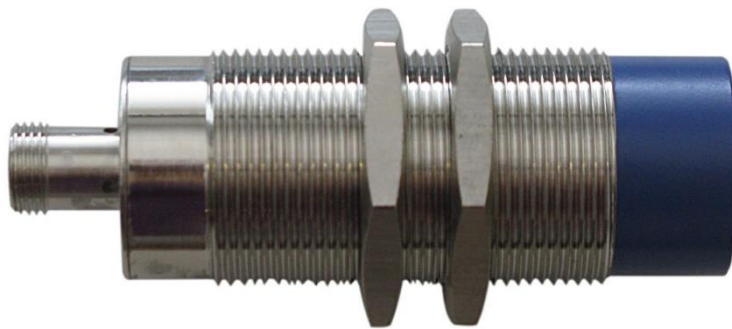


# **125 kHz RFID System**



## **M30 Form Factor BLUEBOX ADVANT LF**

**BLUEBOX**  
RFid System

**RS232 / RS485**

From firmware release 3.13

## Preface

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

**Safety Instructions / Warning - Read before start-up!**

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

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

Issue 1.01a  
– 21. September 2022 –

Phone: +49 621 6690094-0  
Fax: +49 621 6690094-9  
E-Mail: [info@idtronic.de](mailto:info@idtronic.de)  
Web: [idtronic.de](http://idtronic.de)

Subject to alteration without prior notice.  
© Copyright iDTRONIC GmbH 2022  
Printed in Germany

## Table of Contents

1	Introduction.....	6
2	Technical Specifications.....	7
3	Operating Features .....	10
3.1	General Parameters.....	10
4	Communication Features .....	12
4.1	Device Startup.....	13
4.2	Device Reset .....	14
4.3	General Parameters Programming .....	14
4.4	Default Parameters Programming.....	16
4.5	General Parameters Reading .....	16
4.6	FW Version Reading .....	18
4.7	Data Request .....	18
4.8	Queue Data Request.....	19
4.9	Status Reading .....	20
4.10	RF Deactivation.....	21
4.11	RF Activation .....	21
4.12	Write Data to EM4305 Transponder .....	22
4.13	Read ID Code of a EM4305 Transponder .....	23
4.14	Write Data to T5557 Transponder.....	23
4.15	Read ID Code of a T5557 Transponder.....	24
4.16	Write Data to Q5 Transponder .....	25
4.17	Read ID Code of a Q5 Transponder.....	26
4.18	Write Data to HITAG S Transponder .....	26
4.19	Read ID Code of a HITAG 1 / HITAG S Transponder.....	27
4.20	Read Page of a HITAG 1 / HITAG S Transponder .....	28
4.21	Write Page of a HITAG 1 / HITAG S Transponder.....	29
4.22	'Reset' TITAN Transponder .....	30
4.23	'Login' TITAN Transponder .....	30
4.24	'Write Password' TITAN Transponder .....	31
4.25	'Standard Read' TITAN Transponder .....	32
4.26	'Selective Read' TITAN Transponder .....	33
4.27	'Write Word' TITAN Transponder .....	34
4.28	'Write Several Words' TITAN Transponder .....	35
4.29	'Read After Write Word' TITAN Transponder .....	36
4.30	'Spontaneous' Message .....	37
4.30.1	RS232 .....	37
5	Connections.....	38
6	Installation .....	41
7	Status Indications: LEDs .....	43
8	Document Revision History .....	44
A.	Supported Transponders .....	45
B.	Nibble Coding of Transponder Codes .....	48



## 1 Introduction

The **BLUEBOX ADVANT M30 LF** hereinafter named **BLUEBOX** is a little (dimensions of the cylindrical case  $M30 \times 1.5 \times 78$  mm) read/write RFID device operating at 125 kHz and suitable for industrial application. The **BLUEBOX** communicates with a 'host' system (typically a PC or a PLC) through an RS232 (items 5224L and 5227L) or an RS485 (items 5225L or 5228L) serial line and acts as a joint through a set of commands between the host system and a RFID tag present near the antenna. A 'master/slave' protocol is used for the communication between the 'host' system and the **BLUEBOX**. Through the serial line, it is also possible to configure the functional parameters and to upgrade the firmware, the 'BLUEBOX Show' program of the SDK is foreseen to explicate these operations. The **BLUEBOX** is furnished with an integrated RF antenna inside the case and with a 4-pole M12 A-coded male connector (items 5224L and 5225L) or with an open end 1.5mt cable (items 5227L and 5228L).

Hereinafter the available ordering codes.

Ordering Code	Description	Interface
<b>5224L</b>	Read / write 125 kHz RFID device with integrated antenna. Serial RS232 communication interface. M12 connection.	RS232
<b>5225L</b>	Read / write 125 kHz RFID device with one external antenna. Serial RS485 communication interface. M12 connection.	RS485
<b>5227L</b>	Read / write 125 kHz RFID device with integrated antenna. Serial RS232 communication interface. Cable with open end connection.	RS232
<b>5228L</b>	Read / write 125 kHz RFID device with one external antenna. Serial RS485 communication interface. Cable with open end connection.	RS485

## 2 Technical Specifications

### 5224L:

Electrical Features	
Power Supply	24Vdc $\pm 10\%$
Power Ratings	0.6W
Operating Frequency	125 kHz $\pm 2$ kHz
Antenna	Integrated
Reading Distance	10 cm <sup>1</sup>
Supported Transponders	EM410x (UNIQUE), EM4x50 (TITAN) EM4305, T5557, HITAG 1, HITAG S, Q5
Communication Interface	Serial RS232
Status Display	1 bicolor LED
Connections	4-pole M12 A-coded male connector

### 5225L:

Electrical Features	
Power Supply	24Vdc $\pm 10\%$
Power Ratings	0.6W
Operating Frequency	125 kHz $\pm 2$ kHz
Antenna	Integrated
Reading Distance	10 cm <sup>2</sup>
Supported Transponders	EM410x (UNIQUE), EM4x50 (TITAN) EM4305, T5557, HITAG 1, HITAG S, Q5
Communication Interface	Serial RS485
Status Display	1 bicolor LED
Connections	4-pole M12 A-coded male connector

<sup>1</sup> Reading distance depends on transponder type, antenna and environmental conditions.

<sup>2</sup> Reading distance depends on transponder type, antenna and environmental conditions.

## 5227L:

Electrical Features	
Power Supply	24Vdc $\pm 10\%$
Power Ratings	0.6W
Operating Frequency	125 kHz $\pm 2$ kHz
Antenna	Integrated
Reading Distance	10 cm <sup>3</sup>
Supported Transponders	EM410x (UNIQUE), EM4x50 (TITAN) EM4305, T5557, HITAG 1, HITAG S, Q5
Communication Interface	Serial RS232
Connections	2 twisted pairs overall shielded multipolar cable with free cable end. Length 1.5mt.

## 5228L:

Electrical Features	
Power Supply	24Vdc $\pm 10\%$
Power Ratings	0.6W
Operating Frequency	125 kHz $\pm 2$ kHz
Antenna	Integrated
Reading Distance	10 cm <sup>4</sup>
Supported Transponders	EM410x (UNIQUE), EM4x50 (TITAN) EM4305, T5557, HITAG 1, HITAG S, Q5
Communication Interface	Serial RS485
Connections	2 twisted pairs overall shielded multipolar cable with free cable end. Length 1.5mt.

## 5224L, 5225L, 5227L, 5228L:

<sup>3</sup> Reading distance depends on transponder type, antenna and environmental conditions.

<sup>4</sup> Reading distance depends on transponder type, antenna and environmental conditions.



### Mechanical Features

Dimensions	M30 × 1.5 × 78 mm
Material	Nickelled brass, PC
Protection Class	IP65

### 5224L, 5225L, 5227L, 5228L:

### Environmental Conditions

Operating Temperature	-10°C ... +55°C
Storage Temperature	-40°C ... +85°C
Humidity	Up to 95%, non condensing

### 3 Operating Features

In 'continuous' mode the **BLUEBOX** is characterized by the coexistence of 2 'parallel' and asynchronous activities: the transponder identification and the communication with the 'host' system. The 'continuous' identification activity interacts with the communication activity through a buffer that contains the code of the last identified transponder or the 0 code that indicates the absence of a transponder. Due to synchronization and filtering reasons, the buffer is handled by a parameter defined as 'hold time' (to be set in the range of 0 ... 99 seconds, default value 1 second) and allows to extend 'artificially' the presence of the transponder after it leaves the antenna's influence area; this behavior is observable looking at the yellow led status that is 'on' indicating the presence of a transponder. Through the command 'data request' it is possible to get the data contained in the buffer.

The **BLUEBOX** handles also a 31 elements FIFO queue which is combined with a 'filter time' parameter (to be set in a range of 0 ... 99 seconds, default value 1 second) that prevents the queue saturation in case of a transponder 'continuous' presence. When a transponder is identified, the **BLUEBOX** compares it to the previous read transponder. If the transponder is different (it is defined as 'new'), its code will be inserted in the queue and the filter time will be started. Otherwise (the transponder is the same of the previous read one), the **BLUEBOX** verifies if the filter time is expired. In this case (the filter time is expired), the transponder is defined as 'new' and will be processed as described above, otherwise only the filter time will be rearmed. Through the command 'queue data request' and the relative 'ack', it is possible to get the data contained in the queue and unload it.

In 'continuous' mode the **BLUEBOX** can be configured to obtain the behavior of a 'spontaneous' reader that will send a message on the RS232 serial line. This feature is enabled (on) / disabled (off) a flag in the general configuration of the reader.

The **BLUEBOX** allows the execution of 'on request' functions. During the execution of these functions, the 'continuous' identification activity will be suspended temporarily; the involved commands are relative to device configuration and tag read/write specific activities.

If not required, the 'continuous' identification activity can be disabled through a flag defined in the general parameters. In this case, the **BLUEBOX** will only execute the 'on request' commands already defined above.

#### 3.1 General Parameters

Hereinafter the configurable general parameter of the **BLUEBOX**.

Parameter	Description	Range	Default
Network address	Network address of the reader.	000 ... 255	255
Baud rate	Communication baud rate on RS232/RS485 interface.	1200, 2400, 4800, 9600, 19200, 38400	19200
Data bits	Data bits on RS232/RS485 interface.	7, 8	8
Stop bits	Stop bits on RS232/RS485 interface.	1, 2	1
Parity	Parity on RS232/RS485 interface.	None, even, odd	None
Hold time	Buffer management hold time.	0 ... 99 seconds	1 sec
Tag nibble coding	Tag nibble coding (see appendix B).	Normal, reverse	Normal
Filter time	Reading and tag queue management filter time.	0 ... 99 seconds 0 ... 99 minutes	1 sec
'Spontaneous' mode	Spontaneous message activation/deactivation.	Disabled, enabled	Disabled
'Continuous' mode	'Continuous' mode activation/deactivation.	Disabled, enabled	Enabled

## 4 Communication Features

The 'master/slave' protocol expects that the **BLUEBOX** (as 'slave') after the reception of a message send to him by the 'host' (as 'master'), transmits a response message after a minimum time of about 10 ms. By default, the **BLUEBOX** will apply the following parameters: address 255, baud rate 19200, 8 data bits, parity none and 1 stop bit. These parameters can be modified as specified in the 'Parameters programming' protocol command.

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

SOH	Carattere 01h (0x01)
STX	Carattere 02h (0x02)
ETX	Carattere 03h (0x03)
EOT	Carattere 04h (0x04)
ENQ	Carattere 05h (0x05)
ACK	Carattere 06h (0x06)
NAK	Carattere 15h (0x15)
SYN	Carattere 16h (0x16)
CR	Carattere 0Dh (0x0D)
'0'...'9'	Carattere 30h ...39h (0x30 ... 0x39)
'A'...'F'	Carattere 41h ...46h (0x41 ... 0x46)
<..>	Carattere 30h ...39h (0x30 ... 0x39), 41h ...46h (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 checking character or checksum and it 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 of 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** with address 1 will be: SOH '0' '1' ENQ ENQ CR (in hexadecimal: 0x01, 0x30, 0x31, 0x05, 0x05, 0x0D).

#### 4.1 Device Startup

During the startup phase, it is possible to configure the communication parameters of the **BLUEBOX** sending the following message (apply the following default communication settings 19200, n, 8, 1):

**STX '2' 'F' <addn h> <addn l> <bdr> <bit> <stop> <par> ETX <bcc> CR**

Dove:

<add h> <add l>	New address to be set. ASCII encoded byte.
<bdr>	RS232/RS485 communication interface baud rate. ASCII character: <ul style="list-style-type: none"> <li>'0' -&gt; 1200 bps;</li> <li>'1' -&gt; 2400 bps;</li> <li>'2' -&gt; 4800 bps;</li> <li>'3' -&gt; 9600 bps;</li> <li>'4' -&gt; 19200 bps</li> <li>'5' -&gt; 38400 bps.</li> </ul>
<bit>	RS232/RS485 communication interface data bits. ASCII character: <ul style="list-style-type: none"> <li>'7' -&gt; 7 bits;</li> <li>'8' -&gt; 8 bits.</li> </ul>
<stop>	RS232/RS485 communication interface stop bits. ASCII character: <ul style="list-style-type: none"> <li>'1' -&gt; 1 bit;</li> <li>'2' -&gt; 2 bits.</li> </ul>
<par>	RS232/RS485 communication interface parity. ASCII character: <ul style="list-style-type: none"> <li>'0' -&gt; None;</li> <li>'1' -&gt; Even;</li> <li>'2' -&gt; Odd.</li> </ul>

If the **BLUEBOX** is able to execute the command, it answers with:

## STX '2' 'F' '0' '0' <bcc> CR

### 4.2 Device Reset

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

The 'master' sends the following command:

**SOH <adda h> <adda l> STX '3' '0' ETX <bcc> CR**

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

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

### 4.3 General Parameters Programming

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

The 'master' sends the following command:

**SOH <adda h> <adda l> STX '2' 'F' <addn h> <addn l> <bdr> <bit> <stop> <par> <man h> <man l> <code> '0' <filt h> <filt l> <flag h> <flag l> ETX <bcc> CR**

Where:

<adda h> <adda l>	Reader address. ASCII encoded byte.
<addn h> <addn l>	New address to be set. ASCII encoded byte.
<bdr>	RS232/RS485 communication interface baud rate. ASCII character: <ul style="list-style-type: none"> <li>'0' -&gt; 1200 bps;</li> <li>'1' -&gt; 2400 bps;</li> <li>'2' -&gt; 4800 bps;</li> <li>'3' -&gt; 9600 bps;</li> <li>'4' -&gt; 19200 bps;</li> <li>'5' -&gt; 38400 bps.</li> </ul>

<bit>	RS232/RS485 communication interface data bits. ASCII character: <ul style="list-style-type: none"> <li>• '7' -&gt; 7 bits;</li> <li>• '8' -&gt; 8 bits.</li> </ul>
<stop>	RS232/RS485 communication interface stop bits. ASCII character: <ul style="list-style-type: none"> <li>• '1' -&gt; 1 bit;</li> <li>• '2' -&gt; 2 bits.</li> </ul>
<par>	RS232/RS485 communication interface parity. ASCII character: <ul style="list-style-type: none"> <li>• '0' -&gt; None;</li> <li>• '1' -&gt; Even;</li> <li>• '2' -&gt; Odd.</li> </ul>
<man h> <man l>	Hold time. ASCII encoded byte: <ul style="list-style-type: none"> <li>• Decimal 0 ... 99 for time in seconds (0 ... 99 seconds);</li> </ul>
<code>	Nibble coding. ASCII character: <ul style="list-style-type: none"> <li>• '0' -&gt; Normal;</li> <li>• '1' -&gt; Reverse.</li> </ul>
<filt h> <filt l>	Filter time. ASCII encoded byte: <ul style="list-style-type: none"> <li>• Decimal 0 ... 99 for time in seconds (0 ... 99 seconds);</li> <li>• Decimal 100 ... 199 for time in minutes (0 ... 99 minutes).</li> </ul>
<flag h> <flag l>	Flags. ASCII encoded byte whose bits are dedicated to disable (0 value) or enable (1 value) functions: <ul style="list-style-type: none"> <li>• Bit 7 ... bit4: Not used;</li> <li>• Bit 3: 'Spontaneous' mode, (1=ON);</li> <li>• Bit 2 ... bit 1: Not used;</li> <li>• Bit 0: 'Continuous' mode, (1=OFF).</li> </ul>

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

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



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

#### 4.4 Default Parameters Programming

This command is used to set the default values of the communication and 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** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

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



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

#### 4.5 General Parameters Reading

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

The 'master' sends the following command:

**SOH <add h> <add l> STX '2' 'A' ETX <bcc> CR**

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** able to execute the command), it answers with:



**SOH <add h> <add l> STX '2' 'A' <add h> <add l> <bdr> <bit> <stop>  
 <par> <man h> <man l> <code> '0' <filt h> <filt l> <flag h> <flag l>  
 ETX <bcc> CR**

Where:

<adda h> <adda l>	Reader address. ASCII encoded byte.
<bdr>	RS232/RS485 communication interface baud rate. ASCII character: <ul style="list-style-type: none"> <li>• '0' -&gt; 1200 bps;</li> <li>• '1' -&gt; 2400 bps;</li> <li>• '2' -&gt; 4800 bps;</li> <li>• '3' -&gt; 9600 bps;</li> <li>• '4' -&gt; 19200 bps</li> <li>• '5' -&gt; 38400 bps.</li> </ul>
<bit>	RS232/RS485 communication interface data bits. ASCII character: <ul style="list-style-type: none"> <li>• '7' -&gt; 7 bits;</li> <li>• '8' -&gt; 8 bits.</li> </ul>
<stop>	RS232/RS485 communication interface stop bits. ASCII character: <ul style="list-style-type: none"> <li>• '1' -&gt; 1 bit;</li> <li>• '2' -&gt; 2 bits.</li> </ul>
<par>	RS232/RS485 communication interface parity. ASCII character: <ul style="list-style-type: none"> <li>• '0' -&gt; None;</li> <li>• '1' -&gt; Even;</li> <li>• '2' -&gt; Odd.</li> </ul>
<man h> <man l>	Hold time. ASCII encoded byte: <ul style="list-style-type: none"> <li>• Decimal 0 ... 99 for time in seconds (0 ... 99 seconds);</li> </ul>
<code>	Nibble coding. ASCII character: <ul style="list-style-type: none"> <li>• '0' -&gt; Normal;</li> <li>• '1' -&gt; Reverse.</li> </ul>
<filt h> <filt l>	Filter time. ASCII encoded byte: <ul style="list-style-type: none"> <li>• Decimal 0 ... 99 for time in seconds (0 ... 99 seconds);</li> <li>• Decimal 100 ... 199 for time in minutes (0 ... 99 minutes).</li> </ul>

<flag h> <flag l>

Flags. ASCII encoded byte whose bits are dedicated to disable (0 value) or enable (1 value) functions:

- Bit 7 ... bit4: Not used;
- Bit 3: 'Spontaneous' mode, (1=ON);
- Bit 2 ... bit 1: Not used;
- Bit 0: 'Continuous' mode, (1=OFF).

#### 4.6 FW Version Reading

The 'master' sends the following command:

**SOH <add h> <add l> STX '3' '4' ETX <bcc> CR**

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** able to execute the command), it answers with:

**SOH <add h> <add l> STX '3' '4' <vf 01 h> <vf 01 l> <vf 02 h> <vf 02 l> ... <vf 15 h> <vf 15 l> <vf 16 h> <vf 16 l> ETX <bcc> CR**

Where:

<vf 01 h> <vf 01 l>

ASCII coding of the byte 1 of the string.

...

...

<vf 16 h> <vf 16 l>

ASCII coding of the byte 16 of the string.

In this case the 16 bytes are represented by a string of 16 ASCII characters that define the version. Example 'TINYOEM\_LF 1.00' indicates that this is a **TINYOEM** in **LF** configuration (**L**ow **F**requency 125 kHz) with firmware version **1.00**.

#### 4.7 Data Request

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

The 'master' sends the following command:

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

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

**SOH <add h> <add l> STX <code 1 h> <code 1 l> ... <code i h> <code i l> ... <code n h> <code n l> ETX <bcc> CR**

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>
<code i h> <code i l>	i-th byte of the code of the identified tag. ASCII encoded byte.

If the **BLUEBOX** doesn't have any valid UID (no tag present), it will answer with:

**SOH <add h> <add l> STX '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' ETX <bcc> CR**

demonstrating to the 'master' its presence in the network.

#### 4.8 Queue Data Request

In 'continuous' mode, when the **BLUEBOX** finds a 'new' transponder, it inserts the 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** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

**SOH <add h> <add l> STX <code 1 h> <code 1 l> ... <code i h> <code i l> ... <code n h> <code n l> ETX <bcc> CR**

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>
<code i h> <code i l>	i-th byte of the code of the identified tag. ASCII encoded byte.

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**

demonstrating to the 'master' its presence in the network.

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

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

## 4.9 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 <add1> <add0> STX '3' '6' ETX <bcc> CR**

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

**SOH <add h> <add l> STX '3' '6' <sta hh> <sta hl> <sta lh> <sta ll> ETX <bcc> CR**

Where:

<sta hh> <sta hl> <sta  
lh> <sta ll>

BLUEBOX status. ASCII encoded word whose bits has the following meaning:

- Bit 15: Auxiliary reader status (1=ok);
- Bit 14: Not used;
- Bit 13: RF status (0=off, 1=on);
- Bit 12: 'Continuous' mode status (0=disabled, 1=enabled);
- Bit 11...8: Not used;
- Bit 7: Dip switch 4 status (1=off);
- Bit 6: Dip switch 3 status (1=off);
- Bit 5: Dip switch 2 status (1=off);
- Bit 4: Dip switch 1 status (1=off);
- Bit 3...0: Not used.

#### 4.10 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** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

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

#### 4.11 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** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

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

#### 4.12 Write Data to EM4305 Transponder

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

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

The 'master' sends the following command:

**SOH <add h> <add l> STX '1' '9' <code 1 h> ... <code n l> ... <code i h> <code i l> ... <code n h> <code n l> ETX <bcc> CR**

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

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

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

#### 4.13 Read ID Code of a EM4305 Transponder

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

The 'master' sends the following command:

**SOH <add h> <add l> STX '1' '8' ETX <bcc> CR**

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

a) if a transponder is present

**SOH <add h> <add l> STX '1' '8' '0' '0' <code 1 h> <code 1 l> ... <code i h> <code i l> ... <code 4 h> <code 4 l> ETX <bcc> CR**

Where:

i	1 ... 4.
<code i h> <code i 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**

b) if a transponder is not present

**SOH <add h> <add l> STX '1' '8' '0' '1' ETX <bcc> CR**

#### 4.14 Write Data to 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)
- **T5557 BLUEBOX MEDIUM**, the code is constituted by 80 bits divided into 20 nibbles
- **T5557 BLUEBOX LARGE**, the code is constituted by 160 bits divided into 40 nibbles

The 'master' sends the following command:

**SOH <add h> <add l> STX '1' 'D' <code 1 h> ... <code n l> ... <code i h> <code i l> ... <code n h> <code n l> ETX <bcc> CR**

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

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

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

#### 4.15 Read ID Code of a T5557 Transponder

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

The 'master' sends the following command:

**SOH <add h> <add l> STX '1' 'C' ETX <bcc> CR**

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

a) if a transponder is present

**SOH <add h> <add l> STX '1' 'C' '0' '0' <code 1 h> <code 1 l> ... <code i h> <code i l> ... <code 8 h> <code 8 l> ETX <bcc> CR**

Where:



i	1 ... 8.
<code i h> <code i 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**

b) if a transponder is not present

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

#### 4.16 Write Data to 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)
- **Q5 BLUEBOX MEDIUM**, the code is constituted by 80 bits divided into 20 nibbles
- **Q5 BLUEBOX LARGE**, the code is constituted by 160 bits divided into 40 nibbles

The 'master' sends the following command:

**SOH <add h> <add l> STX '2' '1' <code 1 h> ... <code n l> ... <code i h> <code i l> ... <code n h> <code n l> ETX <bcc> CR**

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

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

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

### 4.17 Read ID Code of a Q5 Transponder

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

The 'master' sends the following command:

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

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

a) if a transponder is present

**SOH <add h> <add l> STX '2' '0' '0' '0' <code 1 h> <code 1 l> ... <code i h> <code i l> ... <code 5 h> <code 5 l> ETX <bcc> CR**

Where:

i	1 ... 5.
<code i h> <code i 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**

b) if a transponder is not present

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

### 4.18 Write Data to 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)
- **HITAG S BLUEBOX MEDIUM**, the code is constituted by 80 bits divided into 20 nibbles

The 'master' sends the following command:

**SOH <add h> <add l> STX '2' '3' <code 1 h> ... <code n l> ... <code i h> <code i l> ... <code n h> <code n l> ETX <bcc> CR**

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> </ul>
<code i h> <code i l>	i-th byte of the code of the tag. ASCII encoded byte.

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

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

#### 4.19 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 byte.

The 'master' sends the following command:

**SOH <add h> <add l> STX '2' '2' ETX <bcc> CR**

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

if a transponder is present

**SOH <add h> <add l> STX '2' '2' '0' '0' <code 1 h> <code 1 l> ... <code i h> <code i l> ... <code 4 h> <code 4 l> ETX <bcc> CR**

Where:

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

b) if errors I

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

b) if a transponder is not present

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

#### 4.20 Read 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 UID of the transponder.

The 'master' sends the following command:

**SOH <add h> <add l> STX '2' '4' <code 1 h> <code 1 l> ... <code i h> <code i l> ... <code 4 h> <code 4 l> <pag h> <pag l> ETX <bcc> CR**

Where:

i	1 ... 4.
<code i h> <code i l>	i-th byte of the code of the tag. ASCII encoded byte.
<pag h> <pag l>	Page to be read. ASCII encoded byte (0x00 ... 0x3F for HITAG 1 transponders, 0x00 ... 0x07 for HITAG S256 transponders, 0x00 ... 0x3F for HITAG S2048 transponders).

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), it answers with:

if a transponder is present and the page has been successfully readed

**SOH <add h> <add l> STX '2' '4' '0' '0' <data 1 h> <data 1 l> ... <data i h> <data i l> ... <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

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

b) if a transponder is present but errors I

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

c) if a transponder is not present

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

#### 4.21 Write Page of a HITAG 1 / HITAG S Transponder

This command is used to write a data page of the HITAG 1 / HITAG S transponder, constituted by 32 bits (4 bytes). Note that it is necessary to know the UID of the transponder.

The 'master' sends the following command:

**SOH <add h> <add l> STX '2' '5' <code 1 h> <code 1 l> ... <code i h> <code i l> ... <code 4 h> <code 4 l> <pag h> <pag l> <data 1 h> <data 1 l> ... <data j h> <data j l> ... <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

i	1 ... 4.
<code i h> <code i l>	i-th byte of the code of the tag. ASCII encoded byte.
<pag h> <pag l>	Page to be written. ASCII encoded byte (0x00 ... 0x3F for HITAG 1 transponders, 0x00 ... 0x3F for HITAG S2048 transponders).
J	1 ... 4.
<data i h> <data i l>	i-th byte of the tag page. ASCII encoded byte.

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), 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 a transponder is present but errors occurred

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

c) if a transponder is not present

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

#### 4.22 'Reset' 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.

The 'master' sends the following command:

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

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), 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 a transponder is present but errors occurred

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

c) if a transponder is not present

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

#### 4.23 'Login' 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.

The 'master' sends the following command:

**SOH <add h> <add l> STX '4' '1' <pw 1 h> <pw 1 l> ... <pw i h> <pw i l> ... <pw 4 h> <pw 4 l> ETX <bcc> CR**

Where:

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

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), 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 a transponder is present but errors occurred

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

c) if a transponder is not present

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

#### 4.24 'Write Password' 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.

The 'master' sends the following command:

**SOH <add h> <add l> STX '4' '2' <pw a 1 h> <pw a 1 l> ... <pw a i h> <pw a i l> ... <pw a 4 h> <pw a 4 l> <pw n 1 h> <pw n 1 l> ... <pw n i h> <pw n i l> ... <pw n 4 h> <pw n 4 l> ETX <bcc> CR**

Where:

i	1 ... 4.
<pw a i h> <pw a i l>	i-th byte of the password to use. ASCII encoded byte.
<pw n i h> <pw n i l>	i-th byte of the password to set. ASCII encoded byte.

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), 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 a transponder is present but errors occurred

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

c) if a transponder is not present

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

#### 4.25 'Standard Read' 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.

The 'master' sends the following command:

**SOH <add h> <add l> STX '4' '3' ETX <bcc> CR**

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), 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' <data 1 1 h> <data 1 1 l> ... <data i 1 h> <data i 1 l> ... <data 4 1 h> <data 4 1 l> ... <data 1 j h> <data 1 j l> ... <data i j h> <data i j l> ... <data 4 j h> <data 4 j l> ... <data 1 n h> <data 1 n l> ... <data i n h> <data i n l> ... <data 4 n h> <data 4 n l> ETX <bcc> CR**

Where:

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



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

b) if a transponder is present but errors occurred

**SOH <add h> <add l> STX '4' '3' '0' '2' ETX <bcc> CR**

c) if a transponder is not present

**SOH <add h> <add l> STX '4' '3' '0' '1' ETX <bcc> CR**

#### 4.26 'Selective Read' 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.

The 'master' sends the following command:

**SOH <add h> <add l> STX '4' '4' <add u h> <add u l> <add p h> <add p l> ETX <bcc> CR**

Where:

<add p h> <add p l>

Address of the first page to be read. ASCII encoded byte.

<add u h> <add u l>

Address of the last page to be read. ASCII encoded byte.

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), 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' <data 1 1 h> <data 1 1 l> ... <data i 1 h> <data i 1 l> ... <data 4 1 h> <data 4 1 l> ... <data 1 j h> <data 1 j l> ... <data i j h> <data i j l> ... <data 4 j h> <data 4 j l> ... <data 1 n h> <data 1 n l> ... <data i n h> <data i n l> ... <data 4 n h> <data 4 n l> ETX <bcc> CR**

Where:

i

1 ... 4.

j	1 ... n.
n	Number of pages read.
<data i j h> <data i j l>	i-th byte of the j-th page read. ASCII encoded byte.

b) if a transponder is present but errors occurred

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

c) if a transponder is not present

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

#### 4.27 'Write Word' 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.

The 'master' sends the following command:

**SOH <add h> <add l> STX '4' '5' <add w h> <add w l> <data 1 h> <data 1 l> ... <data i h> <data i l> ... <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

<add w h> <add w l>	Address of the page to be written. ASCII encoded byte.
i	1 ... 4.
<data i h> <data i l>	i-th byte of the page to be written. ASCII encoded byte.

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), 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 a transponder is present but errors occurred

**SOH <add h> <add l> STX '4' '5' '0' '2' ETX <bcc> CR**

c) if a transponder is not present

**SOH <add h> <add l> STX '4' '5' '0' '1' ETX <bcc> CR**

#### 4.28 'Write Several Words' 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.

The 'master' sends the following command:

**SOH <add h> <add l> STX '4' '6' <add w h> <add w l> <data 1 1 h> <data 1 1 l> ... <data i 1 h> <data i 1 l> ... <data 4 1 h> <data 4 1 l> ... <data 1 j h> <data 1 j l> ... <data i j h> <data i j l> ... <data 4 j h> <data 4 j l> ... <data 1 n h> <data 1 n l> ... <data i n h> <data i n l> ... <data 4 n h> <data 4 n l> ETX <bcc> CR**

Where:

<add w h> <add w l>	Address of the page to be written. ASCII encoded byte.
i	1 ... 4.
j	1 ... n.
n	Number of pages to be written
<data i j h> <data i j l>	i-th byte of the j-th page to be written. ASCII encoded byte.

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), 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 a transponder is present but errors occurred

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

c) if a transponder is not present

**SOH <add h> <add l> STX '4' '6' '0' '1' ETX <bcc> CR**

#### 4.29 'Read After Write Word' 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.

The 'master' sends the following command:

**SOH <add h> <add l> STX '4' '7' <add w h> <add w l> <data 1 h> <data 1 l> ... <data i h> <data i l> ... <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

<add w h> <add w l>	Address of the page to be written. ASCII encoded byte.
i	1 ... 4.
<data i h> <data i l>	i-th byte of the j-th page to be written. ASCII encoded byte.

If the addressed **BLUEBOX** is not able to execute the command, it answers with:

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

Otherwise (the addressed **BLUEBOX** is able to execute the command), 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' <data 1 h> <data 1 l> ... <data i h> <data i l> ... <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

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

b) if a transponder is present but errors occurred

**SOH <add h> <add l> STX '4' '7' '0' '2' ETX <bcc> CR**

c) if a transponder is not present

**SOH <add h> <add l> STX '4' '7' '0' '1' ETX <bcc> CR**

## 4.30 'Spontaneous' Message

### 4.30.1 RS232

In 'continuous' mode, if the 'spontaneous' feature is set on, the **BLUEBOX** will send the following message on the RS232 serial line every time that it will find a 'new' tag:

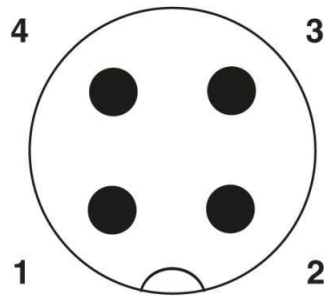
**STX <code 1 h> <code 1 l> ... <code i h> <code i l> ... <code m h> <code m l> ETX <bcc> CR**

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>
<code i h> <code i 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).

## 5 Connections

### 5224L:



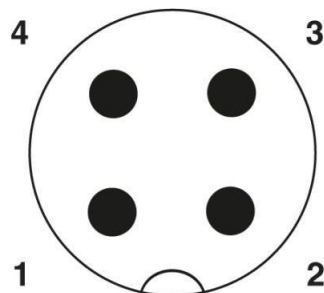
M12 A-coded male

Pin	No	Min	Typical	Max	Description
+PWR	1	18Vdc	24Vdc	36Vdc	Positive DC power supply (+24V ± 10%)
RS232-Tx	2				RS232 Transmit (to host)
-PWR (GND)	3				DC power supply return (GND)
RS232-Rx	4				RS232 Receive (from host)



The shield of the cable is internally connected to GND.

### 5225L:



M12 A-coded male

Pin	No	Min	Typical	Max	Description
+PWR	1	18Vdc	24Vdc	36Vdc	Positive DC power supply (+24V ± 10%)
RS485-RT+	2				RS485 connection (positive)
-PWR (GND)	3				DC power supply return (GND)
RS485-RT-	4				RS485 connection (negative)



The shield of the cable is internally connected to GND.

## 5227L:

Pin	Pair	Wire	Min	Typical	Max	Description
+PWR	Red / black	Red	18Vdc	24Vdc	36Vdc	Positive DC power supply (+24V ± 10%)
-PWR (GND)	Red / black	Black				DC power supply return (GND)
RS232-Tx	White / black	White				RS232 Transmit (to host)
RS232-Rx	White / black	Black				RS232 Receive (from host)



The shield of the cable is internally connected to GND.

## 5228L:

Pin	Pair	Wire	Min	Typical	Max	Description
+PWR	Red / black	Red	18Vdc	24Vdc	36Vdc	Positive DC power supply (+24V ± 10%)
-PWR (GND)	Red / black	Black				DC power supply return (GND)
RS485-RT+	White / black	White				RS485 connection (positive)
RS485-RT-	White / black	Black				RS485 connection (negative)



The shield of the cable is internally connected to GND.

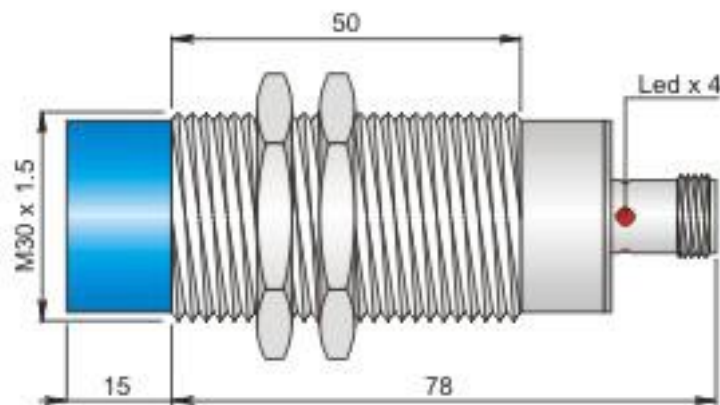


## 6 Installation

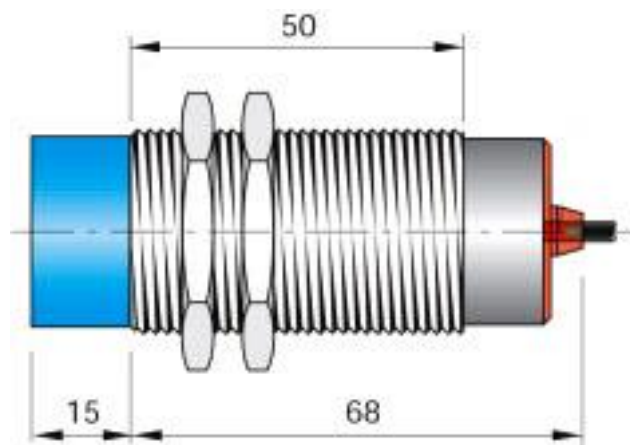
The **BLUEBOX** (items 5224L, 5225L, 5227L, 5228L) is furnished with an integrated RF antenna inside the case.

The read range of an RFID system always depends on various factors like antenna size, transponder size, transponder IC type, orientation between transponder and reader antenna, position of the transponder versus the reader antenna, noise environment, metallic environment, etc. Therefore all data about read ranges can only be typical values measured under laboratory conditions. In real live applications the read range may differ from the data mentioned in the datasheet.

### 5225L, 5226L:



### 5227L, 5228L:








Dimensions in mm.

Fix the **BLUEBOX** using the provided ring nuts and mounting them on non metallic supports (plastic, plexiglas, ...).

Connect the cable wires.

## 7 Status Indications: LEDs

The following table specifies the meaning of LEDs status.

LED	Color	State	Meaning
SYSTEM/TAG	 (red)	On	<ul style="list-style-type: none"> <li>System error.</li> <li>System initialization.</li> <li>System upgrade.</li> </ul>
	 (green)	Blinking	<ul style="list-style-type: none"> <li>Antenna active, no tag detected in 'continuous' mode.</li> </ul>
	 (green)	Slow Blink	<ul style="list-style-type: none"> <li>Antenna not active in 'continuous' mode.</li> <li>'Continuous' mode disabled.</li> </ul>
	 (green)	On	<ul style="list-style-type: none"> <li>Antenna active, tag detected in 'continuous mode.</li> <li>System initialization.</li> </ul>
	 (off)	Off	<ul style="list-style-type: none"> <li>Power supply for the device is missing.</li> <li>Hardware defect.</li> </ul>

### LED state definition

State	Definition
On	The indicator is constantly on
Off	The indicator is constantly off
Blinking	The indicator turns on and off with a frequency of 2 Hz: on for 250 ms, followed by off for 250 ms
Slow Blink	The indicator turns on and off with a frequency of 1 Hz: on for 500 ms, followed by off for 500 ms
Fast Blink	The indicator turns on for 50ms and then off.

## 8 Document Revision History

Date	Revision	Description
24/07/15	1.01	First release.

## A. Supported Transponders

Supported transponders by **BLUEBOX** are:

- 'Continuous' reading of the **UNIQUE** transponder which code is constituted by 40 bits divided into 10 nibbles.
- 'Continuous' reading of the **EM4305 BLUEBOX SHORT** transponder configured as **UNIQUE** transponder (the code is constituted by 40 bits divided into 10 nibbles) with. It has the advantage to be rewritable.
- 'Continuous' reading of the **EM4305 BLUEBOX MEDIUM** transponder configured as 'extended' **UNIQUE** transponder (the code is constituted by 80 bits divided into 20 nibbles). It has the advantage to be rewritable.
- 'Continuous' reading of the **EM4305 BLUEBOX LARGE** transponder configured as 'extended' **UNIQUE** transponder (the code is constituted by 160 bits divided into 40 nibbles). It has the advantage to be rewritable.
- 'On request' reading of the RO code of the **EM4305** transponder, the RO code is constituted by 40 bits divided into 10 nibbles (see **UNIQUE** structure).
- 'On request' writing of all the code of the **EM4305 BLUEBOX SHORT** transponder (the code is constituted by 40 bits divided into 10 nibbles).
- 'On request' writing of all the code of the **EM4305 BLUEBOX MEDIUM** transponder (the code is constituted by 80 bits divided into 20 nibbles).
- 'On request' writing of all the code of the **EM4305 BLUEBOX LARGE**
- 'Continuous' reading of the **T5557 BLUEBOX SHORT** transponder configured as **UNIQUE** transponder (the code is constituted by 40 bits divided into 10 nibbles) with. It has the advantage to be rewritable.
- 'Continuous' reading of the **T5557 BLUEBOX MEDIUM** transponder configured as 'extended' **UNIQUE** transponder (the code is constituted by 80 bits divided into 20 nibbles). It has the advantage to be rewritable.
- 'Continuous' reading of the **T5557 BLUEBOX LARGE** transponder configured as 'extended' **UNIQUE** transponder (the code is constituted by 160 bits divided into 40 nibbles). It has the advantage to be rewritable.
- 'On request' reading of the RO code of the **T5557** transponder, the RO code is constituted by 40 bits divided into 10 nibbles (see **UNIQUE** structure).

- 'On request' writing of all the code of the **T5557 BLUEBOX SHORT** transponder (the code is constituted by 40 bits divided into 10 nibbles).
- 'On request' writing of all the code of the **T5557 BLUEBOX MEDIUM** transponder (the code is constituted by 80 bits divided into 20 nibbles).
- 'On request' writing of all the code of the **T5557 BLUEBOX LARGE**
- 'Continuous' reading of the **Q5 BLUEBOX SHORT** transponder configured as UNIQUE transponder (the code is constituted by 40 bits divided into 10 nibbles) with. It has the advantage to be rewritable.
- 'Continuous' reading of the **Q5 BLUEBOX MEDIUM** transponder configured as 'extended' UNIQUE transponder (the code is constituted by 80 bits divided into 20 nibbles). It has the advantage to be rewritable.
- 'Continuous' reading of the **Q5 BLUEBOX LARGE** transponder configured as 'extended' UNIQUE transponder (the code is constituted by 160 bits divided into 40 nibbles). It has the advantage to be rewritable.
- 'On request' reading of the RO code of the **Q5** transponder, the RO code is constituted by 40 bits divided into 10 nibbles (see UNIQUE structure).
- 'On request' writing of all the code of the **Q5 BLUEBOX SHORT** transponder (the code is constituted by 40 bits divided into 10 nibbles).
- 'On request' writing of all the code of the **Q5 BLUEBOX MEDIUM** transponder (the code is constituted by 80 bits divided into 20 nibbles).
- 'On request' writing of all the code of the **Q5 BLUEBOX LARGE** transponder (the code is constituted by 160 bits divided into 40 nibbles).
- 'Continuous' reading of the **HITAG S256 / S2048 BLUEBOX SHORT** transponder configured as UNIQUE transponder (the code is constituted by 40 bits divided into 10 nibbles). It has the advantage to be rewritable.
- 'Continuous' reading of the **HITAG S256 / S2048 BLUEBOX MEDIUM** transponder configured as 'extended' UNIQUE transponder (the code is constituted by 80 bits divided into 20 nibbles). It has the advantage to be rewritable.
- 'On request' writing of all the code of the **HITAG S256 / S2048 BLUEBOX SHORT** transponder (the code is constituted by 40 bits divided into 10 nibbles).
- 'On request' writing of all the code of the **HITAG S256 / S2048 BLUEBOX MEDIUM** transponder (the code is constituted by 80 bits divided into 20 nibbles).

- 'On request' reading of the RO code (ID) of the **HITAG S32 / S256 / S2048** transponder (the code is constituted by 32 bits).
- 'On request' reading of 1 page (4 bytes) of a **HITAG S2048** 'addressed' transponder (the ID of the transponder must be known).
- 'On request' writing of 1 page (4 bytes) of a **HITAG S2048** 'addressed' transponder (the ID of the transponder must be known).
- **TITAN** transponder management ('reset', 'login', 'write password', 'standard read', 'selective read', 'write word', 'write several words', 'read after write word' commands)

## B. Nibble Coding of Transponder Codes

The nibble coding is explained in the following table. The selected coding is applied when reading the **UNIQUE** transponder and when reading/writing the **BLUEBOX SHORT** transponder (**EM4305, T5557, Q5, HITAG S256 / S2048**).

Nibble	Normal Coding	Reverse Coding	Nibble	Normal Coding	Reverse Coding
0000	'0'	'0'	1000	'8'	'1'
0001	'1'	'8'	1001	'9'	'9'
0010	'2'	'4'	1010	'A'	'5'
0011	'3'	'C'	1011	'B'	'D'
0100	'4'	'2'	1100	'C'	'3'
0101	'5'	'A'	1101	'D'	'B'
0110	'6'	'6'	1110	'E'	'7'
0111	'7'	'E'	1111	'F'	'F'