” ETX <bcc> CR
```

### 2.75.19 MIFARE DESFire Transponder “Credit” Command

This command allows to send a Credit command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “0” “C” <fileid h> <fileid l> <comm h> <comm l>
<data1 h> <data1 l>..
<datai h> <datai l>..
<data4 h> <data4 l> ETX <bcc> CR
```

Where:

|                       |                                                                                                                                   |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <fileid h> <fileid l> | The file identifier. ASCII encoded byte                                                                                           |
| <comm h><br><comm l>  | The communication mode. ASCII encoded bytes: <ul style="list-style-type: none"> <li>• 0x00: Plain</li> <li>• 0x10: MAC</li> </ul> |

|                     |                                                               |
|---------------------|---------------------------------------------------------------|
|                     | <ul style="list-style-type: none"> <li>• 0x30: ENC</li> </ul> |
| i                   | 1..4                                                          |
| <datai h> <datai l> | The value to be credited, LSB first. ASCII encoded byte       |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR
- if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.20 MIFARE DESFire Transponder "LimitedCredit" Command

This command allows to send a LimitedCredit command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "2" with "C" "2"):

SOH <add h> <add l> STX "4" "2" "1" "C" <fileid h> <fileid l> <comm h> <comm l>  
<data1 h> <data1 l>..<>datai h> <datai l>..<>data4 h> <data4 l> ETX <bcc> CR

Where:

|                       |                                                                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <fileid h> <fileid l> | The file identifier. ASCII encoded byte                                                                                                                |
| <comm h><br><comm l>  | The communication mode. ASCII encoded bytes: <ul style="list-style-type: none"> <li>• 0x00: Plain</li> <li>• 0x10: MAC</li> <li>• 0x30: ENC</li> </ul> |
| i                     | 1..4                                                                                                                                                   |
| <datai h> <datai l>   | The value to be credited, LSB first. ASCII encoded byte                                                                                                |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR
- if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR



### 2.75.21 MIFARE DESFire Transponder “Debit” Command

This command allows to send a Debit command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “D” “C” <fileid h> <fileid l> <comm h> <comm l>
<data1 h> <data1 l>..<datai h> <datai l>..<data4 h> <data4 l> ETX <bcc> CR
```

Where:

|                       |                                                                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <fileid h> <fileid l> | The file identifier. ASCII encoded byte                                                                                                                |
| <comm h><br><comm l>  | The communication mode. ASCII encoded bytes: <ul style="list-style-type: none"> <li>• 0x00: Plain</li> <li>• 0x10: MAC</li> <li>• 0x30: ENC</li> </ul> |
| i                     | 1..4                                                                                                                                                   |
| <datai h> <datai l>   | The value to be debited, LSB first. ASCII encoded byte                                                                                                 |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

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

```
SOH <add h> <add l> STX “4” “2” “0” “0” “0” “0” ETX <bcc> CR
```

- b) if errors occurred:

```
SOH <add h> <add l> STX “4” “2” “0” “2” ETX <bcc> CR
```

- c) if no transponder is present:

```
SOH <add h> <add l> STX “4” “2” “0” “1” ETX <bcc> CR
```

### 2.75.22 MIFARE DESFire Transponder “CommitTransaction” Command

This command allows to send a CommitTransaction command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The “master” sends the following command (to work with antenna no. 2, replace “4” “2” with “C” “2”):

```
SOH <add h> <add l> STX “4” “2” “C” “7” ETX <bcc> CR
```

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

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

```
SOH <add h> <add l> STX “4” “2” “0” “0” “0” “0” ETX <bcc> CR
```

- b) if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.75.23 MIFARE DESFire Transponder "AbortTransaction" Command

This command allows to send a AbortTranscation command to a MIFARE DESFire transponder and to receive, in case of successful operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level beforehand, using the RATS command. For more details, see the specific transponder datasheet.

For devices with 2 antennas, the command code 0x42 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "2" with "C" "2"):

SOH <add h> <add l> STX "4" "2" "A" "7" ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "4" "2" "0" "0" "0" "0" ETX <bcc> CR
- b) if errors occurred:  
SOH <add h> <add l> STX "4" "2" "0" "2" ETX <bcc> CR
- c) if no transponder is present:  
SOH <add h> <add l> STX "4" "2" "0" "1" ETX <bcc> CR

### 2.76 ISO 14443B Transponder "Inventory" Command

This command is used to get the UID code of an ISO 14443B transponder that is present near the antenna.

For devices with 2 antennas, the command code 0x20 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xA0 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "2" "0" with "A" "0"):

SOH <add h> <add l> STX "2" "0" ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "2" "0" "0" "0" <type1 h> <type1 l> <UID11 h> <UID11 l>..  
<UID1i h> <UID1i l>..  
<UID1n h> <UID1n l>..  
<typej h> <typej l> <UIDj1 h> <UIDj1 l>..  
<UIDji h> <UIDji l>..  
<UIDjn h> <UIDjn l>..  
<typem h> <typem l> <UIDm1 h> <UIDm1 l>..  
<UIDmi h> <UIDmi l>..  
<UIDmn h> <UIDmn l> ETX <bcc> CR

Where:

|   |                   |
|---|-------------------|
| i | 1..n (UID length) |
| j | 1..m              |

|                     |                                                                     |
|---------------------|---------------------------------------------------------------------|
| m                   | Number of identified tags                                           |
| <typej h> <typej l> | j-th transponder type                                               |
| <UIDji h> <UIDji l> | i-th byte of the UID of the j-th identified tag. ASCII encoded byte |

b) if errors occurred:

SOH <add h> <add l> STX "2" "0" "0" "2" ETX <bcc> CR

c) if no transponder is present:

SOH <add h> <add l> STX "2" "0" "0" "1" ETX <bcc> CR

## 2.77 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 datasheet.

For devices with 2 antennas, the command code 0x21 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xA1 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "2" "1" with "A" "1"):

SOH <add h> <add l> STX "2" "1" <UID1 h> <UID1 l>..<UID8 l> <blk h> <blk l> ETX <bcc> CR

Where:

|                   |                                                     |
|-------------------|-----------------------------------------------------|
| i                 | 1..8                                                |
| <UIDi h> <UIDi l> | i-th byte of the UID of the tag. ASCII encoded byte |
| <blk h> <blk l>   | Data block to read. ASCII encoded byte (0x00..0x0F) |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "2" "1" "0" "0" <data1 h> <data1 l>..ETX <bcc> CR

Where:

|                                                                                                            |                                              |
|------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| <data1 h><br>.. <lt;data2 l&gt;<="" td=""><td>Bytes read from the tag. ASCII encoded byte.</td></lt;data2> | Bytes read from the tag. ASCII encoded byte. |
|------------------------------------------------------------------------------------------------------------|----------------------------------------------|

b) if errors occurred:

SOH <add h> <add l> STX "2" "1" "0" "2" ETX <bcc> CR

c) if no transponder is present:

SOH <add h> <add l> STX "2" "1" "0" "1" ETX <bcc> CR

## 2.78 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 datasheet.

For devices with 2 antennas, the command code 0x22 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xA2 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "2" "2" with "A" "2"):

SOH <add h> <add l> STX "2" "2" <UID1 h> <UID1 l>..

<UID8 l> <data1 h> <data1 l>..<<data2 h> <data2 l> <blk h> <blk l> ETX <bcc> CR

Where:

|                           |                                                     |
|---------------------------|-----------------------------------------------------|
| i                         | 1..8                                                |
| <UIDi h> <UIDi l>         | i-th byte of the UID of the tag. ASCII encoded byte |
| <blk h> <blk l>           | Data block to read. ASCII encoded byte (0x00..0x0F) |
| <data1 h><br>..<<data2 l> | Bytes read from the tag. ASCII encoded byte.        |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "2" "2" "0" "0" ETX <bcc> CR

- b) if errors occurred:

SOH <add h> <add l> STX "2" "2" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "2" "2" "0" "1" ETX <bcc> CR

## 2.79 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. For devices with 2 antennas, the command code 0x48 is used to work with antenna no. 1 (same for devices with 1 antenna), while the command code 0xC8 is used to work with antenna no. 2.

The "master" sends the following command (to work with antenna no. 2, replace "4" "8" with "C" "8"):

SOH <add h> <add l> STX "4" "8" ETX <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "4" "8" "0" "0" <UID11 h> <UID11 l>..  
 <UID1i h> <UID1i l>..<<UID18 h> <UID18 l>..<<UIDj1 h> <UIDj1 l>..<<UIDji h> <UIDji l>  
 ..<UIDj8 h> <UIDj8 l>..<<UIDm1 h> <UIDm1 l>..<<UIDmi h> <UIDmi l>..<<UIDm8 h> <UIDm8 l>  
 ETX <bcc> CR

Where:

|                     |                                                                     |
|---------------------|---------------------------------------------------------------------|
| i                   | 1..8                                                                |
| j                   | 1..m                                                                |
| m                   | Number of identified tags                                           |
| <UIDji h> <UIDji l> | i-th byte of the UID of the j-th identified tag. ASCII encoded byte |

- b) if errors occurred:

SOH <add h> <add l> STX "4" "8" "0" "2" ETX <bcc> CR

- c) if no transponder is present:

SOH <add h> <add l> STX "4" "8" "0" "1" ETX <bcc> CR

## 2.80 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.

The “master” sends the following command:

SOH <add h> <add l> STX “1” “8” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “1” “8” “0” “0” <ID11 h> <ID11 l>..  
 <ID1i h> <ID1i l>..  
 <ID1m h> <ID1m l> “0” <ant1>..  
 <IDj1 h> <IDj1 l>..  
 <IDji h> <IDji l>..  
 <IDjm h> <IDjm l> “0” <antj>..  
 <IDn1 h> <IDn1 l>..  
 <IDni h> <IDni l>..  
 <IDnm h> <IDnm l> “0” <ant n> ETX <bcc> CR

Where:

|                   |                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                 | 1..m                                                                                                                                                                                                                                                                                                                                                                      |
| m                 | ID length                                                                                                                                                                                                                                                                                                                                                                 |
| j                 | 1..n                                                                                                                                                                                                                                                                                                                                                                      |
| n                 | Number of identified tags                                                                                                                                                                                                                                                                                                                                                 |
| <IDji h> <IDji l> | i-th byte of the ID of the j-th identified tag. ASCII encoded byte                                                                                                                                                                                                                                                                                                        |
| <ant j>           | Reading antenna for the j-th identified tag (optional parameter present only if the reading antenna information flag in the general parameters is active, see the reader user manual for more info).<br>ASCII character: <ul style="list-style-type: none"> <li>• “1”: Antenna 1</li> <li>• “2”: Antenna 2</li> <li>• “3”: Antenna 3</li> <li>• “4”: Antenna 4</li> </ul> |

- b) if errors occurred during the transaction:

SOH <add h> <add l> STX “1” “8” “0” “2” ETX <bcc> CR

- c) if no tag is present:

SOH <add h> <add l> STX “1” “8” “0” “1” ETX <bcc> CR

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

SOH <add h> <add l> STX “1” “8” “0” “1” ETX <bcc> CR

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “1” “8” “0” “0” <ID11 h> <ID11 l>..  
 <ID1i h> <ID1i l>..  
 <ID1m h> <ID1m l> <RSSI Q1 h> <RSSI Q1 l> <RSSI I1 h> <RSSI I1 l> “0” <ant1>..  
 <IDj1 h> <IDj1 l>..  
 <IDji h> <IDji l>..  
 <IDjm h> <IDjm l> <RSSI Qj h> <RSSI Qj l>..  
 <IDni h> <IDni l>..  
 <IDnm h> <IDnm l> <RSSI Ij h> <RSSI Ij l> “0” <antj>..  
 <IDn1 h> <IDn1 l>..  
 <IDni h> <IDni l>..  
 <IDnm h> <IDnm l> “0” <ant n> ETX <bcc> CR

<IDnm l> "0" <RSSI Qn h> <RSSI Qn l> <RSSI In h> <RSSI In l> <ant n> ETX <bcc> CR

Where:

|                            |                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                          | 1..m                                                                                                                                                                                                                                                                                                                                                                      |
| m                          | ID length                                                                                                                                                                                                                                                                                                                                                                 |
| j                          | 1..n                                                                                                                                                                                                                                                                                                                                                                      |
| n                          | Number of identified tags                                                                                                                                                                                                                                                                                                                                                 |
| <IDji h> <IDji l>          | i-th byte of the ID of the j-th identified tag. ASCII encoded byte                                                                                                                                                                                                                                                                                                        |
| <RSSI Qj h><br><RSSI Qj l> | The RSSI Q-channel value in dB of the j-th identified tag. ASCII encoded byte                                                                                                                                                                                                                                                                                             |
| <RSSI Ij h><br><RSSI Ij l> | The RSSI I-channel value in dB of the j-th identified tag. ASCII encoded byte                                                                                                                                                                                                                                                                                             |
| <ant j>                    | Reading antenna for the j-th identified tag (optional parameter present only if the reading antenna information flag in the general parameters is active, see the reader user manual for more info).<br>ASCII character: <ul style="list-style-type: none"> <li>• "1": Antenna 1</li> <li>• "2": Antenna 2</li> <li>• "3": Antenna 3</li> <li>• "4": Antenna 4</li> </ul> |

- b) if errors occurred during the transaction:

SOH <add h> <add l> STX "1" "8" "0" "2" ETX <bcc> CR

- c) if no tag is present:

SOH <add h> <add l> STX "1" "8" "0" "1" ETX <bcc> CR

## 2.81 Program EPC of an ISO 18000-63 Transponder

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

The "master" sends the following command:

SOH <add h> <add l> STX "1" "E" <ID1 h> <ID1 l>..<pwd1 h> <pwd1 l>..<data21 h> <data21 l>..<data1n l> <data2n h> <data2n l> ETX <bcc> CR

Where:

|                          |                                                                                                                       |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------|
| i                        | 1..m                                                                                                                  |
| m                        | ID length                                                                                                             |
| <IDi h> <IDi l>          | i-th byte of the ID of the tag. ASCII encoded byte                                                                    |
| j                        | 1..4                                                                                                                  |
| <pwdj h><br><pwdj l>     | j-th byte of the tag access password. ASCII encoded byte. Use a "0" password if the access password is not requested. |
| k                        | 1..n                                                                                                                  |
| n                        | Number of blocks to be written, range 0..31                                                                           |
| <data1k h><br><data1k l> | First byte of the k-th block to be written. ASCII encoded byte                                                        |
| <data2k h><br><data2k l> | Second byte of the k-th block to be written. ASCII encoded byte                                                       |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "1" "E" "0" "0" ETX <bcc> CR
- b) if errors occurred during the transaction:  
SOH <add h> <add l> STX "1" "E" "0" "2" ETX <bcc> CR
- c) if no tag is present:  
SOH <add h> <add l> STX "1" "E" "0" "1" ETX <bcc> CR

## 2.82 Read Data of an ISO 18000-63 Transponder

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

The "master" sends the following command:

```
SOH <add h> <add l> STX "1" "9" <ID1 h> <ID1 l>..

```

Where:

|                        |                                                                                                                                                                         |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                      | 1..m                                                                                                                                                                    |
| m                      | ID length                                                                                                                                                               |
| <IDi h> <IDi l>        | i-th byte of the ID of the tag. ASCII encoded byte                                                                                                                      |
| j                      | 1..4                                                                                                                                                                    |
| <pwdj h><br><pwdj l>   | j-th byte of the tag access password. ASCII encoded byte. Use a "0" password if the access password is not requested.                                                   |
| <bank>                 | Memory bank to be read. ASCII character: <ul style="list-style-type: none"> <li>• "0": Reserved</li> <li>• "1": EPC</li> <li>• "2": TID</li> <li>• "3": User</li> </ul> |
| <saddj h><br><saddj l> | j-th byte of the address of the 1st byte of the 1st block to be read. ASCII encoded byte                                                                                |
| <nblk h> <nblk l>      | Number of blocks to read. ASCII encoded byte (1..64)                                                                                                                    |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "1" "9" "0" "0" <data11 h> <data11 l> <data21 h> <data21 l>..<data2n l> ETX <bcc> CR

Where:

|                          |                                                        |
|--------------------------|--------------------------------------------------------|
| k                        | 1..n                                                   |
| n                        | Number of blocks to be written, range 0..31            |
| <data1k h><br><data1k l> | First byte of the k-th block read. ASCII encoded byte  |
| <data2k h><br><data2k l> | Second byte of the k-th block read. ASCII encoded byte |

- b) if errors occurred during the transaction:  
SOH <add h> <add l> STX "1" "9" "0" "2" ETX <bcc> CR
- c) if no tag is present:  
SOH <add h> <add l> STX "1" "9" "0" "1" ETX <bcc> CR

## 2.83 Write Data of an ISO 18000-63 Transponder

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

The "master" sends the following command:

```
SOH <add h> <add l> STX "1" "A" <ID1 h> <ID1 l>...<IDi h> <IDi l>...<IDm h> <IDm l>
<pwd1 h> <pwd1 l>...<pwdj h> <pwdj l>...<pwd4 h> <pwd4 l> "0" <bank> <sadd1 h> <sadd1 l>
...<saddj h> <saddj l>...<sadd4 h> <sadd4 l> <nblk h> <nblk l> <data11 h> <data11 l>
<data21 h> <data21 l>...<data1k h> <data1k l> <data2k h> <data2k l> <data1n h>
<data1n l> <data2n h> <data2n l> ETX <bcc> CR
```

Where:

|                          |                                                                                                                                                                         |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                        | 1..m                                                                                                                                                                    |
| m                        | ID length                                                                                                                                                               |
| <IDi h> <IDi l>          | i-th byte of the ID of the tag. ASCII encoded byte                                                                                                                      |
| j                        | 1..4                                                                                                                                                                    |
| <pwdj h><br><pwdj l>     | j-th byte of the tag access password. ASCII encoded byte. Use a "0" password if the access password is not requested.                                                   |
| <bank>                   | Memory bank to be read. ASCII character: <ul style="list-style-type: none"> <li>• "0": Reserved</li> <li>• "1": EPC</li> <li>• "2": TID</li> <li>• "3": User</li> </ul> |
| <saddj h><br><saddj l>   | j-th byte of the address of the 1st byte of the 1st block to be read. ASCII encoded byte                                                                                |
| <nblk h> <nblk l>        | Number of blocks to read. ASCII encoded byte (1..64)                                                                                                                    |
| k                        | 1..n                                                                                                                                                                    |
| n                        | Number of blocks to be written, range 0..31                                                                                                                             |
| <data1k h><br><data1k l> | First byte of the k-th block to be written. ASCII encoded byte                                                                                                          |
| <data2k h><br><data2k l> | Second byte of the k-th block to be written. ASCII encoded byte                                                                                                         |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "1" "A" "0" "0" ETX <bcc> CR
- b) if errors occurred during the transaction:  
SOH <add h> <add l> STX "1" "A" "0" "2" ETX <bcc> CR
- c) if no tag is present:



SOH <add h> <add l> STX “1” “A” “0” “1” ETX <bcc> CR

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:

The “master” sends the following command:

SOH <add h> <add l> STX “1” “D” <ID1 h> <ID1 l>..
 <pwd1 h> <pwd1 l>..
 ..<saddj h> <saddj l>..
 <data21 h> <data21 l>..
 <data1n l> <data2n h> <data2n l> ETX <bcc> CR

Where:

|                          |                                                                                                                                                                         |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                        | 1..m                                                                                                                                                                    |
| m                        | ID length                                                                                                                                                               |
| <IDi h> <IDi l>          | i-th byte of the ID of the tag. ASCII encoded byte                                                                                                                      |
| j                        | 1..4                                                                                                                                                                    |
| <pwdj h><br><pwdj l>     | j-th byte of the tag access password. ASCII encoded byte. Use a “0” password if the access password is not requested.                                                   |
| <bank>                   | Memory bank to be read. ASCII character: <ul style="list-style-type: none"> <li>• “0”: Reserved</li> <li>• “1”: EPC</li> <li>• “2”: TID</li> <li>• “3”: User</li> </ul> |
| <saddj h><br><saddj l>   | j-th byte of the address of the 1st byte of the 1st block to be read. ASCII encoded byte                                                                                |
| <nblk h> <nblk l>        | Number of blocks to read. ASCII encoded byte (1..64)                                                                                                                    |
| k                        | 1..n                                                                                                                                                                    |
| n                        | Number of blocks to be written, range 0..31                                                                                                                             |
| <data1k h><br><data1k l> | First byte of the k-th block to be written. ASCII encoded byte                                                                                                          |
| <data2k h><br><data2k l> | Second byte of the k-th block to be written. ASCII encoded byte                                                                                                         |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “1” “D” “0” “0” ETX <bcc> CR

- e) if errors occurred during the transaction:

SOH <add h> <add l> STX “1” “D” “0” “2” ETX <bcc> CR

- f) if no tag is present:

SOH <add h> <add l> STX “1” “D” “0” “1” ETX <bcc> CR

## 2.84 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.

The “master” sends the following command:

```
SOH <add h> <add l> STX “1” “B” <ID1 h> <ID1 l>..

```

Where:

|                      |                                                                                                                                                                                                                                                                                                                                                    |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                    | 1..m                                                                                                                                                                                                                                                                                                                                               |
| m                    | ID length                                                                                                                                                                                                                                                                                                                                          |
| <IDi h> <IDi l>      | i-th byte of the ID of the tag. ASCII encoded byte                                                                                                                                                                                                                                                                                                 |
| j                    | 1..4                                                                                                                                                                                                                                                                                                                                               |
| <pwdj h><br><pwdj l> | j-th byte of the tag access password. ASCII encoded byte. Use a “0” password if the access password is not requested.                                                                                                                                                                                                                              |
| <kill>               | Kill password lock property. ASCII character: <ul style="list-style-type: none"> <li>• “0”: Accessible from all states</li> <li>• “1”: Permanently accessible from all states and may never be locked</li> <li>• “2”: Accessible only from the secured state</li> <li>• “3”: Not accessible from any state</li> <li>• “4”: No change</li> </ul>    |
| <acc>                | Access password lock property. ASCII character: <ul style="list-style-type: none"> <li>• “0”: Accessible from all states</li> <li>• “1”: Permanently accessible from all states and may never be locked</li> <li>• “2”: Accessible only from the secured state</li> <li>• “3”: Not accessible from any state</li> <li>• “4”: No change</li> </ul>  |
| <EPC>                | EPC memory bank lock property. ASCII character: <ul style="list-style-type: none"> <li>• “0”: Accessible from all states</li> <li>• “1”: Permanently accessible from all states and may never be locked</li> <li>• “2”: Accessible only from the secured state</li> <li>• “3”: Not accessible from any state</li> <li>• “4”: No change</li> </ul>  |
| <TID>                | TID memory bank lock property. ASCII character: <ul style="list-style-type: none"> <li>• “0”: Accessible from all states</li> <li>• “1”: Permanently accessible from all states and may never be locked</li> <li>• “2”: Accessible only from the secured state</li> <li>• “3”: Not accessible from any state</li> <li>• “4”: No change</li> </ul>  |
| <user>               | User memory bank lock property. ASCII character: <ul style="list-style-type: none"> <li>• “0”: Accessible from all states</li> <li>• “1”: Permanently accessible from all states and may never be locked</li> <li>• “2”: Accessible only from the secured state</li> <li>• “3”: Not accessible from any state</li> <li>• “4”: No change</li> </ul> |

If the addressed Bluebox cannot execute the command, it answers:

```
SOH <add h> <add l> NAK <bcc> CR
```

Otherwise, it answers with:

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

```
SOH <add h> <add l> STX “1” “B” “0” “0” ETX <bcc> CR
```

- b) if errors occurred during the transaction:  
SOH <add h> <add l> STX "1" "B" "0" "2" ETX <bcc> CR
- c) if no tag is present:  
SOH <add h> <add l> STX "1" "B" "0" "1" ETX <bcc> CR

## 2.85 "Kill" Command of an ISO 18000-63 Transponder

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

The "master" sends the following command:

SOH <add h> <add l> STX "1" "C" <ID1 h> <ID1 l>..<pwd1 h> <pwd1 l>..

Where:

|                      |                                                           |
|----------------------|-----------------------------------------------------------|
| i                    | 1..m                                                      |
| m                    | ID length                                                 |
| <IDi h> <IDi l>      | i-th byte of the ID of the tag. ASCII encoded byte        |
| j                    | 1..4                                                      |
| <pwdj h><br><pwdj l> | j-th byte of the tag's kill password. ASCII encoded byte. |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

- a) if a transponder is present and the command has been successfully executed:  
SOH <add h> <add l> STX "1" "C" "0" "0" ETX <bcc> CR
- b) if errors occurred during the transaction:  
SOH <add h> <add l> STX "1" "C" "0" "2" ETX <bcc> CR
- c) if no tag is present:  
SOH <add h> <add l> STX "1" "C" "0" "1" ETX <bcc> CR

## 2.86 "QT Read" Command of a Monza 4QT Transponder

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

The "master" sends the following command:

SOH <add h> <add l> STX "2" "0" <ID1 h> <ID1 l>..<pwd1 h> <pwd1 l>..

Where:

|                      |                                                                                                                      |
|----------------------|----------------------------------------------------------------------------------------------------------------------|
| i                    | 1..m                                                                                                                 |
| m                    | ID length                                                                                                            |
| <IDi h> <IDi l>      | i-th byte of the ID of the tag. ASCII encoded byte                                                                   |
| j                    | 1..4                                                                                                                 |
| <pwdj h><br><pwdj l> | j-th byte of the tag access password. ASCII encoded byte. Use a "0" password if the access password is not requested |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "2" "0" "0" "0" <QTctrl1 h> <QTctrl1 l> <QTctrl2 h> <QTctrl2 l> ETX <bcc> CR

Where:

|                            |                                                                               |
|----------------------------|-------------------------------------------------------------------------------|
| <QTctrl1 h><br><QTctrl1 l> | 1st byte (MSB) of the QT control field read from the tag. ASCII encoded byte. |
| <QTctrl2 h><br><QTctrl2 l> | 2nd byte (LSB) of the QT control field read from the tag. ASCII encoded byte  |

- b) if errors occurred during the transaction:

SOH <add h> <add l> STX "2" "0" "0" "2" ETX <bcc> CR

- c) if no tag is present:

SOH <add h> <add l> STX "2" "0" "0" "1" ETX <bcc> CR

## 2.87 "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 datasheet.

The "master" sends the following command:

SOH <add h> <add l> STX "2" "1" <ID1 h> <ID1 l>...<IDi h> <IDi l>...<IDm h> <IDm l>  
<pwd1 h> <pwd1 l>...<pwdj h> <pwdj l>...<pwd4 h> <pwd4 l> <persistence h>  
<persistence l> <QTctrl1 h> <QTctrl1 l> <QTctrl2 h> <QTctrl2 l> ETX <bcc> CR

Where:

|                                    |                                                                                                                                                                                                                                  |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                                  | 1..m                                                                                                                                                                                                                             |
| m                                  | ID length                                                                                                                                                                                                                        |
| <IDi h> <IDi l>                    | i-th byte of the ID of the tag. ASCII encoded byte                                                                                                                                                                               |
| j                                  | 1..4                                                                                                                                                                                                                             |
| <pwdj h><br><pwdj l>               | j-th byte of the tag access password. ASCII encoded byte. Use a "0" password if the access password is not requested                                                                                                             |
| <persistence h><br><persistence l> | Indicates whether the QT control is written to non-volatile or volatile memory. ASCII encoded byte: <ul style="list-style-type: none"> <li>0x00: Write to volatile memory</li> <li>0x01: Write to non-volatile memory</li> </ul> |
| <QTctrl1 h><br><QTctrl1 l>         | 1st byte (MSB) of the QT control field read from the tag. ASCII encoded byte.                                                                                                                                                    |
| <QTctrl2 h><br><QTctrl2 l>         | 2nd byte (LSB) of the QT control field read from the tag. ASCII encoded byte                                                                                                                                                     |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "2" "1" "0" "0" ETX <bcc> CR

- b) if errors occurred during the transaction:

SOH <add h> <add l> STX “2” “1” “0” “2” ETX <bcc> CR

- c) if no tag is present:

SOH <add h> <add l> STX “2” “1” “0” “1” ETX <bcc> CR

## 2.88 “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 datasheet.

The “master” sends the following command:

SOH <add h> <add l> STX “2” “2” <chip h> <chip l> <ID1 h> <ID1 l>..

Where:

|                      |                                                                                                                                   |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <chip h> <chip l>    | The chip code. ASCII encoded byte: <ul style="list-style-type: none"> <li>• 0x02: Magnus S2</li> <li>• 0x03: Magnus S3</li> </ul> |
| i                    | 1..m                                                                                                                              |
| m                    | ID length                                                                                                                         |
| <IDi h> <IDi l>      | i-th byte of the ID of the tag. ASCII encoded byte                                                                                |
| j                    | 1..4                                                                                                                              |
| <pwdj h><br><pwdj l> | j-th byte of the tag access password. ASCII encoded byte. Use a “0” password if the access password is not requested              |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX “2” “2” “0” “0” <sens hh> <sens hl> <sens lh> <sens ll> ETX <bcc> CR

Where:

|                                            |                                                       |
|--------------------------------------------|-------------------------------------------------------|
| <sens hh> <sens hl><br><sens lh> <sens ll> | The sensor code read from the tag. ASCII encoded word |
|--------------------------------------------|-------------------------------------------------------|

- b) if errors occurred during the transaction:

SOH <add h> <add l> STX “2” “2” “0” “2” ETX <bcc> CR

- c) if no tag is present:

SOH <add h> <add l> STX “2” “2” “0” “1” ETX <bcc> CR

## 2.89 “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 datasheet.

The “master” sends the following command:

SOH <add h> <add l> STX “2” “3” <chip h> <chip l> <ID1 h> <ID1 l>..

Where:

|                          |                                                                                                                                                                                                    |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <chip h> <chip l>        | The chip code. ASCII encoded byte: <ul style="list-style-type: none"> <li>• 0x02: Magnus S2</li> <li>• 0x03: Magnus S3</li> </ul>                                                                  |
| i                        | 1..m                                                                                                                                                                                               |
| m                        | ID length                                                                                                                                                                                          |
| <IDi h> <IDi l>          | i-th byte of the ID of the tag. ASCII encoded byte                                                                                                                                                 |
| j                        | 1..4                                                                                                                                                                                               |
| <pwdj h><br><pwdj l>     | j-th byte of the tag access password. ASCII encoded byte. Use a "0" password if the access password is not requested                                                                               |
| <match h><br><match l>   | The RSSI threshold match criteria. ASCII encoded byte: <ul style="list-style-type: none"> <li>• 0x00: Match if code is &lt;= threshold</li> <li>• 0x01: Match if code is &gt; threshold</li> </ul> |
| <thhold h><br><thhold l> | The RSSI threshold in the range 0..31. ASCII encoded byte                                                                                                                                          |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "2" "3" "0" "0" <rss i h> <rss i l> ETX <bcc> CR

Where:

|                     |                                                        |
|---------------------|--------------------------------------------------------|
| <rss i h> <rss i l> | The on-chip RSSI read from the tag. ASCII encoded byte |
|---------------------|--------------------------------------------------------|

- b) if errors occurred during the transaction:

SOH <add h> <add l> STX "2" "3" "0" "2" ETX <bcc> CR

- c) if no tag is present:

SOH <add h> <add l> STX "2" "3" "0" "1" ETX <bcc> CR

## 2.90 "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 datasheet.

The "master" sends the following command:

SOH <add h> <add l> STX "2" "4" <chip h> <chip l> <ID1 h> <ID1 l>..  
<IDi h> <IDi l>..  
<IDm h> <IDm l> <pwd1 h> <pwd1 l>..  
<pwdj h> <pwdj l>..  
<pwd4 h> <pwd4 l> ETX <bcc> CR

Where:

|                      |                                                                                                                      |
|----------------------|----------------------------------------------------------------------------------------------------------------------|
| <chip h> <chip l>    | The chip code. ASCII encoded byte: <ul style="list-style-type: none"> <li>• 0x03: Magnus S3</li> </ul>               |
| i                    | 1..m                                                                                                                 |
| m                    | ID length                                                                                                            |
| <IDi h> <IDi l>      | i-th byte of the ID of the tag. ASCII encoded byte                                                                   |
| j                    | 1..4                                                                                                                 |
| <pwdj h><br><pwdj l> | j-th byte of the tag access password. ASCII encoded byte. Use a "0" password if the access password is not requested |

If the addressed Bluebox cannot execute the command, it answers:

SOH <add h> <add l> NAK <bcc> CR

Otherwise, it answers with:

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

SOH <add h> <add l> STX "2" "4" "0" "0" <temp hh> <temp hl> <temp lh> <temp ll> ETX <bcc> CR

Where:

|                     |                                                            |
|---------------------|------------------------------------------------------------|
| <temp hh> <temp hl> | The temperature code read from the tag. ASCII encoded word |
| <temp lh> <temp ll> |                                                            |

- b) if errors occurred during the transaction:

SOH <add h> <add l> STX "2" "4" "0" "2" ETX <bcc> CR

- c) if no tag is present:

SOH <add h> <add l> STX "2" "4" "0" "1" ETX <bcc> CR

## 2.91 "Spontaneous" Message

In "continuous" mode, if the "spontaneous" feature is set (see parameters), the Bluebox will send the following message on the serial line and on the Ethernet channel (if available) every time that it will find a "new" tag.

- a) for LF devices with only 1 antenna:

STX <code1 h> <code1 l>..

Where:

|                     |                                                                                                                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                   | 1..n                                                                                                                                                                                                  |
| n                   | Number of bytes of the tag code: <ul style="list-style-type: none"> <li>• 5: UNIQUE, Bluebox Short</li> <li>• 10: Bluebox Medium</li> <li>• 20: Bluebox Large</li> </ul>                              |
| <codei h> <codei l> | i-th byte of the code of the identified tag. ASCII encoded byte                                                                                                                                       |
| <bcc>               | Block check character or checksum calculated as "xor" of the previous characters starting from STX applying the following rule: if <bcc> = STX or <bcc> = CR, then <bcc> := <bcc> +1 (increment of 1) |

- b) for LF devices with 2 antennas:

STX <code1 h> <code1 l>..

Where:

|                     |                                                                                                                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i                   | 1..n                                                                                                                                                                                                  |
| n                   | Number of bytes of the tag code: <ul style="list-style-type: none"> <li>• 5: UNIQUE, Bluebox Short</li> <li>• 10: Bluebox Medium</li> <li>• 20: Bluebox Large</li> </ul>                              |
| <codei h> <codei l> | i-th byte of the code of the identified tag. ASCII encoded byte                                                                                                                                       |
| <ant h> <ant l>     | The antenna number which has identified the tag, ASCII encoded byte: <ul style="list-style-type: none"> <li>• 0x01: Antenna no.1</li> <li>• 0x02: Antenna no. 2</li> </ul>                            |
| <bcc>               | Block check character or checksum calculated as "xor" of the previous characters starting from STX applying the following rule: if <bcc> = STX or <bcc> = CR, then <bcc> := <bcc> +1 (increment of 1) |

- c) for HF devices with only 1 antenna:

STX <type h> <type l> <UID1 h> <UID1 l>..<<UIDi h> <UIDi l>..<<UIDn h> <UIDn l> ETX  
<bcc> CR

Where:

|                   |                                                                                                                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <type h> <type l> | Transponder type                                                                                                                                                                                      |
| i                 | 1..n (the UID length)                                                                                                                                                                                 |
| <UIDi h> <UIDi l> | i-th byte of the UID of the identified tag. ASCII encoded byte                                                                                                                                        |
| <bcc>             | Block check character or checksum calculated as “xor” of the previous characters starting from STX applying the following rule: if <bcc> = STX or <bcc> = CR, then <bcc> := <bcc> +1 (increment of 1) |

d) for HF devices with 2 antennas:

STX <type h> <type l> <UID1 h> <UID1 l>..<<UIDi h> <UIDi l>..<<UIDn h> <UIDn l>  
<ant h> <ant l> ETX <bcc> CR

Where:

|                   |                                                                                                                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <type h> <type l> | Transponder type                                                                                                                                                                                      |
| i                 | 1..n (the UID length)                                                                                                                                                                                 |
| <UIDi h> <UIDi l> | i-th byte of the UID of the identified tag. ASCII encoded byte                                                                                                                                        |
| <ant h> <ant l>   | The antenna number which has identified the tag, ASCII encoded byte: <ul style="list-style-type: none"> <li>0x01: Antenna no.1</li> <li>0x02: Antenna no. 2</li> </ul>                                |
| <bcc>             | Block check character or checksum calculated as “xor” of the previous characters starting from STX applying the following rule: if <bcc> = STX or <bcc> = CR, then <bcc> := <bcc> +1 (increment of 1) |

e) for UHF devices:

STX <tag h> <tag l> <ID1 h> <ID1 l>..<<IDi h> <IDi l>..<<IDn h> <IDn l> <RSSI Q h>  
<RSSI Q l> <RSSI I h> <RSSI I l> <ant h> <ant l> <dir h> <dir l> <tm1 h> <tm1 l>..  
<tm7 h> <tm7 l> ETX <bcc> CR

Where:

|                                                |                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <tag h> <tag l>                                | Transponder type : <ul style="list-style-type: none"> <li>0x02: ISO 18000-63 (EPC Class-1 Generation-2)</li> </ul> (optional parameter present only if the tag type information flag in the general parameters is active, see the reader user manual for more info)                                                                    |
| i                                              | 1..n (the UID length)                                                                                                                                                                                                                                                                                                                  |
| <IDi h> <IDi l>                                | i-th byte of the UID of the identified tag. ASCII encoded byte                                                                                                                                                                                                                                                                         |
| <RSSI Q h> <RSSI Q l><br><RSSI I h> <RSSI I l> | RSSI Q and I channel info in dB of the identified tag. ASCII encoded bytes<br>(optional parameter present only if the RSSI information flag in the RF configuration parameters is active, see the reader user manual for more info)                                                                                                    |
| <ant h> <ant l>                                | Reading antenna. ASCII character: <ul style="list-style-type: none"> <li>0x01: Antenna 1</li> <li>0x02: Antenna 2</li> <li>0x03: Antenna 3</li> <li>0x04: Antenna 4</li> </ul> (optional parameter present only if the reading antenna information flag in the general parameters is active, see the reader user manual for more info) |
| <dir h> <dir l>                                | Gate crossing direction for the identified tag. ASCII character: <ul style="list-style-type: none"> <li>0x01: Crossing from input 1 to input 2</li> <li>0x02: Crossing from input 2 to input 1</li> </ul> (optional parameter present only if “gate” mode is active, see the reader user manual for more info)                         |



|                                          |                                                                                                                                                                                                                                                                                      |
|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <tm1 h> <tm1 l><br>..<br><tm7 h> <tm7 l> | Timestamp for the identified tag. ASCII encoded byte array of the BCD encoded timestamp with the format yyyyMMddhhmmss.<br>(optional parameter present only if the reading timestamp information flag in the general parameters is active, see the reader user manual for more info) |
| <bcc>                                    | Block check character or checksum calculated as “xor” of the previous characters starting from STX applying the following rule: if <bcc> = STX or <bcc> = CR, then <bcc> := <bcc> +1 (increment of 1)                                                                                |

### 3 Examples

Hereinafter, the firmware version reading, the 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.

The “master” sends the following command:

|       |                               |
|-------|-------------------------------|
| ASCII | <SOH>FF<STX>34<ETX><BELL><CR> |
| Hex   | 01 46 46 02 33 34 03 07 0D    |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex) | Value (ASCII) | Description           |
|-------------|----------|---------------|---------------|-----------------------|
| 0           | 1        | 01            | SOH           | Start of Header       |
| 1..2        | 2        | 46 46         | FF            | Node address = 255    |
| 3           | 1        | 02            | STX           | Start of Text         |
| 4..5        | 2        | 33 34         | 34            | Command code = 0x34   |
| 6           | 1        | 03            | ETX           | End of Text           |
| 7           | 1        | 07            | BELL          | Packet checksum (bcc) |
| 8           | 1        | 0D            | CR            | Carriage Return       |

If the addressed Bluebox cannot execute the command, it answers:

|       |                       |
|-------|-----------------------|
| ASCII | <SOH>FF<NAK><DC4><CR> |
| Hex   | 01 46 46 15 14 0D     |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex) | Value (ASCII) | Description           |
|-------------|----------|---------------|---------------|-----------------------|
| 0           | 1        | 01            | SOH           | Start of Header       |
| 1..2        | 2        | 46 46         | FF            | Node address = 255    |
| 3           | 1        | 15            | NAK           | Not Acknowledge       |
| 4           | 1        | 14            | DC4           | Packet checksum (bcc) |
| 5           | 1        | 0D            | CR            | Carriage Return       |

Otherwise, it answers with:

|       |                                                                                                                            |
|-------|----------------------------------------------------------------------------------------------------------------------------|
| ASCII | <SOH>FF<STX>34424C5545424F585F4C4620312E303020<ETX>}<CR>                                                                   |
| Hex   | 01 46 46 02 33 34 34 32 34 43 35 35 34 35 34 32 34 46 35 38 35 46 34 43 34 36 32 30 33 31 32 45 33 30 33 30 32 30 03 7D 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                                                              | Value (ASCII)                                | Description          |
|-------------|----------|----------------------------------------------------------------------------|----------------------------------------------|----------------------|
| 0           | 1        | 01                                                                         | SOH                                          | Start of Header      |
| 1..2        | 2        | 46 46                                                                      | FF                                           | Node address = 255   |
| 3           | 1        | 02                                                                         | STX                                          | Start of Text        |
| 4..5        | 2        | 33 34                                                                      | 34                                           | Command code = 0x34  |
| 6..38       | 32       | 3432 3443<br>3535 3435<br>3432 3446<br>3538 3546<br>3443 3436<br>3230 3331 | 424C55454<br>24F585F4C<br>4620312E3<br>03020 | FW = BLUEBOX_LF 1.00 |

| Byte Number | Tx Bytes | Tx Data (Hex)          | Value (ASCII) | Description           |
|-------------|----------|------------------------|---------------|-----------------------|
|             |          | 3245 3330<br>3330 3230 |               |                       |
| 39          | 1        | 03                     | ETX           | End of Text           |
| 40          | 1        | 7D                     | }             | Packet checksum (bcc) |
| 41          | 1        | 0D                     | CR            | Carriage Return       |

### 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 held 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.

The “master” sends the following command:

|       |                       |
|-------|-----------------------|
| ASCII | <SOH>FF<ENQ><ENQ><CR> |
| Hex   | 01 46 46 05 05 0D     |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex) | Value (ASCII) | Description           |
|-------------|----------|---------------|---------------|-----------------------|
| 0           | 1        | 01            | SOH           | Start of Header       |
| 1..2        | 2        | 46 46         | FF            | Node address = 255    |
| 3           | 1        | 05            | ENQ           | Enquiry               |
| 4           | 1        | 05            | ENQ           | Packet checksum (bcc) |
| 5           | 1        | 0D            | CR            | Carriage Return       |

If the addressed Bluebox cannot execute the command, it answers:

|       |                       |
|-------|-----------------------|
| ASCII | <SOH>FF<NAK><DC4><CR> |
| Hex   | 01 46 46 15 14 0D     |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex) | Value (ASCII) | Description           |
|-------------|----------|---------------|---------------|-----------------------|
| 0           | 1        | 01            | SOH           | Start of Header       |
| 1..2        | 2        | 46 46         | FF            | Node address = 255    |
| 3           | 1        | 15            | NAK           | Not Acknowledge       |
| 4           | 1        | 14            | DC4           | Packet checksum (bcc) |
| 5           | 1        | 0D            | CR            | Carriage Return       |

Otherwise, it answers with:

- a) for LF devices with only 1 antenna and a transponder has been identified

|       |                                                    |
|-------|----------------------------------------------------|
| ASCII | <SOH>FF<STX>0011223344<ETX><NULL><CR>              |
| Hex   | 01 46 46 02 30 30 31 31 32 32 33 33 34 34 03 00 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description               |
|-------------|----------|--------------------------------|---------------|---------------------------|
| 0           | 1        | 01                             | SOH           | Start of Header           |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255        |
| 3           | 1        | 02                             | STX           | Start of Text             |
| 4..14       | 10       | 3030 3131<br>3232 3333<br>3434 | 0011223344    | Tag code = 00 11 22 33 44 |
| 15          | 1        | 03                             | ETX           | End of Text               |

| Byte Number | Tx Bytes | Tx Data (Hex) | Value (ASCII) | Description           |
|-------------|----------|---------------|---------------|-----------------------|
| 16          | 1        | 00            | NULL          | Packet checksum (bcc) |
| 17          | 1        | 0D            | CR            | Carriage Return       |

b) for LF devices with only 1 antenna but no transponder has been identified

|       |                                                 |
|-------|-------------------------------------------------|
| ASCII | <SOH>FF<STX>0000000000<ETX><NULL><CR>           |
| Hex   | 01 46 46 02 30 30 30 30 30 30 30 30 30 03 00 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description           |
|-------------|----------|--------------------------------|---------------|-----------------------|
| 0           | 1        | 01                             | SOH           | Start of Header       |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255    |
| 3           | 1        | 02                             | STX           | Start of Text         |
| 4..14       | 10       | 3030 3030<br>3030 3030<br>3030 | 0000000000    | Null tag              |
| 15          | 1        | 03                             | ETX           | End of Text           |
| 16          | 1        | 00                             | NULL          | Packet checksum (bcc) |
| 17          | 1        | 0D                             | CR            | Carriage Return       |

c) for LF devices with 2 antennas and both antennas have identified a transponder

|       |                                                                                     |
|-------|-------------------------------------------------------------------------------------|
| ASCII | <SOH>FF<STX>0011223344-0102030405<ETX><NULL><CR>                                    |
| Hex   | 01 46 46 02 30 30 31 31 32 32 33 33 34 34 2D 30 31 30 32 30 33 30 34 30 35 03 02 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description               |
|-------------|----------|--------------------------------|---------------|---------------------------|
| 0           | 1        | 01                             | SOH           | Start of Header           |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255        |
| 3           | 1        | 02                             | STX           | Start of Text             |
| 4..14       | 10       | 3030 3131<br>3232 3333<br>3434 | 0011223344    | Tag code = 00 11 22 33 44 |
| 15          | 1        | 2D                             | -             | Separator antenna 1 / 2   |
| 16..26      | 10       | 3031 3032<br>3033 3034<br>3035 | 0102030405    | Tag code = 01 02 03 04 05 |
| 27          | 1        | 03                             | ETX           | End of Text               |
| 28          | 1        | 02                             | STX           | Packet checksum (bcc)     |
| 29          | 1        | 0D                             | CR            | Carriage Return           |

d) for LF devices with 2 antennas and only antenna 1 identified a transponder

|       |                                                                                  |
|-------|----------------------------------------------------------------------------------|
| ASCII | <SOH>FF<STX>0011223344-0000000000<ETX><NULL><CR>                                 |
| Hex   | 01 46 46 02 30 30 31 31 32 32 33 33 34 34 2D 30 30 30 30 30 30 30 30 30 03 2D 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex) | Value (ASCII) | Description        |
|-------------|----------|---------------|---------------|--------------------|
| 0           | 1        | 01            | SOH           | Start of Header    |
| 1..2        | 2        | 46 46         | FF            | Node address = 255 |
| 3           | 1        | 02            | STX           | Start of Text      |

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description               |
|-------------|----------|--------------------------------|---------------|---------------------------|
| 4..14       | 10       | 3030 3131<br>3232 3333<br>3434 | 0011223344    | Tag code = 00 11 22 33 44 |
| 15          | 1        | 2D                             | -             | Separator antenna 1 / 2   |
| 16..26      | 10       | 3030 3030<br>3030 3030<br>3030 | 0000000000    | Null tag                  |
| 27          | 1        | 03                             | ETX           | End of Text               |
| 28          | 1        | 05                             | ENQ           | Packet checksum (bcc)     |
| 29          | 1        | 0D                             | CR            | Carriage Return           |

e) for LF devices with 2 antennas and only antenna 2 identified a transponder

|       |                                                                                     |
|-------|-------------------------------------------------------------------------------------|
| ASCII | <SOH>FF<STX>0000000000-0102030405<ETX><NULL><CR>                                    |
| Hex   | 01 46 46 02 30 30 30 30 30 30 30 30 30 30 2D 30 31 30 32 30 33 30 34 30 35 03 02 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description               |
|-------------|----------|--------------------------------|---------------|---------------------------|
| 0           | 1        | 01                             | SOH           | Start of Header           |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255        |
| 3           | 1        | 02                             | STX           | Start of Text             |
| 4..14       | 10       | 3030 3030<br>3030 3030<br>3030 | 0000000000    | Null tag                  |
| 15          | 1        | 2D                             | -             | Separator antenna 1 / 2   |
| 16..26      | 10       | 3031 3032<br>3033 3034<br>3035 | 0102030405    | Tag code = 01 02 03 04 05 |
| 27          | 1        | 03                             | ETX           | End of Text               |
| 28          | 1        | 07                             | BELL          | Packet checksum (bcc)     |
| 29          | 1        | 0D                             | CR            | Carriage Return           |

f) for LF devices with 2 antennas but no antenna has identified a transponder

|       |                                                                                  |
|-------|----------------------------------------------------------------------------------|
| ASCII | <SOH>FF<STX>0000000000-0000000000<ETX><NULL><CR>                                 |
| Hex   | 01 46 46 02 30 30 30 30 30 30 30 30 30 30 2D 30 30 30 30 30 30 30 30 30 03 2D 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description             |
|-------------|----------|--------------------------------|---------------|-------------------------|
| 0           | 1        | 01                             | SOH           | Start of Header         |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255      |
| 3           | 1        | 02                             | STX           | Start of Text           |
| 4..14       | 10       | 3030 3030<br>3030 3030<br>3030 | 0000000000    | Null tag                |
| 15          | 1        | 2D                             | -             | Separator antenna 1 / 2 |
| 16..26      | 10       | 3030 3030<br>3030 3030<br>3030 | 0000000000    | Null tag                |
| 27          | 1        | 03                             | ETX           | End of Text             |

| Byte Number | Tx Bytes | Tx Data (Hex) | Value (ASCII) | Description           |
|-------------|----------|---------------|---------------|-----------------------|
| 28          | 1        | 2D            | -             | Packet checksum (bcc) |
| 29          | 1        | 0D            | CR            | Carriage Return       |

g) for HF devices with only 1 antenna and a transponder has been identified

|       |                                                    |
|-------|----------------------------------------------------|
| ASCII | <SOH>FF<STX>1111223344<ETX><NULL><CR>              |
| Hex   | 01 46 46 02 31 31 31 31 32 32 33 33 34 34 03 00 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)          | Value (ASCII) | Description                  |
|-------------|----------|------------------------|---------------|------------------------------|
| 0           | 1        | 01                     | SOH           | Start of Header              |
| 1..2        | 2        | 46 46                  | FF            | Node address = 255           |
| 3           | 1        | 02                     | STX           | Start of Text                |
| 4..5        | 2        | 31 31                  | 11            | Tag type = MIFARE 1k (UID=4) |
| 6..14       | 8        | 3131 3232<br>3333 3434 | 11223344      | Tag code = 11 22 33 44       |
| 15          | 1        | 03                     | ETX           | End of Text                  |
| 16          | 1        | 00                     | NULL          | Packet checksum (bcc)        |
| 17          | 1        | 0D                     | CR            | Carriage Return              |

h) for HF devices with only 1 antenna but no transponder had been identified

|       |                                                    |
|-------|----------------------------------------------------|
| ASCII | <SOH>FF<STX> 0000000000<ETX><NULL><CR>             |
| Hex   | 01 46 46 02 30 30 30 30 30 30 30 30 30 30 03 00 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description           |
|-------------|----------|--------------------------------|---------------|-----------------------|
| 0           | 1        | 01                             | SOH           | Start of Header       |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255    |
| 3           | 1        | 02                             | STX           | Start of Text         |
| 4..14       | 10       | 3030 3030<br>3030 3030<br>3030 | 0000000000    | Null tag              |
| 15          | 1        | 03                             | ETX           | End of Text           |
| 16          | 1        | 00                             | NULL          | Packet checksum (bcc) |
| 17          | 1        | 0D                             | CR            | Carriage Return       |

i) for HF devices with 2 antennas and both antennas identified a transponder

|       |                                                                                     |
|-------|-------------------------------------------------------------------------------------|
| ASCII | <SOH>FF<STX>1111223344-1201020304<ETX><CR>                                          |
| Hex   | 01 46 46 02 31 31 31 31 32 32 33 33 34 34 2D 31 32 30 31 30 32 30 33 30 34 03 29 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)          | Value (ASCII) | Description                  |
|-------------|----------|------------------------|---------------|------------------------------|
| 0           | 1        | 01                     | SOH           | Start of Header              |
| 1..2        | 2        | 46 46                  | FF            | Node address = 255           |
| 3           | 1        | 02                     | STX           | Start of Text                |
| 4..5        | 2        | 31 31                  | 11            | Tag type = MIFARE 1k (UID=4) |
| 6..14       | 8        | 3131 3232<br>3333 3434 | 11223344      | Tag code = 11 22 33 44       |
| 15          | 1        | 2D                     | -             | Separator antenna 1 / 2      |
| 16..17      | 2        | 31 32                  | 12            | Tag type = MIFARE 4k (UID=4) |

| Byte Number | Tx Bytes | Tx Data (Hex)          | Value (ASCII) | Description            |
|-------------|----------|------------------------|---------------|------------------------|
| 18..26      | 8        | 3031 3032<br>3033 3034 | 01020304      | Tag code = 01 02 03 04 |
| 27          | 1        | 03                     | ETX           | End of Text            |
| 28          | 1        | 29                     | )             | Packet checksum (bcc)  |
| 29          | 1        | 0D                     | CR            | Carriage Return        |

j) for HF devices with 2 antennas and only antenna 1 identified a transponder

|       |                                                                                  |
|-------|----------------------------------------------------------------------------------|
| ASCII | <SOH>FF<STX>1111223344-0000000000<ETX><ENQ><CR>                                  |
| Hex   | 01 46 46 02 31 31 31 31 32 32 33 33 34 34 2D 30 30 30 30 30 30 30 30 03 2D<br>0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description                  |
|-------------|----------|--------------------------------|---------------|------------------------------|
| 0           | 1        | 01                             | SOH           | Start of Header              |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255           |
| 3           | 1        | 02                             | STX           | Start of Text                |
| 4..5        | 2        | 31 31                          | 11            | Tag type = MIFARE 1k (UID=4) |
| 6..14       | 8        | 3131 3232<br>3333 3434         | 11223344      | Tag code = 11 22 33 44       |
| 15          | 1        | 2D                             | -             | Separator antenna 1 / 2      |
| 16..26      | 10       | 3030 3030<br>3030 3030<br>3030 | 0000000000    | Null tag                     |
| 27          | 1        | 03                             | ETX           | End of Text                  |
| 28          | 1        | 05                             | ENQ           | Packet checksum (bcc)        |
| 29          | 1        | 0D                             | CR            | Carriage Return              |

k) for HF devices with 2 antennas and only antenna 2 identified a transponder

|       |                                                                                     |
|-------|-------------------------------------------------------------------------------------|
| ASCII | <SOH>FF<STX>0000000000-1201020304<ETX><BELL><CR>                                    |
| Hex   | 01 46 46 02 30 30 30 30 30 30 30 30 30 2D 31 32 30 31 30 32 30 33 30 34 03 07<br>0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description                  |
|-------------|----------|--------------------------------|---------------|------------------------------|
| 0           | 1        | 01                             | SOH           | Start of Header              |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255           |
| 3           | 1        | 02                             | STX           | Start of Text                |
| 4..14       | 10       | 3030 3030<br>3030 3030<br>3030 | 0000000000    | Null tag                     |
| 15          | 1        | 2D                             | -             | Separator antenna 1 / 2      |
| 16..17      | 2        | 31 32                          | 12            | Tag type = MIFARE 4k (UID=4) |
| 18..26      | 8        | 3031 3032<br>3033 3034         | 01020304      | Tag code = 01 02 03 04       |
| 27          | 1        | 03                             | ETX           | End of Text                  |
| 28          | 1        | 07                             | BELL          | Packet checksum (bcc)        |
| 29          | 1        | 0D                             | CR            | Carriage Return              |

l) for HF devices with 2 antennas but no transponder has been identified

|       |                                             |
|-------|---------------------------------------------|
| ASCII | <SOH>FF<STX>0000000000-0000000000<ETX>-<CR> |
|-------|---------------------------------------------|

|     |                                                                                  |
|-----|----------------------------------------------------------------------------------|
| Hex | 01 46 46 02 30 30 30 30 30 30 30 30 30 30 2D 30 30 30 30 30 30 30 30 30 03 2D 0D |
|-----|----------------------------------------------------------------------------------|

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description             |
|-------------|----------|--------------------------------|---------------|-------------------------|
| 0           | 1        | 01                             | SOH           | Start of Header         |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255      |
| 3           | 1        | 02                             | STX           | Start of Text           |
| 4..14       | 10       | 3030 3030<br>3030 3030<br>3030 | 0000000000    | Null tag                |
| 15          | 1        | 2D                             | -             | Separator antenna 1 / 2 |
| 16..26      | 10       | 3030 3030<br>3030 3030<br>3030 | 0000000000    | Null tag                |
| 27          | 1        | 03                             | ETX           | End of Text             |
| 28          | 1        | 2D                             | -             | Packet checksum (bcc)   |
| 29          | 1        | 0D                             | CR            | Carriage Return         |

- m) for UHF devices with 1 antenna and only one transponder was found (transponder info, antenna info, RSSI info and tag read count are not active and the device is not in “gate” mode

|       |                                                                                                                      |
|-------|----------------------------------------------------------------------------------------------------------------------|
| ASCII | <SOH>FF<STX>3000000102030405060708090A0BFF27<ETX><ENQ><CR>                                                           |
| Hex   | 01 46 46 02 33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37 03 05 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                                                                                        | Value (ASCII)                                | Description                                                                                                |
|-------------|----------|------------------------------------------------------------------------------------------------------|----------------------------------------------|------------------------------------------------------------------------------------------------------------|
| 0           | 1        | 01                                                                                                   | SOH                                          | Start of Header                                                                                            |
| 1..2        | 2        | 46 46                                                                                                | FF                                           | Node address = 255                                                                                         |
| 3           | 1        | 02                                                                                                   | STX                                          | Start of Text                                                                                              |
| 4..36       | 32       | 3330 3030<br>3030 3031<br>3032 3033<br>3034 3035<br>3036 3037<br>3038 3039<br>3041 3042<br>4646 3237 | 30000001<br>02030405<br>06070809<br>0A0BFF27 | Tag code = 33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37 |
| 37          | 1        | 03                                                                                                   | ETX                                          | End of Text                                                                                                |
| 38          | 1        | 05                                                                                                   | ENQ                                          | Packet checksum (bcc)                                                                                      |
| 39          | 1        | 0D                                                                                                   | CR                                           | Carriage Return                                                                                            |

### 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.

The “master” sends the following command:

|       |                       |
|-------|-----------------------|
| ASCII | <SOH>FF<SYN><ETB><CR> |
| Hex   | 01 46 46 16 17 0D     |

And in detail:



| Byte Number | Tx Bytes | Tx Data (Hex) | Value (ASCII) | Description           |
|-------------|----------|---------------|---------------|-----------------------|
| 0           | 1        | 01            | SOH           | Start of Header       |
| 1..2        | 2        | 46 46         | FF            | Node address = 255    |
| 3           | 1        | 16            | SYN           | Synchronous idle      |
| 4           | 1        | 17            | ETB           | Packet checksum (bcc) |
| 5           | 1        | 0D            | CR            | Carriage Return       |

If the addressed Bluebox cannot execute the command, it answers:

|       |                       |
|-------|-----------------------|
| ASCII | <SOH>FF<NAK><DC4><CR> |
| Hex   | 01 46 46 15 14 0D     |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex) | Value (ASCII) | Description           |
|-------------|----------|---------------|---------------|-----------------------|
| 0           | 1        | 01            | SOH           | Start of Header       |
| 1..2        | 2        | 46 46         | FF            | Node address = 255    |
| 3           | 1        | 15            | NAK           | Not Acknowledge       |
| 4           | 1        | 14            | DC4           | Packet checksum (bcc) |
| 5           | 1        | 0D            | CR            | Carriage Return       |

Otherwise, it answers with:

a) for LF devices with only 1 antenna

|       |                                                    |
|-------|----------------------------------------------------|
| ASCII | <SOH>FF<STX>0011223344<ETX><NULL><CR>              |
| Hex   | 01 46 46 02 30 30 31 31 32 32 33 33 34 34 03 00 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description               |
|-------------|----------|--------------------------------|---------------|---------------------------|
| 0           | 1        | 01                             | SOH           | Start of Header           |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255        |
| 3           | 1        | 02                             | STX           | Start of Text             |
| 4..14       | 10       | 3030 3131<br>3232 3333<br>3434 | 0011223344    | Tag code = 00 11 22 33 44 |
| 15          | 1        | 03                             | ETX           | End of Text               |
| 16          | 1        | 00                             | NULL          | Packet checksum (bcc)     |
| 17          | 1        | 0D                             | CR            | Carriage Return           |

b) for LF devices with 2 antennas

|       |                                                          |
|-------|----------------------------------------------------------|
| ASCII | <SOH>FF<STX>001122334401<ETX><STX><CR>                   |
| Hex   | 01 46 46 02 30 30 31 31 32 32 33 33 34 34 30 31 03 02 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description               |
|-------------|----------|--------------------------------|---------------|---------------------------|
| 0           | 1        | 01                             | SOH           | Start of Header           |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255        |
| 3           | 1        | 02                             | STX           | Start of Text             |
| 4..14       | 10       | 3030 3131<br>3232 3333<br>3434 | 0011223344    | Tag code = 00 11 22 33 44 |
| 15..16      | 2        | 30 31                          | 01            | Antenna = 1               |
| 17          | 1        | 03                             | ETX           | End of Text               |
| 18          | 1        | 02                             | STX           | Packet checksum (bcc)     |
| 19          | 1        | 0D                             | CR            | Carriage Return           |

c) for HF devices with only 1 antenna

|       |                                                    |
|-------|----------------------------------------------------|
| ASCII | <SOH>FF<STX>1111223344<ETX><NULL><CR>              |
| Hex   | 01 46 46 02 31 31 31 31 32 32 33 33 34 34 03 00 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)          | Value (ASCII) | Description                  |
|-------------|----------|------------------------|---------------|------------------------------|
| 0           | 1        | 01                     | SOH           | Start of Header              |
| 1..2        | 2        | 46 46                  | FF            | Node address = 255           |
| 3           | 1        | 02                     | STX           | Start of Text                |
| 4..5        | 2        | 31 31                  | 11            | Tag type = MIFARE 1k (UID=4) |
| 6..14       | 8        | 3131 3232<br>3333 3434 | 11223344      | Tag code = 11 22 33 44       |
| 15          | 1        | 03                     | ETX           | End of Text                  |
| 16          | 1        | 00                     | NULL          | Packet checksum (bcc)        |
| 17          | 1        | 0D                     | CR            | Carriage Return              |

d) for HF devices with 2 antennas

|       |                                                          |
|-------|----------------------------------------------------------|
| ASCII | <SOH>FF<STX>111122334401<ETX><STX><CR>                   |
| Hex   | 01 46 46 02 31 31 31 31 32 32 33 33 34 34 30 31 03 02 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)          | Value (ASCII) | Description                  |
|-------------|----------|------------------------|---------------|------------------------------|
| 0           | 1        | 01                     | SOH           | Start of Header              |
| 1..2        | 2        | 46 46                  | FF            | Node address = 255           |
| 3           | 1        | 02                     | STX           | Start of Text                |
| 4..5        | 2        | 31 31                  | 11            | Tag type = MIFARE 1k (UID=4) |
| 6..14       | 8        | 3131 3232<br>3333 3434 | 11223344      | Tag code = 11 22 33 44       |
| 15..16      | 2        | 30 31                  | 01            | Antenna = 1                  |
| 17          | 1        | 03                     | ETX           | End of Text                  |
| 18          | 1        | 02                     | STX           | Packet checksum (bcc)        |
| 19          | 1        | 0D                     | CR            | Carriage Return              |

e) for UHF devices with 1 antenna (transponder info, antenna info and RSSI info are not active, and the device is not in "gate" mode)

|       |                                                                                                                      |
|-------|----------------------------------------------------------------------------------------------------------------------|
| ASCII | <SOH>FF<STX>3000000102030405060708090A0BFF27<ETX><ENQ><CR>                                                           |
| Hex   | 01 46 46 02 33 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30<br>39 30 41 30 42 46 46 32 37 03 05 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                                                              | Value (ASCII)                                | Description                                                                                                   |
|-------------|----------|----------------------------------------------------------------------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| 0           | 1        | 01                                                                         | SOH                                          | Start of Header                                                                                               |
| 1..2        | 2        | 46 46                                                                      | FF                                           | Node address = 255                                                                                            |
| 3           | 1        | 02                                                                         | STX                                          | Start of Text                                                                                                 |
| 4..36       | 32       | 3330 3030<br>3030 3031<br>3032 3033<br>3034 3035<br>3036 3037<br>3038 3039 | 30000001<br>02030405<br>06070809<br>0A0BFF27 | Tag code = 33 30 30 30 30 30 30 31 30 32 30 33 30 34<br>30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37 |

| Byte Number | Tx Bytes | Tx Data (Hex)          | Value (ASCII) | Description           |
|-------------|----------|------------------------|---------------|-----------------------|
|             |          | 3041 3042<br>4646 3237 |               |                       |
| 37          | 1        | 03                     | ETX           | End of Text           |
| 38          | 1        | 05                     | ENQ           | Packet checksum (bcc) |
| 39          | 1        | 0D                     | CR            | Carriage Return       |

If the queue is empty, the Bluebox will answer with:

|       |                                                 |
|-------|-------------------------------------------------|
| ASCII | <SOH>FF<STX>0000000000<ETX><NULL><CR>           |
| Hex   | 01 46 46 02 30 30 30 30 30 30 30 30 30 03 00 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description           |
|-------------|----------|--------------------------------|---------------|-----------------------|
| 0           | 1        | 01                             | SOH           | Start of Header       |
| 1..2        | 2        | 46 46                          | FF            | Node address = 255    |
| 3           | 1        | 02                             | STX           | Start of Text         |
| 4..14       | 10       | 3030 3030<br>3030 3030<br>3030 | 0000000000    | Null tag              |
| 15          | 1        | 03                             | ETX           | End of Text           |
| 16          | 1        | 00                             | NULL          | Packet checksum (bcc) |
| 17          | 1        | 0D                             | CR            | Carriage Return       |

To delete the received code from the queue, the “master” replies to the Bluebox with:

|       |                        |
|-------|------------------------|
| ASCII | <SOH>FF<ACK><BELL><CR> |
| Hex   | 01 46 46 06 07 0D      |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex) | Value (ASCII) | Description           |
|-------------|----------|---------------|---------------|-----------------------|
| 0           | 1        | 01            | SOH           | Start of Header       |
| 1..2        | 2        | 46 46         | FF            | Node address = 255    |
| 3           | 1        | 06            | ACK           | Acknowledge           |
| 4           | 1        | 07            | BELL          | Packet checksum (bcc) |
| 5           | 1        | 0D            | CR            | Carriage Return       |

### 3.4 “Spontaneous” Message

In “continuous” mode, if the “spontaneous” feature is set (see parameters), the Bluebox will send the following message on the serial line and on the Ethernet channel (if available) every time it finds a “new” tag.

a) for LF devices with only 1 antenna

|       |                                           |
|-------|-------------------------------------------|
| ASCII | <STX>0011223344<ETX><SOH><CR>             |
| Hex   | 02 30 30 31 31 32 32 33 33 34 34 03 01 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description               |
|-------------|----------|--------------------------------|---------------|---------------------------|
| 0           | 1        | 06                             | STX           | Start of Text             |
| 1..11       | 10       | 3030 3131<br>3232 3333<br>3434 | 0011223344    | Tag code = 00 11 22 33 44 |
| 12          | 1        | 03                             | ETX           | End of Text               |
| 13          | 1        | 01                             | SOH           | Packet checksum (bcc)     |
| 14          | 1        | 0D                             | CR            | Carriage Return           |

b) for LF devices with 2 antennas

|       |                                                 |
|-------|-------------------------------------------------|
| ASCII | <STX>001122334401<ETX><NULL><CR>                |
| Hex   | 02 30 30 31 31 32 32 33 33 34 34 30 31 03 00 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)                  | Value (ASCII) | Description               |
|-------------|----------|--------------------------------|---------------|---------------------------|
| 0           | 1        | 06                             | STX           | Start of Text             |
| 1..11       | 10       | 3030 3131<br>3232 3333<br>3434 | 0011223344    | Tag code = 00 11 22 33 44 |
| 12..13      | 2        | 30 31                          | 01            | Antenna = 1               |
| 14          | 1        | 03                             | ETX           | End of Text               |
| 15          | 1        | 00                             | NULL          | Packet checksum (bcc)     |
| 16          | 1        | 0D                             | CR            | Carriage Return           |

c) for HF devices with only 1 antenna

|       |                                           |
|-------|-------------------------------------------|
| ASCII | <STX>1111223344<ETX><SOH><CR>             |
| Hex   | 02 31 31 31 31 32 32 33 33 34 34 03 01 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)          | Value (ASCII) | Description                  |
|-------------|----------|------------------------|---------------|------------------------------|
| 0           | 1        | 06                     | STX           | Start of Text                |
| 1..2        | 2        | 31 31                  | 11            | Tag type = MIFARE 1k (UID=4) |
| 3..11       | 8        | 3131 3232<br>3333 3434 | 11223344      | Tag code = 11 22 33 44       |
| 12          | 1        | 03                     | ETX           | End of Text                  |
| 13          | 1        | 01                     | SOH           | Packet checksum (bcc)        |
| 14          | 1        | 0D                     | CR            | Carriage Return              |

d) for HF devices with 2 antennas

|       |                                                 |
|-------|-------------------------------------------------|
| ASCII | <STX>111122334401<ETX><NULL><CR>                |
| Hex   | 02 31 31 31 31 32 32 33 33 34 34 30 31 03 00 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex)          | Value (ASCII) | Description                  |
|-------------|----------|------------------------|---------------|------------------------------|
| 0           | 1        | 06                     | STX           | Start of Text                |
| 1..2        | 2        | 31 31                  | 11            | Tag type = MIFARE 1k (UID=4) |
| 3..11       | 8        | 3131 3232<br>3333 3434 | 11223344      | Tag code = 11 22 33 44       |
| 12..13      | 2        | 30 31                  | 01            | Antenna = 1                  |
| 14          | 1        | 03                     | ETX           | End of Text                  |
| 15          | 1        | 00                     | NULL          | Packet checksum (bcc)        |
| 16          | 1        | 0D                     | CR            | Carriage Return              |

e) for UHF devices (transponder info, antenna info and RSSI info are not active, and the device is not in "gate" mode)

|       |                                                                                                                |
|-------|----------------------------------------------------------------------------------------------------------------|
| ASCII | <STX>3000000102030405060708090A0BFF27<ETX><ENQ><CR>                                                            |
| Hex   | 02 33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41<br>30 42 46 46 32 37 03 05 0D |

And in detail:

| Byte Number | Tx Bytes | Tx Data (Hex) | Value (ASCII) | Description   |
|-------------|----------|---------------|---------------|---------------|
| 0           | 1        | 02            | STX           | Start of Text |

| Byte Number | Tx Bytes | Tx Data (Hex)                                                                                        | Value (ASCII)                                | Description                                                                                                   |
|-------------|----------|------------------------------------------------------------------------------------------------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| 1..33       | 32       | 3330 3030<br>3030 3031<br>3032 3033<br>3034 3035<br>3036 3037<br>3038 3039<br>3041 3042<br>4646 3237 | 30000001<br>02030405<br>06070809<br>0A0BFF27 | Tag code = 33 30 30 30 30 30 30 31 30 32 30 33 30 34<br>30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37 |
| 34          | 1        | 03                                                                                                   | ETX                                          | End of Text                                                                                                   |
| 35          | 1        | 05                                                                                                   | ENQ                                          | Packet checksum (bcc)                                                                                         |
| 36          | 1        | 0D                                                                                                   | CR                                           | Carriage Return                                                                                               |

## 4 Getting Started with C

### 4.1 Command / Reply Checksum

The code below shows how to calculate the command and reply checksum.

```
1. #define SOH 0x01
2. #define EOT 0x04
3. #define CR 0x0D
4.
5. unsigned char checksum_bb_cmd(unsigned char* ptBuffer, unsigned long nLength) {
6. unsigned char bChecksum = 0;
7. unsigned long nCounter;
8.
9. for (nCounter = 0; nCounter < nLength; ++nCounter)
10. bChecksum ^= ptBuffer[nCounter];
11.
12. if ((bChecksum == SOH || (bChecksum == EOT || (bChecksum == CR)))
13. ++bChecksum;
14.
15. return bChecksum;
16. }
```

### 4.2 “Spontaneous” Message Checksum

The code below shows how to calculate the “spontaneous” message checksum.

```
1. #define STX 0x02
2. #define CR 0x0D
3.
4. unsigned char checksum_bb_spt(unsigned char* ptBuffer, unsigned long nLength) {
5. unsigned char bChecksum = 0;
6. unsigned long nCounter;
7.
8. for (nCounter = 0; nCounter < nLength; ++nCounter)
9. bChecksum ^= ptBuffer[nCounter];
10.
11. if ((bChecksum == STX) || (bChecksum == CR))
12. ++bChecksum;
13.
14. return bChecksum;
15. }
```

### 4.3 Command / Reply Management

The code below shows how to build a packet to send to a Bluebox and decode the reply received.  
Hereinafter some conversion functions.

```
1. unsigned char bin_to_hex(unsigned char bValue) {
2. bValue &= 0x0F;
3. if (bValue < 10)
4. return ('0' + bValue);
5. return (bValue - 10 + 'A');
```

```

6. }
7.
8. void byte_to_dchar(unsigned char* pdc, unsigned char bValue) {
9. pdc[0] = bin_to_hex((bValue & 0xF0) >> 4);
10. pdc[1] = bin_to_hex(bValue & 0x0F);
11. }
12.
13. void multi_b2dc(unsigned char* pdc, unsigned char* ptb, unsigned long nLength) {
14. unsigned long nCount;
15.
16. for (nCount = 0; nCount < nLength; ++nCount, pdc += 2)
17. byte_to_dchar(pdc, ptb[nCount]);
18. }
19.
20. unsigned char hex_to_bin(unsigned char bValue) {
21. if ((bValue >= '0') && (bValue <= '9'))
22. return (bValue - '0');
23.
24. if ((bValue >= 'A') && (bValue <= 'F'))
25. return (bValue - 'A' + 10);
26.
27. if ((bValue >= 'a') && (bValue <= 'f'))
28. return (bValue - 'a' + 10);
29.
30. return (0);
31. }
32.
33. unsigned char dchar_to_byte(unsigned char* pdc) {
34. return ((hex_to_bin(pdc[0]) << 4) | hex_to_bin(pdc[1]));
35. }
36.
37. void multi_dc2b(unsigned char*ptb, unsigned char *pdc, unsigned long nLength) {
38. unsigned long nCount;
39.
40. for (nCount = 0; nCount < nLength; ++nCount, pdc += 2)
41. ptb[nCount] = dchar_to_byte(pdc);
42. }

```

Hereinafter the code to build the command packet.

```

1. #define SOH 0x01
2. #define STX 0x02
3. #define EOT 0x04
4. #define CR 0x0D
5.
6. typedef enum {
7. BLUEBOX_CMD_GET_FIRMWARE = 0x00,
8. BLUEBOX_CMD_EMPTY_COMMAND
9. } BLUEBOX_COMMANDS_E;
10.
11. typedef struct {

```

```

12. char* szCommand;
13. char* szDescription;
14. } BLUEBOX_CMD_INFO_T, *BLUEBOX_PCMD_INFO_T;
15.
16. const BLUEBOX_CMD_INFO_T tCommands[] = {
17. {"34", "Get firmware"},
18. {"00", "Empty command (ENQ, SYN, ACK)"}
19. };
20.
21. unsigned char g_bMessage[1024];
22.
23. void pack_command(BLUEBOX_COMMANDS_E eCommand, unsigned char* pSendData,
24. unsigned long nSendData, unsigned char bRawCommand) {
25. unsigned long nSend = 0;
26. unsigned char* pSend = &g_bMessage[0];
27.
28. if (pReplyData != NULL) {
29. // Pack the command to send.
30. if (bRawCommand != 0) {
31. // Only 1 char to send in raw mode
32. nSend = (unsigned long) sprintf((char*) pSend, "%c%02X%c", SOH, 0xFF, bRawCommand);
33. } else {
34. nSend = (unsigned long) sprintf((char*) pSend, "%c%02X%c%s", SOH, 0xFF, STX,
35. tCommands[eCommand].szCommand);
36. multi_b2dc(pSend + nSend, pSendData, nSendData);
37. nSend += nSendData * 2;
38. nSend += (unsigned long) sprintf((char*) (pSend + nSend), "%c", ETX);
39. }
40.
41. // Checksum and carriage return
42. pSend[nSend++] = checksum_bb_cmd(pSend, nSend);
43. pSend[nSend++] = CR;
44.
45. // add code to send data here
46. }
47. }

```

For example to send the “FW Version Reading” command

```
pack_command(BLUEBOX_CMD_GET_FIRMWARE, NULL, 0, 0x00);
```

and to send a “Data Request” command

```
pack_command(BLUEBOX_CMD_EMPTY_COMMAND, NULL, 0, 0x05);
```

Hereinafter the code to decode the reply received.

```

1. #define SOH 0x01
2. #define STX 0x02
3. #define EOT 0x04
4. #define ACK 0x06
5. #define CR 0x0D
6. #define NAK 0x15

```



```

7. #define __TRUE 1
8. #define __FALSE 0
9.
10. bool check_command(unsigned char bAddress, unsigned char* pData, unsigned long nData) {
11. bool fRetVal = __FALSE;
12.
13. if ((pData != NULL) && (nData >= 0x06)) {
14. unsigned long nLength;
15. unsigned char bTemp[3];
16. nLength = (unsigned long) sprintf((char*) bTemp, "%c%02X", SOH, bAddress);
17. if (memcmp(pData, bTemp, nLength) == 0) {
18. if (pData[nData - 0x02] == checksum__bb_cmd(pData, nData - 0x02))
19. fRetVal = __TRUE;
20. }
21. }
22.
23. return fRetVal;
24. }
25.
26. bool process_reply(unsigned char* pData, unsigned long nData, unsigned char* pDataReply,
27. unsigned long* nDataReply, bool fIsRawCommand) {
28. bool fRetVal = __FALSE;
29.
30. // Check if command is correct
31. if (check_command(0xFF, pData, nData)) {
32. switch (pData[0x03]) {
33. case STX:
34. fRetVal = __TRUE;
35. // Convert to hex
36. multi_dc2b(pDataReply, pData + (fIsRawCommand ? 4 : 6),
37. nData - (fIsRawCommand ? 7 : 9));
38. *nDataReply = (nData - (fIsRawCommand ? 7 : 9)) >> 1;
39. break;
40.
41. case ACK:
42. fRetVal = __TRUE;
43. break;
44.
45. case NAK:
46. fRetVal = __FALSE;
47. break;
48. }
49. }
50.
51. return fRetVal;
52. }

```

For example to decode a “FW Version Reading” reply

```

1. // add code to get the reply (from 0x01 to 0x0D chars) with buffer overflow check algorithm here
2.

```

```
3. char szVersion[65];
4. unsigned char pDataReply[1024];
5. unsigned long nDataReply = 0;
6.
7. if (process_reply(pData, nData, pDataReply, &nDataReply, __FALSE)) {
8. memset((unsigned char*) szVersion, 0, sizeof(szVersion));
9. memcpy((unsigned char*) szVersion, pDataReply, nDataReply);
10. // the firmware version is in szVersion
11. }
```

## 5 Document Revision History

| Date     | Revision | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 30.08.16 | 1.00     | Initial release                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 01.09.16 | 1.01     | <p>Added 5221L, 5222L, 5231L, 5232L, 5241L and 5242L readers support to this manual.</p> <p>Added 5221U-S, 5222U-S, 5237U-S, 5238U-S, 5231U and 5232U 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> |
| 19.09.16 | 1.02     | <p>Added 5325U, 5325U-RTC, 5335U, 5335U-RTC, 5345U, 5345U-RTC, 5326U, 5326U-RTC, 5336U, 5336U-RTC, 5346U and 5346U-RTC readers support to this manual.</p> <p>Added 5426U, 5426U-RTC, 5426U-G, 5426U-RTC-G, 5526U, 5526U-RTC, 5526U-G and 5526U-RTC-G readers support to this manual.</p> <p>Added 'Temperature Reading' reply format for readers different from 52xxL, 52xxH and 52xxU.</p> <p>Updated the 'Reading Test Activation/Deactivation' command.</p> <p>Added the 'Power Test' command. Updated the supported commands table in appendix E.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 26.09.16 | 1.03     | <p>Added the 'Antennas Auto-Tuning' command.</p> <p>Added 1021U, 1031U, 1021U-S, 1031U-S, 1041U, 1051U, 1061U, 1071U, 1061U-S, 1071U-S, 5224U and 5225U readers support to this manual.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 30.09.16 | 1.04     | <p>Updated the 5221U-S, 5222U-S, 5237U-S, 5238U-S, 5231U and 5232U firmware versions object of this manual.</p> <p>Added 'RF Power Test' command to 5221U-S, 5222U-S, 5237U-S, 5238U-S, 5231U and 5232U readers.</p> <p>Added 'Antennas Auto-Tuning' command to 5031U and 5032U readers.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 04.10.16 | 1.05     | Added 3122L, 3122L-I, 3122H, 3122H-I, 3122U, 3122U I, 5121L, 5131L, 5121H, 5131H, 5121U and 5131U readers support to this manual.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 25.10.16 | 1.06     | <p>Added 7027U reader support to this manual.</p> <p>Added 'Number of Records Reading', 'Record Database Reset', 'Current Record Request' and 'Current Record Dequeue' commands.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23.12.16 | 1.07     | <p>Added 1021N and 1031N readers support to this manual.</p> <p>Added 'ISO 14443A-4 Transponder RATS Command' and 'ISO 14443A-4 Transponder Generic Command' commands.</p> <p>Deleted supported tags appendixes.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 12.07.17 | 1.08     | <p>Corrected errors in table with supported readers object of this manual.</p> <p>Corrected errors in table with firmware versions object of this manual.</p> <p>Corrected all the read and write commands titles.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| Date     | Revision | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          |          | Added the variable size (max 240 bytes) null terminated string parameters management in 'Read Configuration Parameters' and 'Write Configuration Parameters' commands.<br>Fixed the 'Buffer Data Request', 'Queue Data Request' and 'Spontaneous Message' reply data in case of UHF device.                                                                                                                                                                                                                                                                                                                                                                                                        |
| 10.10.17 | 1.09     | Added 1021L, 1031L and 1041L readers support to this manual.<br>Added 1021H, 1031H, 1041H and 1051H readers support to this manual.<br>Added 3122N and 3122N-I readers support to this manual.<br>Added 3221L, 3222L, 3221N and 3222N readers support to this manual.<br>Updated reader's firmware versions object of this manual.<br>Added 'Write Data to an HITAG 2 Transponder', 'Read ID of an HITAG 2 Transponder', 'Read Page of an HITAG 2 Transponder' and 'Write Page of an HITAG 2 Transponder' commands.                                                                                                                                                                                |
| 07.11.17 | 1.10     | Updated reader's firmware versions object of this manual.<br>Added the MIFARE DESFire Transponder 'Generic Command' with subcommands.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 15.01.18 | 1.11     | Updated reader's firmware versions object of this manual.<br>Added the 5325U-FCC, 5325U-RTC-FCC, 5326U-FCC and 5326U-RTC-FCC readers support to this manual.<br>Added the 'RF Sensitivity' Test, Read Reflected Power and Read RSSI Power commands.<br>Added the reading timestamp information to 'Buffer Data Request', 'Queue Data Request' and 'Spontaneous Message'.<br>Added the 'QT Read' and 'QT Write' commands of an Impinj Monza 4QT transponder.<br>Added the 'Read Sensor Code' and 'Read On-Chip RSSI' commands of an RFMicron Magnus S2 and S3 transponders.<br>Added the 'Read Temperature Code' of an RFMicron Magnus S3. Splitted the supported commands table per device family. |
| 30.01.18 | 1.12     | Updated reader's firmware versions object of this manual.<br>Added the 3122H-E, 3122H-K, 3122N-M, 3122U-K and 7062L readers support to this manual.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 08.02.18 | 1.13     | Updated reader's firmware versions object of this manual.<br>Corrected the 'Write Data to an HITAG S Transponder' command reply.<br>Cancelled the 'Write Data to an HITAG 2 Transponder' command.<br>Corrected the antenna nr 2 support in '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.<br>Added the PicoPass transponder 'Inventory' command.<br>Updated the supported commands tables.                                                                                                                                                                                                                  |
| 29.05.18 | 1.14     | Updated reader's firmware versions object of this manual.<br>Added 'Re-Read an Unqueued Record', 'Start Continuous Read Records' and 'Stop Continuous Read Records' commands.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 08.06.18 | 1.15     | Updated reader's firmware versions object of this manual.<br>Fixed the 'Read Temperature' and 'Read Date/Time' commands.<br>Updated the supported commands tables.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 05.07.18 | 1.16     | Added the 'Firmware Upgrade' command and procedure.<br>Updated the supported commands tables.<br>Added the XMODEM protocol overview.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 01.08.18 | 1.17     | Updated reader's firmware versions object of this manual.<br>Added the RSSI Q and I channel info in 'Buffer Data Request', 'Queue Data Request' and 'Spontaneous Message'.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 22.10.18 | 1.18     | Updated reader's firmware versions object of this manual.<br>Added the 5721U reader support to this manual.<br>Removed custom readers support from this manual.<br>Added the tag read count info in 'Buffer Data Request'.                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

| Date     | Revision | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          |          | Updated the 'Buffer Data Request', 'Queue Data Request' and 'Spontaneous Message' examples.<br>Minor corrections.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 26.10.18 | 1.19     | Added the 3223L and 3223N readers support to this manual.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 14.12.18 | 1.20     | Updated reader's firmware versions object of this manual.<br>Added the 'Sleep Mode' command.<br>Updated the supported commands tables.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 11.01.19 | 1.21     | Updated reader's firmware versions object of this manual.<br>Added the max RSSI Q and I channel info in 'Buffer Data Request'.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 27.05.19 | 1.22     | Added the 5327U[-FCC], 5337U, 5347U, 5328U[-RTC][FCC], 5338U[-RTC], 5348U[-RTC], 5427U[-G], 5428U[-RTC][-G], 5527U[-G], 5528U[-RTC][-G] and 5346U-BL readers support to this manual.<br>Replaced the 'Read MAC Address' command with 'Read Ethernet MAC Address' command.<br>Added the 'Read Bluetooth MAC Address' command.<br>Updated the supported commands tables.                                                                                                                                                                                                                                                                                                                                                             |
| 02.09.19 | 1.23     | Updated reader's firmware versions object of this manual.<br>Replaced 'Write General Parameters' command with 'Write ROM General Parameters' command.<br>Replaced 'Write Configuration Parameters' command with 'Write ROM Configuration Parameters' command.<br>Added the 'Write RAM Configuration Parameters' command.<br>Replaced the 'Set Default Parameters' command with 'Write ROM Default Parameters' command.<br>Replaced the 'Read General Parameters' command with 'Read RAM General Parameters' command.<br>Replaced the 'Read Configuration Parameters' command with 'Read RAM Configuration Parameters' command.<br>Added the 'Read ROM Configuration Parameters' command.<br>Updated the supported commands tables. |
| 14.11.19 | 1.24     | Updated reader's firmware versions object of this manual.<br>Removed custom readers from this manual.<br>Replaced ISO 18000-6C with ISO 18000-63.<br>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.<br>Updated the supported commands tables.                                                                                                                                                                                                                                                                                                                                                                                                  |
| 04.05.20 | 1.25     | Added the 1021U-FCC, 1021U-BRA, 1021U-S-FCC, 1021U-S-BRA, 1031U-FCC, 1031U-BRA, 1031U-S-FCC, 1031U-S-BRA, 1041U-FCC, 1041U-BRA, 1051U-FCC, 1051U-BRA, 1061U-FCC, 1061U-BRA, 1061U-S-FCC, 1061U-S-BRA, 1071U-FCC, 1071U-BRA, 1071U-S-FCC, 1071U-S-BRA readers support to this manual.<br>Updated the reader's description object of this manual.<br>Updated the supported commands tables.<br>Format changes and document fixes in all sections.                                                                                                                                                                                                                                                                                    |
| 07.06.21 | 1.26     | Added the 1021L v4 and 1031L v4 readers support to this manual.<br>Added the 1021H v2 and 1021H v2 readers support to this manual.<br>Added 3122L v2+4 and 3122H v2+2 readers support to this manual.<br>Added 3221L v4, 3222L v4 and 3223L v1+4 readers support to this manual.<br>Updated the reader's description object of this manual.<br>Updated the supported commands tables.                                                                                                                                                                                                                                                                                                                                              |
| 20.12.21 | 1.27     | Updated the reader's description object of this manual.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

| Date     | Revision | Description                                                                                       |
|----------|----------|---------------------------------------------------------------------------------------------------|
| 24.01.22 | 1.28     | Updated the reader's description object of this manual.<br>Updated the supported commands tables. |
| 05.08.25 | 1.29     | Format changes and document fixes in all sections.<br>Removed XMODEM Protocol Overview.           |

## 6 Supported Commands Table

### 6.1 OEM Devices

|                                       | 1021L v3, 1031L v3,<br>1041L v3 | 1021L v4, 1031L v4 | 1021H v1, 1031H v1,<br>1041H v1 | 1051H v1 | 1021H v2, 1031H v2 | 1021N v1, 1031N v1 | 1021U, 1031U, 1061U,<br>1071U | 1041U, 1051U |
|---------------------------------------|---------------------------------|--------------------|---------------------------------|----------|--------------------|--------------------|-------------------------------|--------------|
| Device Reset                          | ✓                               | ✓                  | ✓                               | ✓        | ✓                  | ✓                  | ✓                             | ✓            |
| Read Device Serial Number             |                                 | ✓                  |                                 |          | ✓                  | ✓                  | ✓                             | ✓            |
| Read Ethernt MAC Address              |                                 |                    |                                 |          |                    |                    |                               |              |
| Read Bluetooth MAC Address            |                                 |                    |                                 |          |                    |                    |                               |              |
| Read Firmware Version                 | ✓                               | ✓                  | ✓                               | ✓        | ✓                  | ✓                  | ✓                             | ✓            |
| Firmware Upgrade                      | ✓                               | ✓                  | ✓                               | ✓        | ✓                  | ✓                  | ✓                             | ✓            |
| Read Temperature                      |                                 |                    |                                 |          |                    |                    |                               |              |
| Read Date/Time                        |                                 |                    |                                 |          |                    |                    |                               |              |
| Write Date/Time                       |                                 |                    |                                 |          |                    |                    |                               |              |
| Write ROM General Parameters          | ✓                               | ✓                  | ✓                               | ✓        | ✓                  | ✓                  | ✓                             | ✓            |
| Write RAM Configuration Parameters    |                                 |                    |                                 |          |                    |                    | ✓                             | ✓            |
| Write ROM Configuration Parameters    |                                 |                    |                                 |          |                    |                    | ✓                             | ✓            |
| Write ROM Default Parameters          | ✓                               | ✓                  | ✓                               | ✓        | ✓                  | ✓                  | ✓                             | ✓            |
| Read RAM General Parameters           | ✓                               | ✓                  | ✓                               | ✓        | ✓                  | ✓                  | ✓                             | ✓            |
| Read RAM Configuration Parameters     |                                 |                    |                                 |          |                    |                    | ✓                             | ✓            |
| Read ROM Configuration Parameters     |                                 |                    |                                 |          |                    |                    | ✓                             | ✓            |
| Sleep Mode                            |                                 |                    |                                 |          |                    |                    | ✓                             | ✓            |
| '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                    | ✓                               | ✓                  | ✓                               | ✓        | ✓                  | ✓                  | ✓                             | ✓            |
| Read Number of Records                |                                 |                    |                                 |          |                    |                    |                               |              |
| Reset Record Database                 |                                 |                    |                                 |          |                    |                    |                               |              |
| Request Current Record                |                                 |                    |                                 |          |                    |                    |                               |              |
| Unqueue Current Record                |                                 |                    |                                 |          |                    |                    |                               |              |
| Re-Read an Unqueued Record            |                                 |                    |                                 |          |                    |                    |                               |              |
| Start Continuous Read Records         |                                 |                    |                                 |          |                    |                    |                               |              |
| Stop Continuous Read Records          |                                 |                    |                                 |          |                    |                    |                               |              |
| Write Data to an EM4305 Transponder   | ✓                               | ✓                  |                                 |          |                    |                    |                               |              |
| Read ID Code of an EM4305 Transponder | ✓                               | ✓                  |                                 |          |                    |                    |                               |              |

|                                                               | 1021L v3, 1031L v3, 1041L v3 | 1021L v4, 1031L v4 | 1021H v1, 1031H v1, 1041H v1 | 1051H v1 | 1021H v2, 1031H v2 | 1021N v1, 1031N v1 | 1021U, 1031U, 1061U, 1071U | 1041U, 1051U |
|---------------------------------------------------------------|------------------------------|--------------------|------------------------------|----------|--------------------|--------------------|----------------------------|--------------|
| 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                        | ✓                            | ✓                  |                              |          |                    |                    |                            |              |
| 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 'Generic' Command                       |                              |                    | ✓                            |          | ✓                  | ✓                  |                            |              |
| ISO 14443A Transponders 'Inventory' Command                   |                              |                    | ✓                            | ✓        | ✓                  | ✓                  |                            |              |
| Read a Data Block of a MIFARE Mini/1k/4k (UID 4) Transponder  |                              |                    | ✓                            | ✓        | ✓                  | ✓                  |                            |              |
| Write a Data Block of a MIFARE Mini/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 14443-4 Transponder 'Generic' Command                     |                              |                    | ✓                            | ✓        | ✓                  | ✓                  |                            |              |
| MIFARE DESFire 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                  |                              |                    |                              |          |                    |                    | ✓                          | ✓            |



|                                                                           | 1021L v3, 1031L v3,<br>1041L v3 | 1021L v4, 1031L v4 | 1021H v1, 1031H v1,<br>1041H v1 | 1051H v1 | 1021H v2, 1031H v2 | 1021N v1, 1031N v1 | 1021U, 1031U, 1061U,<br>1071U | 1041U, 1051U |
|---------------------------------------------------------------------------|---------------------------------|--------------------|---------------------------------|----------|--------------------|--------------------|-------------------------------|--------------|
| 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                             |                                 |                    |                                 |          |                    |                    | ✓                             | ✓            |
| '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 S2 / S3 Transponder |                                 |                    |                                 |          |                    |                    | ✓                             | ✓            |

## 6.2 Desktop Devices

|                                       | 3122L v2+3 | 3122L v2+4 | 3122H v2+1 | 3122H v2+2 | 3122N v2+1 | 3122U | 3221L v3, 3222L v3 | 3221L v4, 3122L v4 | 3221N v1, 3222N v1 | 3221N v2, 3222N v2 | 3223L v1+3 | 3223L v1+4 | 3223N v1+1 | 3223N v1+2 |
|---------------------------------------|------------|------------|------------|------------|------------|-------|--------------------|--------------------|--------------------|--------------------|------------|------------|------------|------------|
| Device Reset                          | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     | ✓                  | ✓                  | ✓                  | ✓                  | ✓          | ✓          | ✓          | ✓          |
| Read Device Serial Number             | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     |                    | ✓                  | ✓                  | ✓                  | ✓          | ✓          | ✓          | ✓          |
| Read Ethernet MAC Address             |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| Read Bluetooth MAC Address            |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| Read Firmware Version                 | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     | ✓                  | ✓                  | ✓                  | ✓                  | ✓          | ✓          | ✓          | ✓          |
| Firmware Upgrade                      | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     | ✓                  | ✓                  | ✓                  | ✓                  | ✓          | ✓          | ✓          | ✓          |
| Read Temperature                      |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| Read Date/Time                        |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| Write Date/Time                       |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| Write ROM General Parameters          | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     | ✓                  | ✓                  | ✓                  | ✓                  | ✓          | ✓          | ✓          | ✓          |
| Write RAM Configuration Parameters    | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     |                    |                    |                    |                    | ✓          | ✓          | ✓          | ✓          |
| Write ROM Configuration Parameters    | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     |                    |                    |                    |                    | ✓          | ✓          | ✓          | ✓          |
| Write ROM Default Parameters          | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     | ✓                  | ✓                  | ✓                  | ✓                  | ✓          | ✓          | ✓          | ✓          |
| Read RAM General Parameters           | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     | ✓                  | ✓                  | ✓                  | ✓                  | ✓          | ✓          | ✓          | ✓          |
| Read RAM Configuration Parameters     | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     |                    |                    |                    |                    | ✓          | ✓          | ✓          | ✓          |
| Read ROM Configuration Parameters     | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     |                    |                    |                    |                    | ✓          | ✓          | ✓          | ✓          |
| Sleep Mode                            |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| '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                    | ✓          | ✓          | ✓          | ✓          | ✓          | ✓     | ✓                  | ✓                  | ✓                  | ✓                  | ✓          | ✓          | ✓          | ✓          |
| Read Number of Records                |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| Reset Record Database                 |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| Request Current Record                |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| Unqueue Current Record                |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| Re-Read an Unqueued Record            |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| Start Continuous Read Records         |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| Stop Continuous Read Records          |            |            |            |            |            |       |                    |                    |                    |                    |            |            |            |            |
| 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   | ✓          | ✓          |            |            |            |       | ✓                  | ✓                  |                    |                    | ✓          | ✓          |            |            |

|                                                               | 3122L v2+3 | 3122L v2+4 | 3122H v2+1 | 3122H v2+2 | 3122N v2+1 | 3122U | 3221L v3, 3222L v3 | 3221L v4, 3122L v4 | 3221N v1, 3222N v1 | 3221N v2, 3222N v2 | 3223L v1+3 | 3223L v1+4 | 3223N v1+1 | 3223N v1+2 |
|---------------------------------------------------------------|------------|------------|------------|------------|------------|-------|--------------------|--------------------|--------------------|--------------------|------------|------------|------------|------------|
| 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                        | ✓          | ✓          |            |            |            |       | ✓                  | ✓                  |                    |                    | ✓          | ✓          |            |            |
| 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 'Generic' Command                       |            |            | ✓          | ✓          | ✓          |       |                    |                    | ✓                  | ✓                  |            |            | ✓          | ✓          |
| ISO 14443A Transponders 'Inventory' Command                   |            |            | ✓          | ✓          | ✓          |       |                    |                    | ✓                  | ✓                  |            |            | ✓          | ✓          |
| Read a Data Block of a MIFARE Mini/1k/4k (UID 4) Transponder  |            |            | ✓          | ✓          | ✓          |       |                    |                    | ✓                  | ✓                  |            |            | ✓          | ✓          |
| Write a Data Block of a MIFARE Mini/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 14443*-4 Transponder 'Generic' Command                    |            |            | ✓          | ✓          | ✓          |       |                    |                    | ✓                  | ✓                  |            |            | ✓          | ✓          |
| MIFARE DESFire 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                      |            |            |            |            | ✓          |       |                    |                    |                    |                    |            |            |            |            |

|                                                                           | 3122L v2+3 | 3122L v2+4 | 3122H v2+1 | 3122H v2+2 | 3122N v2+1 | 3122U | 3221L v3, 3222L v3 | 3221L v4, 3122L v4 | 3221N v1, 3222N v1 | 3221N v2, 3222N v2 | 3223L v1+3 | 3223L v1+4 | 3223N v1+1 | 3223N v1+2 |
|---------------------------------------------------------------------------|------------|------------|------------|------------|------------|-------|--------------------|--------------------|--------------------|--------------------|------------|------------|------------|------------|
| Write Data of an ISO 18000-63 Transponder                                 |            |            |            |            | ✓          |       |                    |                    |                    |                    |            |            |            |            |
| Lock Data of an ISO 18000-63 Transponder                                  |            |            |            |            | ✓          |       |                    |                    |                    |                    |            |            |            |            |
| 'Kill' Command of an ISO 18000-63 Transponder                             |            |            |            |            | ✓          |       |                    |                    |                    |                    |            |            |            |            |
| '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 S2 / S3 Transponder |            |            |            |            | ✓          |       |                    |                    |                    |                    |            |            |            |            |

### 6.3 Industrial Devices

|                                       | 5121L, 5131L | 5121H, 5131H | 5121U, 5131U | 5221L, 5222L, 5231L, | 5221H, 5222H, 5231H, | 5221U-S, 5222U-S, | 5231U, 5232U | 5224U, 5225U | 532xU, 533xU, 534xU | 532xU-RTC, 533xU-RTC, | 5346U-BL | 542xU, 552xU | 542xU-RTC, 552xU-RTC | 5721U |
|---------------------------------------|--------------|--------------|--------------|----------------------|----------------------|-------------------|--------------|--------------|---------------------|-----------------------|----------|--------------|----------------------|-------|
| Device Reset                          | ✓            | ✓            | ✓            | ✓                    | ✓                    | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Read Device Serial Number             | ✓            | ✓            | ✓            | ✓                    | ✓                    | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Read Ethernet MAC Address             |              |              |              | ✓                    | ✓                    | ✓                 | ✓            |              | ✓                   | ✓                     | ✓        | ✓            | ✓                    |       |
| Read Bluetooth MAC Address            |              |              |              |                      |                      |                   |              |              |                     |                       | ✓        |              |                      |       |
| Read Firmware Version                 | ✓            | ✓            | ✓            | ✓                    | ✓                    | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Firmware Upgrade                      | ✓            | ✓            | ✓            | ✓                    | ✓                    | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Read Temperature                      |              |              |              | ✓                    | ✓                    | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    |       |
| Read Date/Time                        |              |              |              | ✓                    | ✓                    | ✓                 | ✓            |              |                     | ✓                     | ✓        |              | ✓                    |       |
| Write Date/Time                       |              |              |              | ✓                    | ✓                    | ✓                 | ✓            |              |                     | ✓                     | ✓        |              | ✓                    |       |
| Write ROM General Parameters          | ✓            | ✓            | ✓            | ✓                    | ✓                    | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Write RAM Configuration Parameters    |              |              |              |                      |                      |                   |              | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Write ROM Configuration Parameters    | ✓            | ✓            | ✓            | ✓                    | ✓                    | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Write ROM Default Parameters          | ✓            | ✓            | ✓            | ✓                    | ✓                    | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Read RAM General Parameters           | ✓            | ✓            | ✓            | ✓                    | ✓                    | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Read RAM Configuration Parameters     | ✓            | ✓            | ✓            | ✓                    | ✓                    | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Read ROM Configuration Parameters     |              |              |              |                      |                      |                   |              | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Sleep Mode                            |              |              |              |                      |                      |                   |              |              |                     |                       |          |              |                      |       |
| '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                    | ✓            | ✓            | ✓            | ✓                    | ✓                    | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Read Number of Records                |              |              |              |                      |                      |                   |              |              | ✓                   | ✓                     | ✓        | ✓            | ✓                    |       |
| Reset Record Database                 |              |              |              |                      |                      |                   |              |              | ✓                   | ✓                     | ✓        | ✓            | ✓                    |       |
| Request Current Record                |              |              |              |                      |                      |                   |              |              | ✓                   | ✓                     | ✓        | ✓            | ✓                    |       |
| Unqueue Current Record                |              |              |              |                      |                      |                   |              |              | ✓                   | ✓                     | ✓        | ✓            | ✓                    |       |
| Re-Read an Unqueued Record            |              |              |              |                      |                      |                   |              |              | ✓                   | ✓                     | ✓        | ✓            | ✓                    |       |
| Start Continuous Read Records         |              |              |              |                      |                      |                   |              |              | ✓                   | ✓                     | ✓        | ✓            | ✓                    |       |
| Stop Continuous Read Records          |              |              |              |                      |                      |                   |              |              | ✓                   | ✓                     | ✓        | ✓            | ✓                    |       |
| 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   | ✓            |              |              | ✓                    |                      |                   |              |              |                     |                       |          |              |                      |       |

|                                                               | 5121L, 5131L | 5121H, 5131H | 5121U, 5131U | 5221L, 5222L, 5231L, | 5221H, 5222H, 5231H, | 5221U-S, 5222U-S, | 5231U, 5232U | 5224U, 5225U | 532xU, 533xU, 534xU | 532xU-RTC, 533xU-RTC, | 5346U-BL | 542xU, 552xU | 542xU-RTC, 552xU-RTC | 5721U |
|---------------------------------------------------------------|--------------|--------------|--------------|----------------------|----------------------|-------------------|--------------|--------------|---------------------|-----------------------|----------|--------------|----------------------|-------|
| 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                        | ✓            |              |              | ✓                    |                      |                   |              |              |                     |                       |          |              |                      |       |
| 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 'Generic' Command                       |              | ✓            |              |                      | ✓                    |                   |              |              |                     |                       |          |              |                      |       |
| ISO 14443A Transponders 'Inventory' Command                   |              | ✓            |              |                      | ✓                    |                   |              |              |                     |                       |          |              |                      |       |
| Read a Data Block of a MIFARE Mini/1k/4k (UID 4) Transponder  |              | ✓            |              |                      | ✓                    |                   |              |              |                     |                       |          |              |                      |       |
| Write a Data Block of a MIFARE Mini/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 14443°-4 Transponder 'Generic' Command                    |              | ✓            |              |                      | ✓                    |                   |              |              |                     |                       |          |              |                      |       |
| MIFARE DESFire 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                      |              |              | ✓            |                      |                      | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |

|                                                                           | 5121L, 5131L | 5121H, 5131H | 5121U, 5131U | 5221L, 5222L, 5231L, | 5221H, 5222H, 5231H, | 5221U-S, 5222U-S, | 5231U, 5232U | 5224U, 5225U | 532xU, 533xU, 534xU | 532xU-RTC, 533xU-RTC, | 5346U-BL | 542xU, 552xU | 542xU-RTC, 552xU-RTC | 5721U |
|---------------------------------------------------------------------------|--------------|--------------|--------------|----------------------|----------------------|-------------------|--------------|--------------|---------------------|-----------------------|----------|--------------|----------------------|-------|
| Write Data of an ISO 18000-63 Transponder                                 |              |              | ✓            |                      |                      | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| Lock Data of an ISO 18000-63 Transponder                                  |              |              | ✓            |                      |                      | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| 'Kill' Command of an ISO 18000-63 Transponder                             |              |              | ✓            |                      |                      | ✓                 | ✓            | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |
| '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 S2 / S3 Transponder |              |              | ✓            |                      |                      |                   |              | ✓            | ✓                   | ✓                     | ✓        | ✓            | ✓                    | ✓     |