UHF
RFID System

BLUEBOX INDUSTRIAL UHF LONG-RANGE

RS232 / RS485 / Ethernet

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.

Table of Contents

1 Introduction

The **BLUEBOX INDUSTRIAL UHF LONG RANGE** hereinafter named **BLUEBOX** is an UHF read/write RFID device operating in the 860 MHz to 960 MHz frequency band and suitable for industrial application, it communicates with a 'host' system (typically a PC or a PLC) through a RS232/RS485 serial line (item 5031U) or through an 10-100M Ethernet connection (item 5032U). The **BLUEBOX** acts as a joint through a set of commands between the host system and one or more RFID transponders (or tags) present near the antennas. Up to 4 external antennas can be connected to the **BLUEBOX** allowing good read range (up to 10 m) and directivity in function of the characteristics and positioning of the antennas. The same 'master/slave' protocol is used for the communication between the host system ('master') and the **BLUEBOX** ('slave'), independently of the kind of connection (point to point, multi-drop net, Ethernet). Through these communication channels, it is also possible to configure the functional parameters and to upgrade the firmware. Furthermore the **BLUEBOX** is able to handle two digital outputs (relays) and two opto-isolated digital inputs. The **BLUEBOX** is equipped with spring-cage connection terminal blocks in order to facilitate the electrical wiring, the internal fixing holes of the enclosure allow an easy installation.

2 Technical Specifications

Power supply	10Vdc – 27 Vdc
Power rating	Max 15 W
Operating frequency	860 MHz … 960 MHz
Output power	Max 1W (30 dBm) software programmable
Antenna	Up to 4 external antennas
Reading distance	10 mt[1]
Supported tags	ISO 18000-6B and ISO 18000-6C (EPC Class-1 Generation-2)
Communication interface	RS232 / RS485 Ethernet 10-100M
Signalling	3 Led, Buzzer
Digital inputs	2 opto-isolated inputs Voltage 12 Vdc … 24 Vdc Max current 20 mA Debounce 16 ms
Digital outputs	2 outputs (relay) NO/NC contacts Max 1 A @ 30 Vdc Max 0.5 A @ 125 Vac
Dimensions	120 x 220 x 80 mm
Protection Class	IP 65
Operating temperature	-10°C … +45°C
Connection	Spring-cage terminal blocks (cable section: 0.5 ... 1.5 mm^2)

[1] Reading distance depends on transponder type, antenna and environmental conditions.

3 Operating Features

The **BLUEBOX** can support ISO 18000-6 tags of type B and type C (EPC Class-1 Generation-2); it is possible to select which type of tags will be supported (only type B tags, only type C tags, both) through flags defined in the general parameters.

In 'continuous' mode the **BLUEBOX** is characterized by the coexistence of 2 'parallel' and asynchronous activities: the tag identification (inventory) 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 tags or that is empty indicating the absence of tags. Due to synchronization and filtering reasons, the buffer is handled for each identified tag by a parameter defined as 'hold time' (same as 'filter time' defined below, to be set in the range of 0 … 99 seconds or 0 ... 99 minutes, default value 10 seconds) and allows to extend 'artificially' the presence of the tag after it leaves the antenna's influence area; this behaviour is observable looking at the yellow led status that is 'on' indicating the presence of tags and also through the activation of the relay nr 1 (if its 'automatic' management is enabled by the flag defined in the general parameters). Through the command 'data request' it is possible to get the data contained in the buffer (tag/s UID and respective reading antenna); the indication of the reading antenna can be enabled/disabled through a flag defined in the general parameters.

The **BLUEBOX** handles also a 31 elements FIFO queue which is combined with the 'filter time' general parameter (to be set in a range of 0 … 99 seconds or 0 ... 99 minutes, default value 10 seconds) that prevents the queue saturation in case of a tag 'continuous' presence. When a tag is identified, the **BLUEBOX** verifies if it belongs to the list of read tags. If the tag do not belong to the list (it is defined as 'new'), its code will be inserted in the queue, a filter time assigned to the tag will be started and the buzzer will be activated for 0.5 seconds (if its 'automatic' management is enabled by the flag defined in the general parameters). Otherwise (the tag belong to the list of read tags), the **BLUEBOX** verifies if the relative filter time is expired. In this case (the filter time is expired), the tag is defined as 'new' and will be processed as described above, otherwise only the relative 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 (tag UID and respective reading antenna) and unload it; the indication of the reading antenna can be enabled/disabled through a flag defined in the general parameters.

In 'continuous' mode the **BLUEBOX** can be configured to obtain the behaviour of a 'spontaneous' reader that will send a message on the RS232 serial line and on the Ethernet channel (if available) every time that a 'new' transponder

is identified. This feature is enabled (on) / disabled (off) by the switch 2 of the dip switch SW1. It could be also configured to obtain the behaviour of a 'spontaneous' reader on the RS485 serial line, in this case the message sent need an 'ack' as confirmation of the reception success condition to avoid and correct collisions on RS485 bus. This feature is enabled (on) / disabled (off) by the switch 2 of the dip switch SW1.

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.

Up to 4 RF antennas can be connected to the **BLUEBOX**; in the of more than one antenna enabled, the system will activate alternatively only one antenna at a time for the duration specified for each antenna in the RF antennas parameters.

List of configurable general parameters:

Parameters	Range / Choice	Default
Network address	000 … 255	255
Baud rate	1200, 2400, 4800, 9600, 19200, 38400	19200
Data bits	7, 8	8
Stop bits	1, 2	1
Parity	None, even, odd	None
Filter time	0 ... 99 seconds / 0 ... 99 minutes	10 sec
Tag type	B and/or C (EPC Class-1 Generation-2)	B and C
Buzzer	Disabled, enabled	Enabled
Output relay 1	Disabled, enabled	Disabled
Reading antenna information	Disabled, enabled	Disabled

Transponder type information	Disabled, enabled	Disabled
'Continuous mode'	Disabled, enabled	Enabled

List of configurable RF parameters:

Parameter	Range / Choice	Default
Antenna 1 power	0 ... 30 dBm	30 dBm
Antenna 1 activation time	1 … 65535 ms	1000 ms
Antenna 2 power	0 ... 30 dBm	30 dBm
Antenna 2 activation time	1 … 65535 ms	1000 ms
Antenna 3 power	0 ... 30 dBm	30 dBm
Antenna 3 activation time	1 … 65535 ms	1000 ms
Antenna 4 power	0 ... 30 dBm	30 dBm
Antenna 4 activation time	1 … 65535 ms	1000 ms

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	Character 01h (0x01)
STX	Character 02h (0x02)
ETX	Character 03h (0x03)
EOT	Character 04h (0x04)
ENQ	Character 05h (0x05)
ACK	Character 06h (0x06)
NAK	Character 15h (0x15)
SYN	Character 16h (0x16)
CR	Character 0Dh (0x0D)
'0'...'9'	Character 30h …39h (0x30 … 0x39)
'A'…'F'	Character 41h …46h (0x41 … 0x46)
<..>	Character 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 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> '0' '0' <type h> <type l> <filt h> <filt l> <flag h> <flag l> ETX <bcc> CR

Dove:

<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: • '0' -> 1200 bps; • '1' -> 2400 bps; • '2' -> 4800 bps; • '3' -> 9600 bps; • '4' -> 19200 bps • '5' -> 38400 bps.
<bit>	RS232/RS485 communication interface data bits. ASCII character: • '7' -> 7 bits; • '8' -> 8 bits.
<stop>	RS232/RS485 communication interface stop bits. ASCII character: • '1' -> 1 bit;

	• '2' -> 2 bits.
<par>	RS232/RS485 communication interface parity. ASCII character: • '0' -> None; • '1' -> Even; • '2' -> Odd.
<type h> <type l>	Tag type. ASCII encoded byte whose bits are dedicated to disable (0 value) or enable (1 value) functions: • Bit 7 ... 2: not used; • Bit 1: C type; • Bit 0: B type.
<filt h> <filt l>	Filter time. ASCII encoded byte: • Decimal 0 … 99 for time in seconds (0 … 99 seconds); • Decimal 100 … 199 for time in minutes (0 … 99 minutes).
<flag h> <flag l>	Flags. ASCII encoded byte whose bits are dedicated to disable (0 value) or enable (1 value) functions: • Bit 7: automatic buzzer management; • Bit 6: automatic relay 1 management; • Bit 5: reading antenna indication; • Bit 4: transponder type indication; • Bit 3 … bit 1: not used; • Bit 0: to disable 'continuous' mode.

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

Note: after the command execution, the **BLUEBOX** resets itself to apply the new parameters.

4.2 RF Parameters Programming

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

The 'master' sends the following command:

SOH <add h> <add l> STX '3' 'B' <pwr 1 hh> <pwr 1 hl> <pwr 1 lh> <pwr 1 ll> <dur 1 hh> <dur 1 hl> <dur 1 lh> dur 1 ll> ... <pwr i hh> <pwr i hl> <pwr i lh> <pwr i ll> <dur i hh> <dur i hl> <dur i lh> <dur i ll> ... <pwr 4 hh> <pwr 4 hl> <pwr 4 lh> <pwr 4 ll> <dur4 hh> <dur4 hl> <dur4 lh> dur4 ll> ETX <bcc> CR

Where:

i	1 … 4
<pwr i hh> <pwr i hl> <pwr i lh> <pwr i ll>	i-th antenna power in 0.1 dBm. ASCII encoded word (0 dBm … 30 dBm)
<dur i hh> <dur i hl> <dur i lh> <dur i ll>	i-th antenna activation time in ms. ASCII encoded word (1 ms … 65535 ms)

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

Note: after the command execution, the **BLUEBOX** resets itself to apply the new parameters.

4.3 Default Parameters Programming

This command is used to set the default values of the communication, general and RF parameters of the **BLUEBOX**. If the Ethernet channel is available, the default IP address 192.168.004.200 and port 3000 will be set.

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

Note: after the command execution, the **BLUEBOX** resets itself to apply the new parameters.

4.4 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> '0' '0' <type h> <type l> <filt d> <filt u> <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: • '0' -> 1200 bps; • '1' -> 2400 bps; • '2' -> 4800 bps;

	• '3' -> 9600 bps; • '4' -> 19200 bps • '5' -> 38400 bps.
<bit>	RS232/RS485 communication interface data bits. ASCII character: • '7' -> 7 bits; • '8' -> 8 bits.
<stop>	RS232/RS485 communication interface stop bits. ASCII character: • '1' -> 1 bit; • '2' -> 2 bits.
<par>	RS232/RS485 communication interface parity. ASCII character: • '0' -> None; • '1' -> Even; • '2' -> Odd.
<type h> <type l>	Tag type. ASCII encoded byte whose bits are dedicated to disable (0 value) or enable (1 value) functions: • Bit 7 ... 2: not used; • Bit 1: C type; • Bit 0: B type.
<filt h> <filt l>	Filter time. ASCII encoded byte: • Decimal 0 … 99 for time in seconds (0 … 99 seconds); • Decimal 100 … 199 for time in minutes (0 … 99 minutes).
<flag h> <flag l>	Flags. ASCII encoded byte whose bits are dedicated to disable (0 value) or enable (1 value) functions: • Bit 7: automatic buzzer management; • Bit 6: automatic relay 1 management; • Bit 5: reading antenna indication; • Bit 4: transponder type indication; • Bit 3 … bit 1: not used; • Bit 0: to disable 'continuous' mode.

4.5 RF Parameters Reading

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

The 'master' sends the following command:

SOH <add h> <add l> STX '3' '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 '3' 'A' <pwr 1 hh> <pwr 1 hl> <pwr 1 lh> <pwr 1 ll> <dur 1 hh> <dur 1 hl> <dur 1 lh> dur 1 ll> … <pwr i hh> <pwr i hl> <pwr i lh> <pwr i ll> <dur i hh> <dur i hl> <dur i lh> <dur i ll> … <pwr 4 hh> <pwr 4 hl> <pwr 4 lh> <pwr 4 ll> <dur4 hh> <dur4 hl> <dur4 lh> dur4 ll> ETX <bcc> CR

Where:

i	1 … 4
<pwr i hh> <pwr i hl> <pwr i lh> <pwr i ll>	i-th antenna power in 0.1 dBm. ASCII encoded word (0 dBm … 30 dBm)
<dur i hh> <dur i hl> <dur i lh> <dur i ll>	i-th antenna activation time in ms. ASCII encoded word (1 ms … 65535 ms)

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 'BRIDGE_UHF_1.01l' indicates that this is this is a BLUEBOX **BRIDGE** in **UHF** configuration (**U**ltra **H**igh **F**requency 860 - 960 MHz) with firmware version **1.01l long range**.

This command could also be used to read the firmware version of the RFID front-end:

SOH <add h> <add l> STX '3' '4' '0' '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** 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 63 h> <vf 63 l> <vf 64 h> <vf 64 l> ETX <bcc> CR

Where:

<vf 01 h> <vf 01 l>	ASCII coding of the byte 1 of the string.
…	…
<vf 64 h> <vf 64 l>	ASCII coding of the byte 64 of the string.

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:

a) flag for reading antenna indication and transponder type indication disabled:

SOH <add h> <add l> STX <UID 1 1 h> <UID 1 1 l> … <UID 1 i h> <UID 1 i l> … <UID 1 m h> <UID 1 m l> '-' … '-' <UID j 1 h> <UID j 1 l> … <UID j i h> <UID j i l> … <UID j m h> <UID j m l> '-' … '-' <UID n 1 h> <UID n 1 l> … <UID n i h> <UID n i l> … <UID n m h> <UID n m l> ETX <bcc> CR

b) flag for reading antenna enabled and for transponder type disabled:

SOH <add h> <add l> STX <UID 1 1 h> <UID 1 1 l> … <UID 1 i h> <UID 1 i l> … <UID 1 m h> <UID 1 m l> '0' <ant 1> '-' … '-' <UID j 1 h> <UID j 1 l> … <UID j i h> <UID j i l> … <UID j m h> <UID j m l> '0' <ant j> '-' … '-' <UID n 1 h> <UID n 1 l> … <UID n i h> <UID n i l> … <UID n m h> <UID n m l> '0' <ant n> ETX <bcc> CR

c) flag for reading antenna disabled and for transponder type enabled:

SOH <add h> <add l> STX <tag 1 h> <tag 1 l><UID 1 1 h> <UID 1 1 l> … <UID 1 i h> <UID 1 i l> … <UID 1 m h> <UID 1 m l> '-' … '-' <tag j h> <tag j l> <UID j 1 h> <UID j 1 l> … <UID j i h> <UID j i l> … <UID j m h> <UID j m l> '-' … '-' <tag n h> <tag n l> <UID n 1 h> <UID n 1 l> … <UID n i h> <UID n i l> … <UID n m h> <UID n m l> ETX <bcc> CR

d) flag for reading antenna indication and transponder type indication enabled:

SOH <add h> <add l> STX <tag 1 h> <tag 1 l> <UID 1 1 h> <UID 1 1 l> … <UID 1 i h> <UID 1 i l> … <UID 1 m h> <UID 1 m l> '0' <ant 1> '-' … '-' <tag j h> <tag j l> <UID j 1 h> <UID j 1 l> … <UID j i h> <UID j i l> … <UID j m h> <UID j m l> '0' <ant j> '-' … '-' <tag n h> <tag n l> <UID n 1 h> <UID n 1 l> … <UID n i h> <UID n i l> … <UID n m h> <UID n m l> '0' <ant n> ETX <bcc> CR

Where:

i	1 … m.
m	UID length, 8 byte for ISO 18000-6B tags and max 20 byte for ISO 18000-6C (EPC Class-1 Generation-2) tags.
j	1 … n.
n	Number of identified tags.
<tag j h> <tag j l>	Transponder type for the j-th identified tag. ASCII encoded byte: • 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC Class-1Generation-2).
<UID j i h> <UID j i l>	i-th byte of the UID of the j-th identified tag. ASCII encoded byte..
'-'	Separator 0x5F.
<ant j>	Reading antenna for the j-th identified tag. ASCII character: • '1' -> Antenna 1; • '2' -> Antenna 2; • '3' -> Antenna 3; • '4' -> Antenna 4.

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:

a) flag for reading antenna indication and transponder type indication disabled:

SOH <add h> <add l> STX <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> ETX <bcc> CR

b) flag for reading antenna enabled and for transponder type disabled:

SOH <add h> <add l> STX <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> '0' <ant> ETX <bcc> CR

c) flag for reading antenna disabled and for transponder type enabled:

SOH <add h> <add l> STX <tag h> <tag l> <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> ETX <bcc> CR

d) flag for reading antenna indication and transponder type indication enabled:

SOH <add h> <add l> STX <tag h> <tag l> <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> '0' <ant> ETX <bcc> CR

Where:

i	1 … m.
m	UID length, 8 byte for ISO 18000-6B tags and max 20 byte for ISO 18000-6C (EPC Class-1 Generation-2) tags.

<tag h> <tag l>	Transponder type for the j-th identified tag. ASCII encoded byte: • 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC Class-1Generation-2).
<UID i h> <UID i l>	i-th byte of the UID of the identified tag. ASCII encoded byte.
<ant j>	Reading antenna for the j-th identified tag. ASCII character: • '1' -> Antenna 1; • '2' -> Antenna 2; • '3' -> Antenna 3; • '4' -> Antenna 4.

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 Digital Output (Relay) Activation

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

The 'master' sends the following command:

SOH <add h> <add l> STX '3' '7' <can d> <can u> <dur h> <dur l> ETX <bcc> CR

Where:

<can d> <can u>	Relay to activate. ASCII encoded byte: • 0x01 -> Relay 1; • 0x02 -> Relay 2.

<dur h> <dur l>	Activation time. ASCII encoded byte: • 0x01 … 0x63 (1 ... 99 seconds -> in case of 'impulsive' relay activation; • 0x81 -> 'Continuous' activation; • 0x80 -> Deactivation.

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

4.10 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 (RFID frontend) detected; • Bit 14: Not used; • Bit 13: RF status (0=off, 1=on);

	• Bit 12: 'Continuous' mode status (0=disabled, 1=enabled); • Bit 11…10: Not used; • Bit 9: Relay 2 status (1=activated); • Bit 8: Relay 1 status (1=activated); • 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…2: Not used; • Bit 1: Input 2 status (1=activated); • Bit 0: Input 1 status (1=activated).

4.11 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

4.12 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

4.13 Inventory of Type B Tags

This command is used to get the list of the UID (composed of 8 bytes) of the identified type B 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' '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 reader is able to execute the command), it answers with:

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

SOH <add h> <add l> STX '1' '0' '0' '0' <UID 1 1 h> <UID 1 1 l> … <UID 1 i h> <UID 1 i l> … <UID 1 8 h> <UID 1 8 l> … <UID j 1 h> <UID j 1 l>… <UID j i h> <UID j i l>… <UID j 8 h> <UID j 8 l> … <UID n 1 h> <UID n 1 l> … <UID n i h> <UID n i l> … <UID n 8 h> <UID n 8 l> ETX <bcc> CR

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

SOH <add h> <add l> STX '1' '0' '0' '0' <UID 1 1 h> <UID 1 1 l> … <UID 1 i h> <UID 1 i l> … <UID 1 8 h> <UID 1 8 l> '0' <ant 1> … <UID j 1 h> <UID j 1 l>… <UID j i h> <UID j i l>… <UID j 8 h> <UID j 8 l> '0' <ant j> … <UID n 1 h> <UID n 1 l> … <UID n i h> <UID n i l> … <UID n 8 h> <UID n 8 l> '0' <ant n> ETX <bcc> CR

Where:

i	1 … 8.
j	1 … n.
n	Number of identified tags.
<UID j i h> <UID j i l>	i-th byte of the UID of the j-th identified tag. ASCII encoded byte.
<ant j>	Reading antenna for the j-th identified tag. ASCII character: • '1' -> Antenna 1; • '2' -> Antenna 2; • '3' -> Antenna 3; • '4' -> Antenna 4.

b) if no tag is present

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

The command can also be used imposing selection criteria, in this case the command assume the following form:

SOH <add h> <add l> STX '1' '0' <cmd h> <cmd l> <sadd h> <sadd l> <msk h> <msk l> <data 1 h> <data 1 l> … <data i h> <data i l> … <data 8 h> <data 8 l> ETX <bcc> CR

Where:

<cmd h> <cmd l>	Selection command code. ASCII encoder byte: • '0' '0' (0x00): GROUP_SELECT_EQ • '0' '1' (0x01): GROUP_SELECT_NE • '0' '2' (0x02): GROUP_SELECT_GT • '0' '3' (0x03): GROUP_SELECT_LT
<sadd h> <sadd l>	Address of the 1st byte of the tags to be compared with the bytes of the data field according with the mask field. ASCII encoded byte.
<msk h> <msk l>	Mask to be used in the comparison. ASCII encoded byte whose bits have the following meaning: • Bit 7: Enables (if set to '1') /disables (if set to '0') the comparison of byte 1 of the following data

	field with the 1st specified byte of the tags; • Bit 6: Enables (if set to '1') /disables (if set to '0') the comparison of byte 2 of the following data field with the 2nd specified byte of the tags; • ... • Bit 0: Enables (if set to '1') /disables (if set to '0') the comparison of byte 8 of the following data field with the 8th specified byte of the tags.
i	1 … 8
<data i h> <data i l>	i-th byte of data to be compared with tags data. ASCII encoded byte.

Note that when the short form command is used the following values are assumed: 0x01 for cmd, 0x00 for sadd, 0xC0 for msk, 0x00 ... 0x00 for data.

4.14 Read Data from Type B Tags

This command is used to get data blocks (data block → 8 consecutive bytes) of a known (UID) type B tag.

The 'master' sends the following command:

SOH <add h<add l> STX '1' '1' <UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 8 h> <UID 8 l> <sadd h> <sadd l> <nblk h> <nblk l> ETX <bcc> CR

Where:

i	1 … 8.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
<sadd h> <sadd l>	Address of the 1st byte of the 1st block of memory to read. ASCII encoded byte.
<nblk h> <nblk l>	Number of blocks of memory to be read (1 … 8). 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 reader is able to execute the command), it answers with:

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

SOH <add h> <add l> STX '1' '1' '0' '0' <data 1 1 h> <data 1 1 l> … <data i 1 h> <data i 1 l> … <data 8 1 h> <data 8 1 l> … <data 1 j h> <data 1 j l> … <data i j h> <data i j l> … <data 8 j h> <data 8 j l> … <data 1 n h> <data 1 n l> … <data i n h> <data i n l> … <data 8 n h> <data 8 n l> ETX <bcc> CR

Where:

i	1 … 8.
j	1 … n.
n	Number of blocks of memory read.
<data i j h> <data i j l>	i-th byte of the j-th block of memory read from tag. ASCII encoded byte.

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

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

c) if the addressed tag is not present

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

4.15 Write Data into Type B Tags

This command is used to write data on a known (UID) type B tag.

The 'master' sends the following command:

SOH <add h<add l> STX '1' '2' <UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 8 h> <UID 8 l> <sadd h> <sadd l> <nbr h> <nbr l> <data 1 h> <data 1 l> … <data j h> <data j l> … <data n h> <data n l> ETX <bcc> CR

Where:

i	1 … 8.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoed byte.
<sadd h> <sadd l>	Address of the 1st byte of the 1st block of memory to be written. ASCII encoded byte.
<nblk h> <nblk l>	Number of bytes to be written (1 … 32). ASCII encoded byte.
j	1 … n.
n	Number of bytes to be written, range 1 … 32.
<data j h> <data j l>	j-th byte 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 the addressed tag is present and the data bytes have been successfully written

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

b) if the addressed tag is present but errors occurred

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

c) if the addressed tag is not present

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

4.16 Inventory of Type C Tags

This command is used to get the list of the UID (variable size, max 20 bytes) of the identified type C 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** is not able to execute the command, it answers with:

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

Otherwise (the reader is able to execute the command), it answers with:

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

SOH <add h> <add l> STX '1' '8' '0' '0' <UID 1 1 h> <UID 1 1 l> … <UID 1 i h> <UID 1 i l> … <UID 1 16 h> <UID 1 16 l> … <UID j 1 h> <UID j 1 l>… <UID j i h> <UID j i l>… <UID j 16 h> <UID j 16 l> … <UID n 1 h> <UID n 1 l> … <UID n i h> <UID n i l> … <UID n 16 h> <UID n 16 l> ETX <bcc> CR

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

SOH <add h> <add l> STX '1' '8' '0' '0' <UID 1 1 h> <UID 1 1 l> … <UID 1 i h> <UID 1 i l> … <UID 1 16 h> <UID 1 16 l> '0' <ant 1> … <UID j 1 h> <UID j 1 l>… <UID j i h> <UID j i l>… <UID j 16 h> <UID j 16 l> '0' <ant j> … <UID n 1 h> <UID n 1 l> … <UID n i h> <UID n i l> … <UID n 16 h> <UID n 16 l> '0' <ant n> ETX <bcc> CR

Where:

i	1 … m.
m	UID length, max 20 bytes.
j	1 … n.
n	Number of identified tags.
<UID j i h> <UID j i l>	i-th byte of the UID of the j-th identified tag. ASCII encoded byte.
<ant j>	Reading antenna for the j-th identified tag. ASCII character: • '1' -> Antenna 1; • '2' -> Antenna 2; • '3' -> Antenna 3; • '4' -> Antenna 4.

b) if some error is occurred during the transaction

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

b) if no tag is present

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

4.17 Read Data from Type C Tags

This command is used to get data blocks (data block → 2 consecutive bytes) of a known (UID) type C tag.

The 'master' sends the following command:

SOH <add h<add l> STX '1' '9' <UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 16 h> <UID 16 l> <pwd 1 h> <pwd 1 l> … <pwd j h> <pwd j l> … <pwd 4 h> <pwd 4 l> '0' <bank> <sadd 1 h> <sadd 1 l> <sadd j h> <sadd j l> <sadd 4 h> <sadd 4 l> <nblk h> <nblk l> ETX <bcc> CR

Where:

i	1 … m.
m	UID length, max 20 bytes.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoed byte.
j	1 … 4.
<pwd j h> <pwd j 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: • '0': Reserved. • '1': TID. • '2': EPC. • '3': User.
<sadd j h> <sadd j 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 block to be read. ASCII encoded byte (1 … 4).

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

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

Otherwise (the reader is able to execute the command), it answers with:

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

SOH <add h> <add l> STX '1' '9' '0' '0' <data 1 1 h> <data 1 1 l> … <data i 1 h> <data i 1 l> … <data 2 1 h> <data 2 1 l> … <data 1 j h> <data 1 j l> … <data i j h> <data i j l> … <data 2 j h> <data 2 j l> … <data 1 n h> <data 1 n l> … <data i n h> <data i n l> … <data 2 n h> <data 2 n l> ETX <bcc> CR

Where:

i	1 … 2.
j	1 … n.
n	Number of blocks read.
<data i j h> <data i j l>	i-th byte of the j-th block of memory read from tag. ASCII encoded byte.

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

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

c) if the addressed tag is not present

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

4.18 Write Data into Type C Tags

This command is used to write data on a known (UID) type C tag.

The 'master' sends the following command:

SOH <add h<add l> STX '1' 'A' <UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 16 h> <UID 16 l> <pwd 1 h> <pwd 1 l> … <pwd j h>

<pwd j l> … <pwd 4 h> <pwd 4 l> '0' <bank> <sadd 1 h> <sadd 1 l> <sadd j h> <sadd j l> <sadd 4 h> <sadd 4 l> <data 1 1 h> <data 1 1 l> … <data w 1 h> <data w 1 l> … <data 2 1 h> <data 2 1 l> … <data 1 k h> <data 1 k l> … <data w k h> <data w k l> … <data 2 k h> <data 2 k l> … <data 1 n h> <data 1 n l> … <data w n h> <data w n l> … <data 2 n h> <data 2 n l> ETX <bcc> CR

Where:

i	1 … m.
m	UID length, max 20 bytes.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
j	1 … 4.
<pwd j h> <pwd j 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 written. ASCII character: • '0': Reserved. • '1': TID. • '2': EPC. • '3': User.
<sadd j h> <sadd j l>	j-th byte of the address of the 1st byte of the 1st block to be written. ASCII encoded byte.
w	1 … 2.
k	1 … n.
n	Number of block sto be written, range 1 … 4.
<data w k l> <data w k l>	k-th byte of the w-th block 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 the addressed tag is present and the data bytes have been successfully written

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

b) if the addressed tag is present but errors occurred

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

c) if the addressed tag is not present

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

4.19 Lock Type C Tags

This command is used to lock individual password and/or individual memory banks on a known (UID) type C tag.

The 'master' sends the following command:

SOH <add h<add l> STX '1' 'B' <UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 16 h> <UID 16 l> <pwd 1 h> <pwd 1 l> … <pwd j h> <pwd j l> … <pwd 4 h> <pwd 4 l> '0' <kil> '0' <acc> '0' <EPC> '0' <TID> '0' <user> ETX <bcc> CR

Where:

i	1 … m.
m	UID length, max 20 bytes.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
j	1 … 4.
<pwd j h> <pwd j 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: • '0': Accessible from all states; • '1': Permanently accessible from all states and may never be locked; • '2': Accessible only from the secured state; • '3': Not accessible from any state;

	• '4': No change.
<acc>	Access password lock property. ASCII character: • '0': Accessible from all states; • '1': Permanently accessible from all states and may never be locked; • '2': Accessible only from the secured state; • '3': Not accessible from any state; • '4': No change.
<EPC>	EPC memory bank lock property. ASCII character: • '0': Writable from all states; • '1': Permanently writable from all states and may never be locked; • '2': Writable only from the secured state; • '3': Not writable from any state; • '4': No change.
<TID>	TID memory bank lock property. ASCII character: • '0': Writable from all states; • '1': Permanently writable from all states and may never be locked; • '2': Writable only from the secured state; • '3': Not writable from any state; • '4': No change.
<user>	User memory bank lock property. ASCII character: • '0': Writable from all states; • '1': Permanently writable from all states and may never be locked; • '2': Writable only from the secured state; • '3': Not writable from any state; • '4': No change.

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 the addressed tag is present and it has been successfully locked

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

b) if the addressed tag is present but errors occurred

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

c) if the addressed tag is not present

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

4.20 Kill Type C Tags

This command is used to kill a known (UID) type C tag.

The 'master' sends the following command:

SOH <add h<add l> STX '1' 'C' <UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 16 h> <UID 16 l> <pwd 1 h> <pwd 1 l> … <pwd j h> <pwd j l> … <pwd 4 h> <pwd 4 l> '0' ETX <bcc> CR

Dove:

i	1 … m.
m	UID length, max 20 bytes.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
j	1 … 4.
<pwd j h> <pwd j l>	j-th byte of the kill password. 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 the addressed tag is present and the tag is been successfully killed

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

b) if the addressed tag is present but errors occurred

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

c) if the addressed tag is not present

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

4.21 'Spontaneous' Message

4.21.1 RS232 Serial Line or Ethernet

In 'continuous' mode, if the 'spontaneous' feature is set on (see dip switch settings in the next paragraph), the **BLUEBOX** will send the following message on the RS232 serial line and on the Ethernet channel (if available) every time that it will find a 'new' tag:

a) flag for reading antenna indication and transponder type indication disabled:

STX <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> ETX <bcc> CR

b) flag for reading antenna enabled and for transponder type disabled:

STX <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> '0' <ant> ETX <bcc> CR

c) flag for reading antenna disabled and for transponder type enabled:

STX <tag h> <tag l> <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> ETX <bcc> CR

d) flag for reading antenna indication and transponder type indication enabled:

STX <tag h> <tag l> <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> '0' <ant> ETX <bcc> CR

Where:

i	1 … m.
m	UID length, max 20 bytes.
<tag h> <tag l>	Transponder type. ASCII encoded byte:

	• 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC C1G2).
<UID i h> <UID i l>	i-th byte of the UID of the identified tag. ASCII encoded byte.
<ant>	Reading antenna. ASCII character: • '1' -> Antenna 1; • '2' -> Antenna 2; • '3' -> Antenna 3; • '4' -> Antenna 4.
<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).

4.21.2 RS485 Serial Line

In 'continuous' mode, if the 'spontaneous' feature is set on (see dip switch settings in the next paragraph), the **BLUEBOX** will send the following message on the RS485 serial line every time that it will find a 'new' tag:

a) flag for reading antenna indication and transponder type indication disabled:

SOH <add h> <add l> STX <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> ETX <bcc> CR

b) flag for reading antenna enabled and for transponder type disabled:

SOH <add h> <add l> STX <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> '0' <ant> ETX <bcc> CR

c) flag for reading antenna disabled and for transponder type enabled:

SOH <add h> <add l> STX <tag h> <tag l> <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> ETX <bcc> CR

d) flag for reading antenna indication and transponder type indication enabled:

SOH <add h> <add l> STX <tag h> <tag l> <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID m h> <UID m l> '0' <ant> ETX <bcc> CR

Where:

i	1 … m.
m	UID length, max 20 bytes.
<tag h> <tag l>	Transponder type. ASCII encoded byte: • 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC C1G2).
<UID i h> <UID i l>	i-th byte of the UID of the identified tag. ASCII encoded byte.
<ant>	Reading antenna. ASCII character: • '1' -> Antenna 1; • '2' -> Antenna 2; • '3' -> Antenna 3; • '4' -> Antenna 4.

To correct any collisions on the RS485 bus, the **BLUEBOX** repeats the message (up to 10 times) until the 'master' does not respond with:

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

The timeout in receiving reply from the 'master' is set at 250 ms, repetition of the message is not immediate but delayed by a pseudo-random time, calculated using the network address as seed, between 1 ms and 5000 ms (5 sec.). This will avoid repeated collisions on the RS485 bus.

4.22 Reset

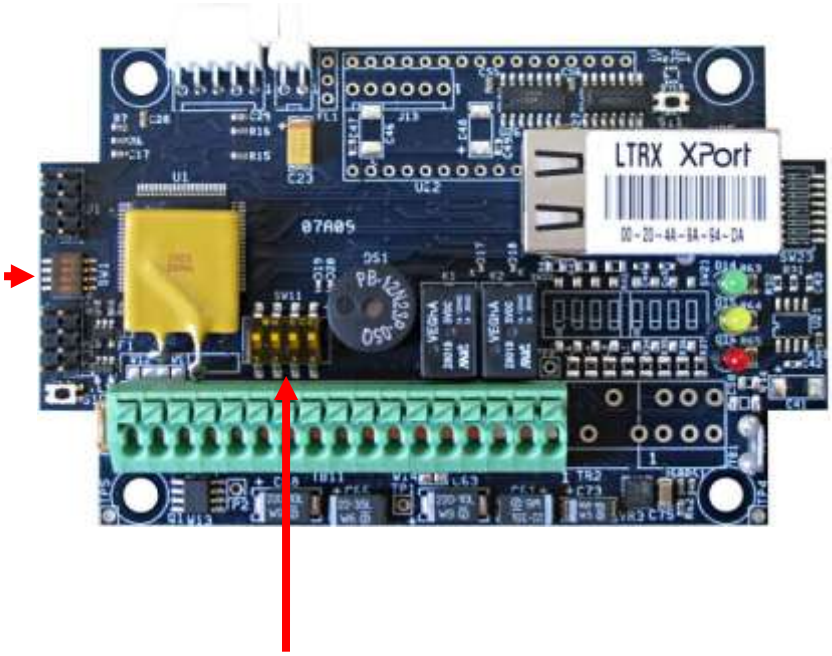
The 'master' sends the following command:

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

The **BLUEBOX** immediately restarts itself.

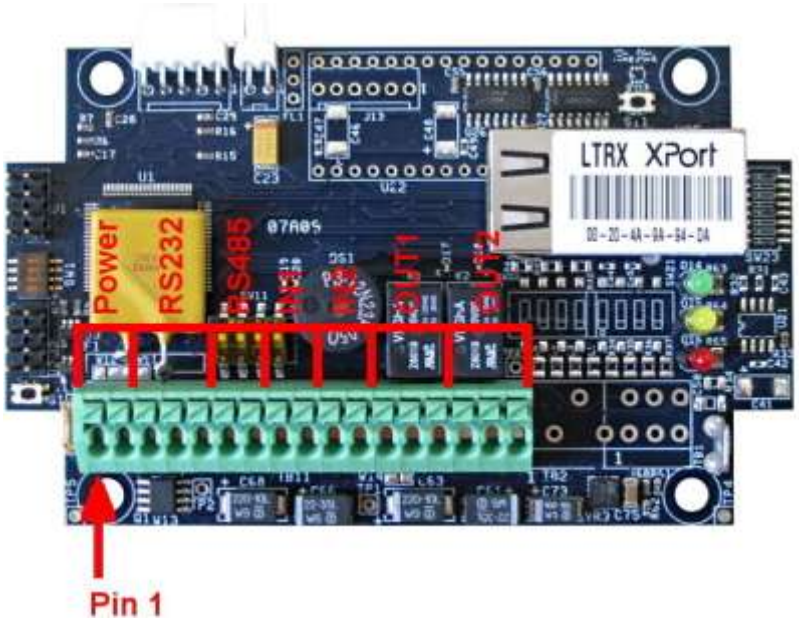
5 Hardware Settings: Dip Switch

SW1	
Dip 1	On: force 255, 19200, 8, n, 1.
Dip 2	On: enables 'spontaneous' mode on RS232 serial line or Ethernet.
Dip 3	On: enables 'spontaneous' mode on RS485 serial line.
Dip 4	Not used.



SW11	
Dip 1	On: RS485 fail-safe resistor connected to +5V.
Dip 2	On: RS485 fail-safe resistor connected to gnd.
Dip 3	On: RS485 120Ω line termination resistor connected.
Dip 4	Not used.

6 Connections

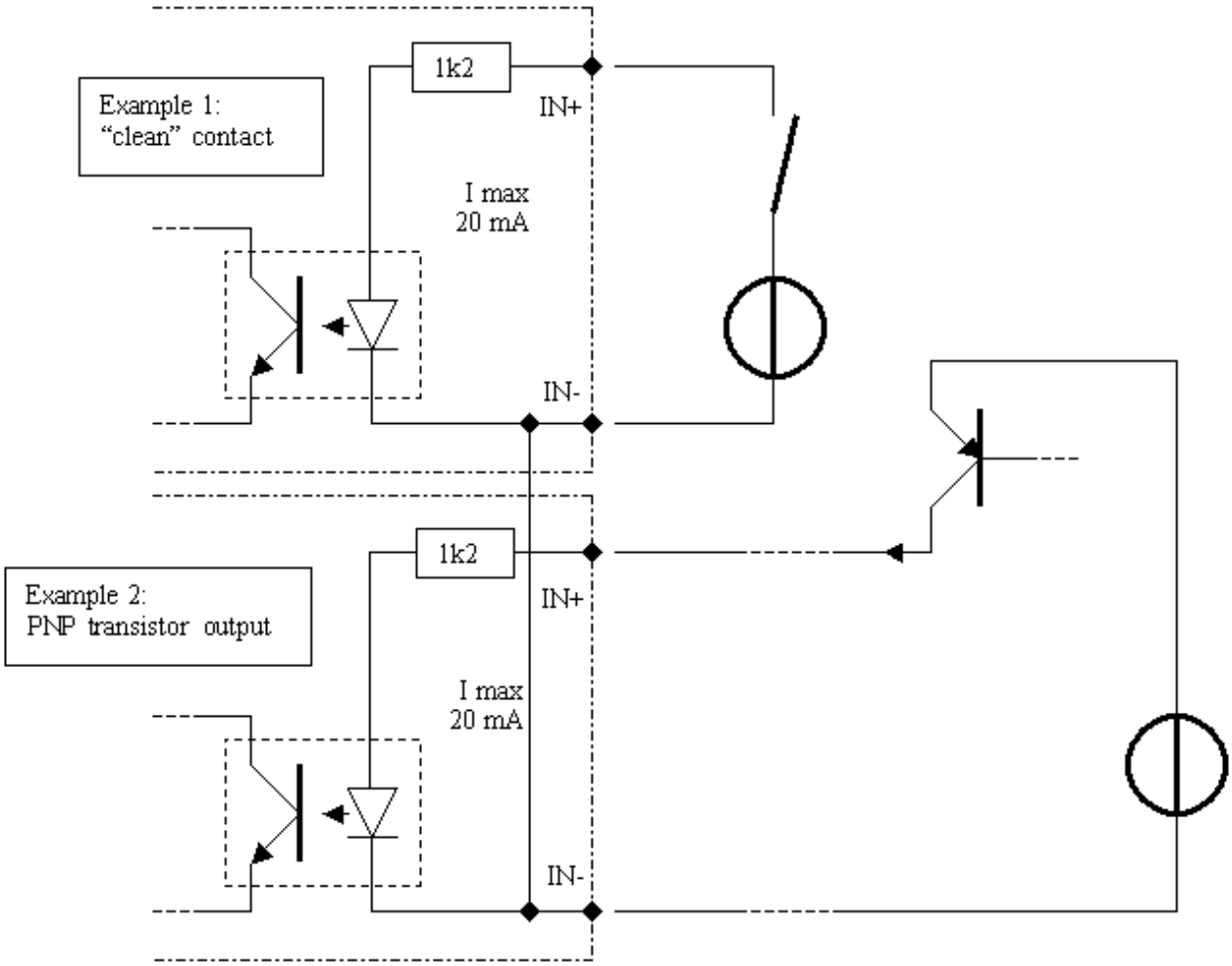


Power supply, serial line, input and output connections:

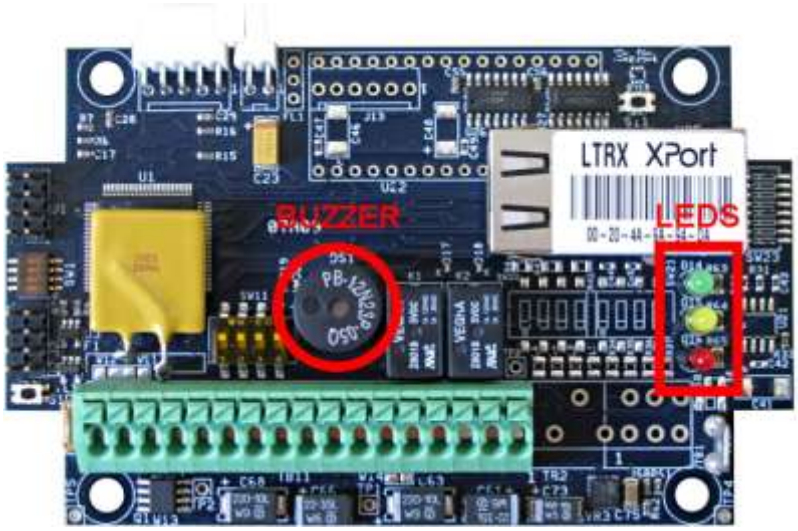
Pin	No	Min	Typical	Max	Description
+ PWR	1	10Vdc	12Vdc/24Vdc	27Vdc	DC power supply
- PWR (Gnd)	2				DC power supply return path
RS232 Tx	3				RS232 connection (output to host)
RS232 Rx	4				RS232 connection (input from host)
RS232 Gnd	5				RS232 connection (reference)
RS485 RT+	6				RS485 connection (positive)
RS485 RT-	7				RS485 connection (negative)
IN1 +	8	10Vdc	12Vdc/24Vdc (10mA/20mA)	27Vdc	Input 1 connection
IN1 -	9				Input 1 connection reference (same as Input 2 ref.)
IN2 +	10	10Vdc	12Vdc/24Vdc (10mA/20mA)	27Vdc	Input 2 connection

IN2 -	11				Input 2 connection reference (same as Input 1 ref.)
OUT1 NA	12			1A@30Vdc 0.5A@125Vac	Relay 1 NO contact connection
OUT1 COM	13				Relay 1 COMmon contact connection
OUT1 NC	14			1A@30Vdc 0.5A@125Vac	Relay 1 NC contact connection
OUT2 NA	15			1A@30Vdc 0.5A@125Vac	Relay 2 NO contact connection
OUT2 COM	16				Relay 2 COMmon contact connection
OUT2 NC	17			1A@30Vdc 0.5A@125Vac	Relay 2 NC contact connection

7 Digital Input Wiring Examples



8 Status Indications: Led and Buzzer



LEDs: the following table specifies the meaning of the LEDs.

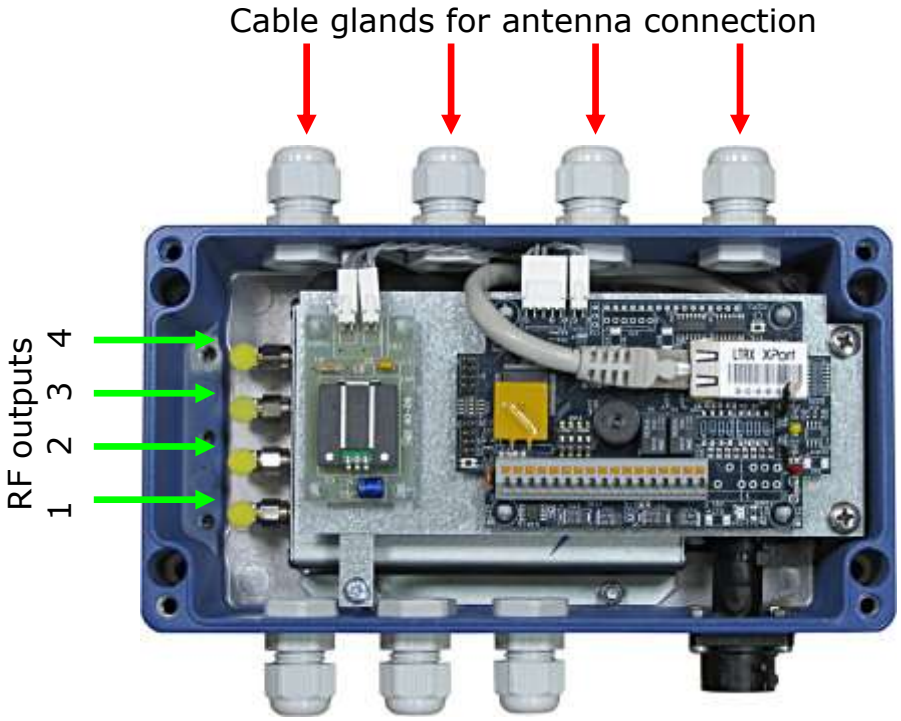
Led	Status	Description
Green	On	Power supply ok
Yellow	Flashing On	Tag absent Tag present
Red	On	Missing parameters / SW1-1 switched on

Buzzer: if the 'automatic' management of the buzzer is enabled by the flag defined in the general parameters, the buzzer will be activated for 0.5 seconds at every identification of a 'new' tag.

LEDs and buzzer behaviour during the **BLUEBOX** start up: LEDs will be switched on during the whole initialization phase (about 5 seconds), at the end of the initialization phase, the buzzer will be switched on for 0.5 seconds, afterwards the LEDs and the buzzer will work as described above.

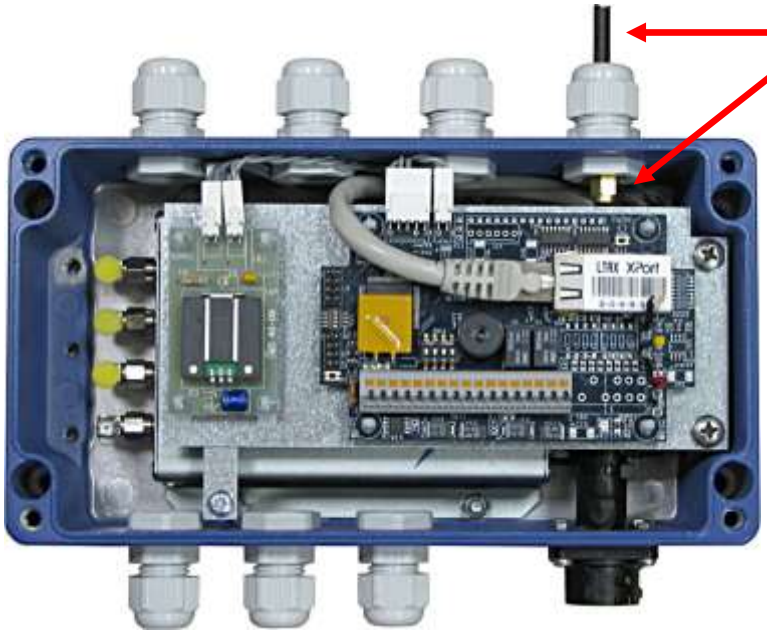
9 Antennas

The **BLUEBOX** is equipped with 4 cable glands (one for each possible antenna connection) which allow the passage of the antenna preassembled cable (RG58 terminated with a SMA connector). The SMA connector of the antenna cable fits directly to the internal UHF controller RF output connector. The antennas are available in various models (items 902xU).

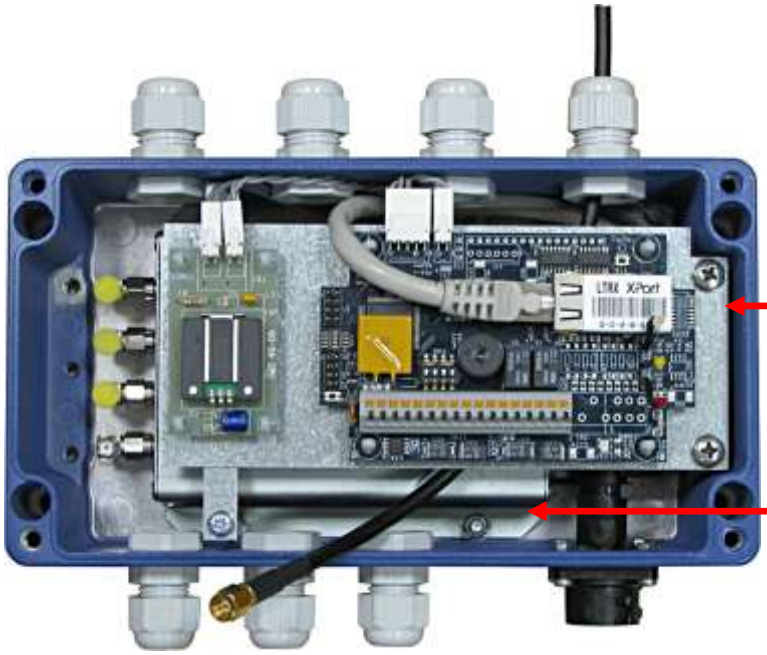


The read range of an RFID system always depends on various factors like antenna size, tag size, tag IC type, orientation between tag and reader antenna, position of the tag versus the reader antenna, noise environment, metallic environment, RF power, etc. Therefore all data about read ranges can only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

The following pictures depict through an example the sequence of operations that have to be executed for connecting an external RF antenna to the **BLUEBOX**.



Pass the antenna cable through the cable gland



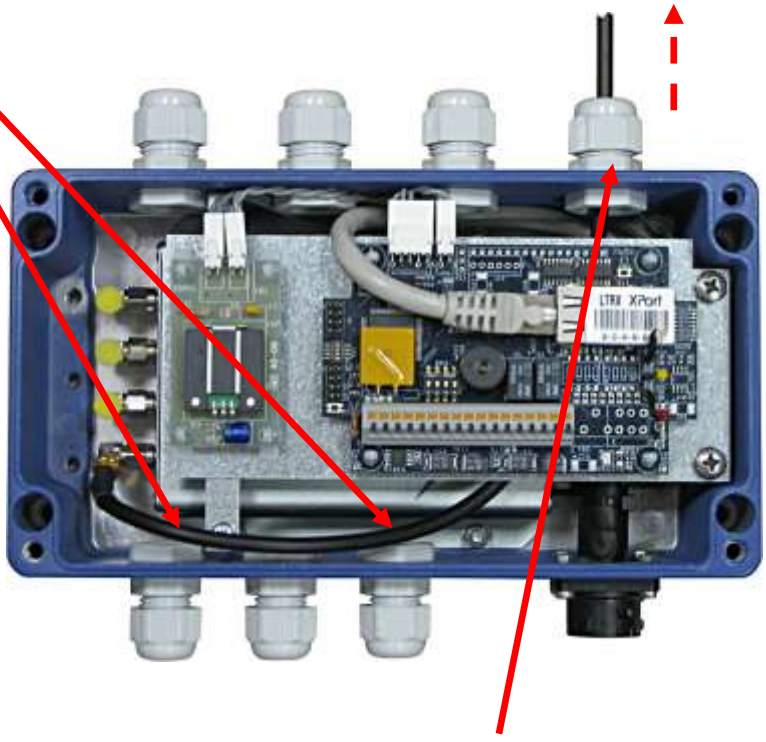
Pass the antenna cable between the electronic board support and the UHF controller

Electronic board support

UHF controller

Connect the antenna cable to the RF output

Arrange the cable length pulling it out in order to allow an easy reassembling of the cover of the enclosure, adopt appropriate radius of curvature of the cable (do not damage it)



In conclusion, screw the cable gland in order to get the IP65 protection classification

9.1 Circularly Polarised Antenna, 245x235x40 mm

9021U

Operating frequency	860 MHz – 960 MHz
Polarisation	Circular
Impedance	50 Ohm
Connector	SMA, male
Dimensions	245x235x40 mm
Protection class	IP 65
Connection cable	RG58
Connection cable length	3 m
Operating temperature	-10°C … +55°C
Reading distance	<= 10 m

9.2 Linearly Polarised Antenna, 245x235x40 mm

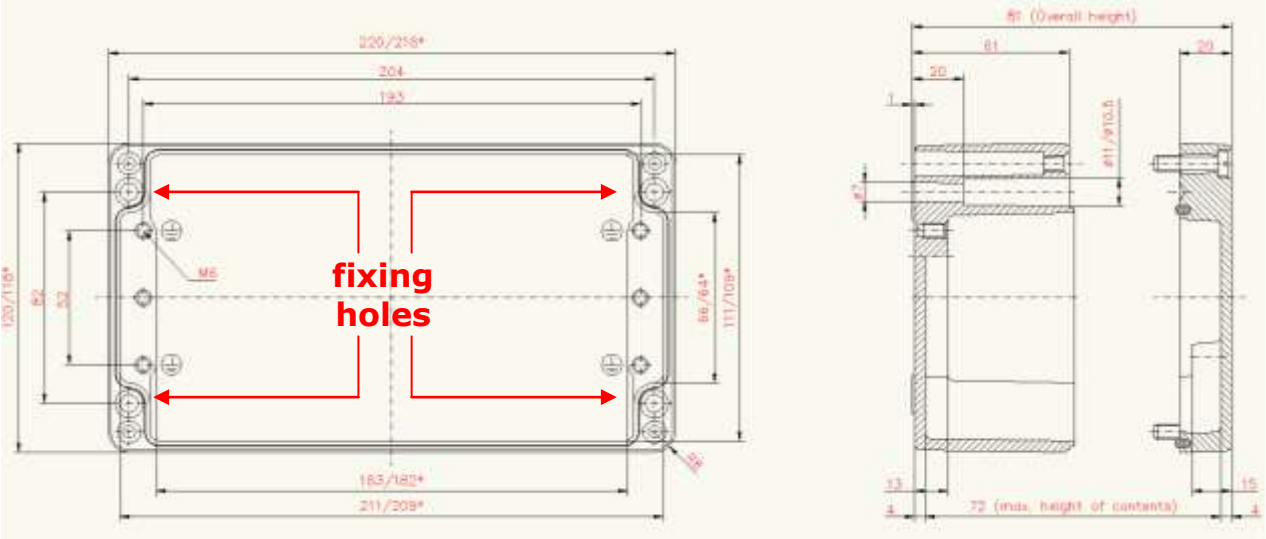
9022U

Operating frequency	860 MHz – 960 MHz
Polarisation	Linear
Impedance	50 Ohm
Connector	SMA, male
Dimensions	245x235x40 mm
Protection class	IP 65
Connection cable	RG58
Connection cable length	3 m
Operating temperature	-10°C … +55°C
Reading distance	<= 10 m

10 Installation

To install the **BLUEBOX**, it is necessary to remove the cover of the enclosure by loosening the 4 locking screws.

Fix the bottom of the enclosure to a support (wall, column, ..) using the 4 holes (already provided within the enclosure) and choosing suitable screws.



Pass the connecting cables through the cable glands (PG). Connect the cable wires (section: 0.5 ... 1.5 mm^2) to the spring-cage terminal blocks using suitable end terminals.

Verify the HW settings (dip switches).

Reassemble and screw up the case cover. It is recommended to pay particular attention avoiding to damage the LEDs.

The antennas (items 9021U and 9022U) can be fixed using the four M6 threaded inserts disposed on the back.

11 Ethernet Module

The **BLUEBOX** is also available with an Ethernet communication interface (item 5032U). In order to get an IP65 protection classification, the housing is equipped with an appropriate female RJ45 panel receptacle and is furnished with a protective composite plug that allows to assemble a standard RJ45 male cord-set; a push pull mechanism is used for the coupling. In order to work in a LAN, the Ethernet module inside the **BLUEBOX** must be configured applying the procedure described below.

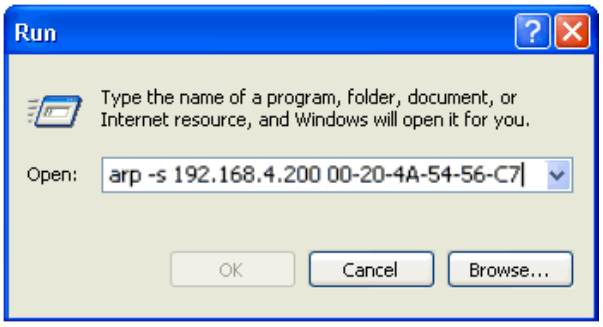
Ethernet connector

plug assembly of a standard RJ45 male cord set

11.1 Configuration Menu Access

To set up the Ethernet module, it is necessary to connect the **BLUEBOX** to an Ethernet network (LAN) applying the reported procedure on a PC (connected to the same network without gateways). Since in the first starting operation the Ethernet module do not have a known IP address, the following procedure will assign a temporary IP address, in order to connect the module and set all the necessary parameters.

1) Click on ***START*** from the toolbar and select ***RUN***; the following window will appear and then type the command described below:

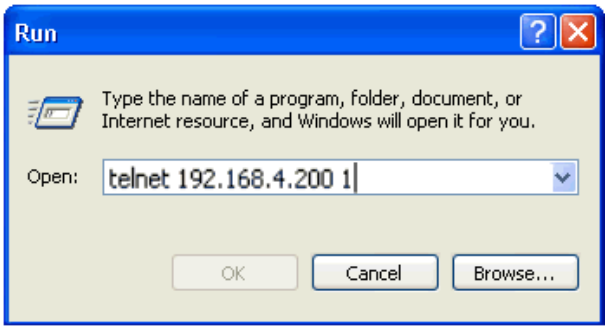


a) **arp** is the command used to assign a temporary IP address (if this command is not available on your PC, then it is necessary to install the TCP/IP protocol on your Microsoft Windows operating system);

b) **192.168.4.200** is the temporary IP address assigned to the **BLUEBOX;**

c) **00-20-4A-54-56-C7** is the MAC address of the **BLUEBOX** Ethernet module, as printed on the metal housing of the module. This example describes the MAC address of the module used to redact this document.

2) To activate the address, connect to Telnet port 1. Type the following command in the Run window. If Telnet is not available on PC, install it from Win9x.

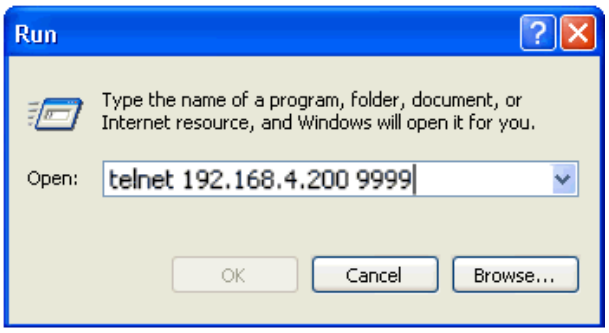


3) Therefore, it will appear either this message, or the following momentary message (that will automatically disappear)

```
Connessione a 192.168.4.200...
```

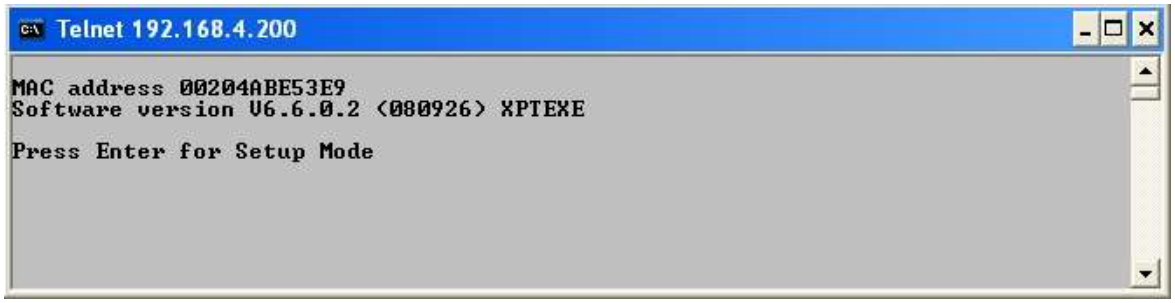
Although these error messages, the address has been temporarily assigned in the right way to the **BLUEBOX** Ethernet module and it is possible to continue.

4) At this point it is possible to enter in the module configuration menu by connecting to Telnet port 9999, as shown in the following picture:

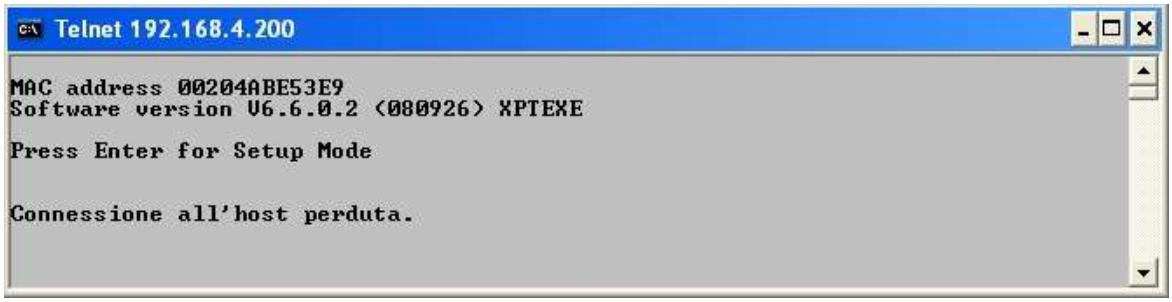


Note: The temporary address assigned in the previous points 1 and 2 will be deleted after the first reset of the module. It is recommended to save as soon as possible the configuration.

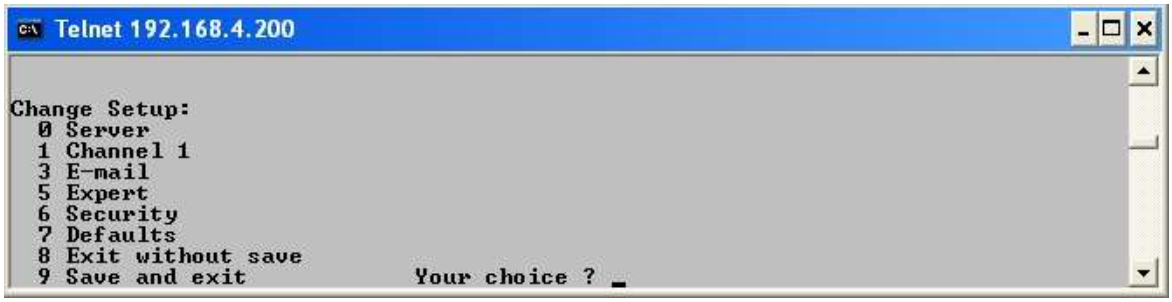
The access to the configuration menu is shown in the following screen shot:



Press ENTER within 5 seconds to enter the configuration menu, otherwise the Ethernet module will be reset automatically as shown below; to retry the access to the configuration menu, it is necessary to repeat all the operations described before.



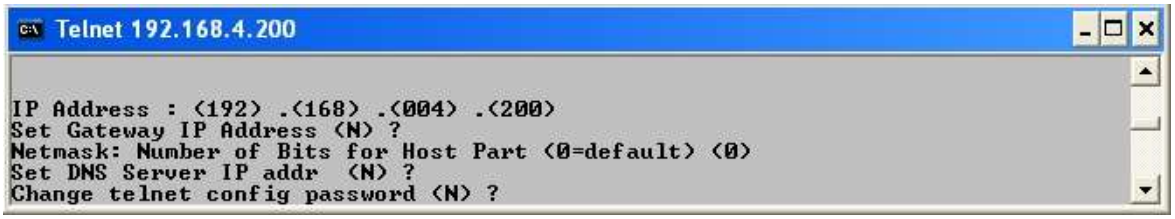
Pressing the ENTER key within 5 seconds, the following screen shot will appear:



The configuration menu permits to know and modify the module parameters. In this case of integration of the Ethernet module in the BLUEBOX, only the primordial parameters are explained, the other ones must be set to the indicated values.

11.2 Configuration Menù: Server Configuration (0 choice)

It is possible to configure the module basic network parameters by pressing 0 (zero) and then ENTER to confirm the selection. The current values are displayed in parenthesis.



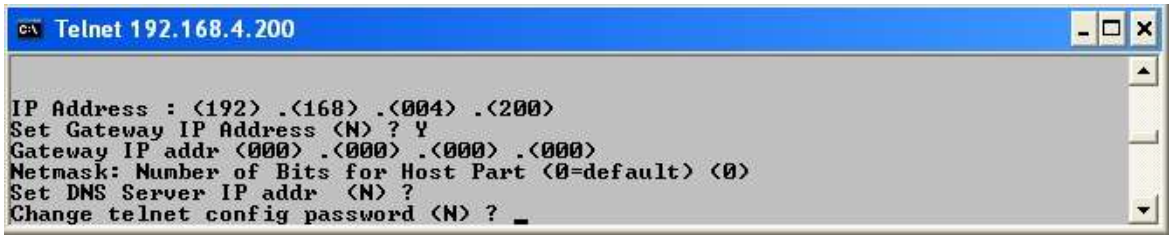
- IP Address

  The IP address is a fundamental parameter used to communicate with the module, distinguishing it in univocally in the network. For this reason it is not possible to assign the same IP address to different modules connected to the same network. If an address is set and is already in use on any other unit in the same network, the module will generate an error

code through the leds and it won't be connected to the network. Contact always the network administrator to get valid addresses. Enter each octet and press ENTER between each section; to maintain the same value press only ENTER.

- Gateway Address

  The gateway address, or router, allows communication to other LAN segments. The gateway address should be the IP address of the router connected to the same LAN segment as the module. The gateway address must be within the local network. The default is N (No), meaning the gateway address has not been set. To set the gateway address, type Y and enter the address. Contact always the network administrator to get valid addresses.



- Net Mask

  A net mask defines the number of bits taken from the IP address that are assigned for the host part. This number is defined in function of the used network class that is in function of the number of connectable devices: up to 16.777.216 for class A, up to 65.534 for class B and up to 254 for class C.

  The module suggests the number of bits for the host (according to the IP address set) and calculates automatically the net mask. With the value set to 0 (default), the net mask is calculated automatically, according to the following tables:

Class	Valid IP Addresses	Reserved IP Addresses
A	Da: 1.0.0.0 a: 126.0.0.0	0.0.0.0 127.0.0.0
B	Da: 128.1.0.0 a: 191.254.0.0	128.0.0.0 191.255.0.0

Class	Valid IP Addresses	Reserved IP Addresses
C	Da: 192.0.1.0 a: 223.255.254.0	192.0.0.0 223.255.255.0

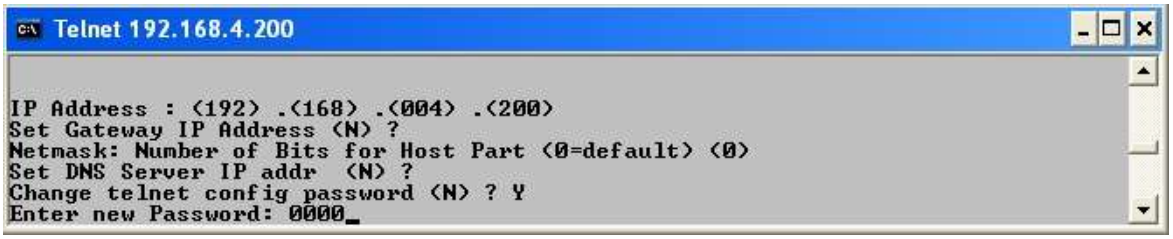
Standard IP Network Netmasks			
Network Class	**Network Bits**	**Host Bits**	**Netmask**
A	8	24	255.0.0.0
B	16	16	255.255.0.0
C	24	8	255.255.255.0

For IP addresses and Net Masks not in accordance with the previous tables, refer to the following examples set the value according to the reported Host Bits values:

Netmask examples and related Host Bits			
Netmask	**Host Bits**	**Netmask**	**Host Bits**
255.255.255.252	2	255.255.254.0	9
255.255.255.248	3	255.255.252.0	10
255.255.255.240	4	255.255.248.0	11
255.255.255.224	5	255.255.240.0	12
255.255.255.192	6	…	…
255.255.255.128	7	255.128.0.0	23
255.255.255.0	8	255.0.0.0	24

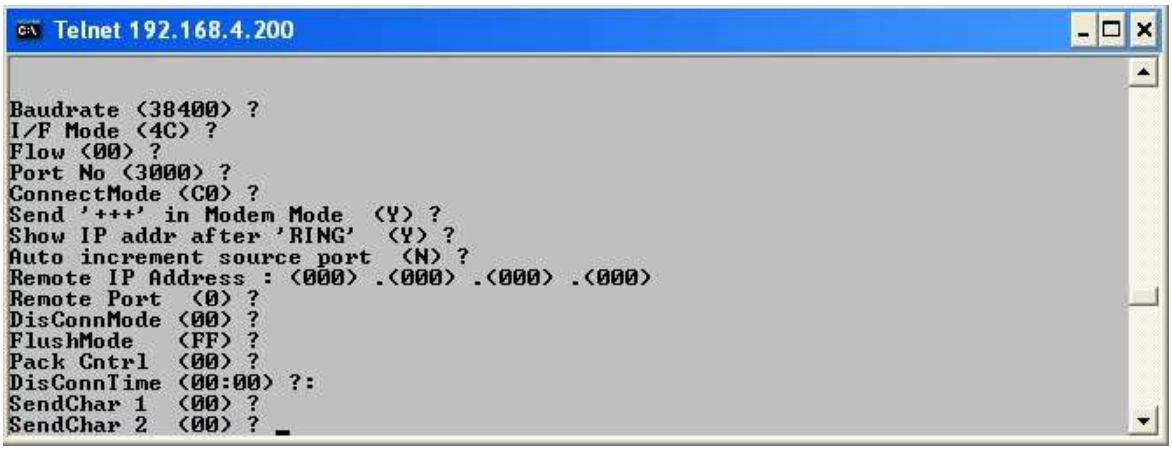
- Telnet Password

  Allows to set a password, composed of 4 characters, to protect the network access of the configuration menu of the module. Note: keep the password in a safe place. In case of loss it is necessary to send the module to technical assistance. To set the telnet password, type Y and enter the new password.



11.3 Configuration Menu: Channel 1 Configuration (1 choice)

It is possible to modify the module serial communication parameters (Ethernet module versus **BLUEBOX** link) by pressing 1 and ENTER to confirm the selection.



Set the values according to the following table:

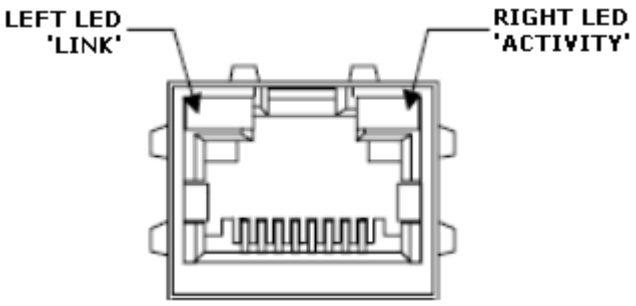
Parameter	Value
Baudrate	38400
I/F Mode	4C
Flow	00
Port No	3000
Connect Mode	C0
Send '+++' in Modem Mode	Y (indifferente)
Auto increment source port	N
Remote IP Address	000.000.000.000 (indifferente)

Parameter	Value
Remote Port	00
DisConnMode	00
Flush Mode	FF
Pack Control	00
DisConnTime	00:00
Send Char 1	00
Send Char 2	00

11.4 Configuration Menu Exit (7/8/9 choice)

When the Ethernet module configuration of the **BLUEBOX** is finished, exit the menu without saving the changes by pressing 8, or saving them by pressing 9. It is also possible to restore the initial factory values, by pressing 7.

11.5 Status LEDs



The following table specifies the meaning of 2 bi-color status leds integrated in the RJ45 connector of the module inside the **BLUEBOX**:

LINK LED Lato SX (left)		ACTIVITY LED Lato DX (right)	
Color	**Meaning**	**Color**	**Meaning**
Off	No link	Off	No activity
Amber	10 Mbps	Amber	Half duplex
Green	100 Mbps	Green	Full duplex

11.6 Configuration Troubleshooting Problems

PROBLEMA	MOTIVO	SOLUZIONE
When the ARP –S command is issued and an error message appear	The currently logged-in user does not have the correct rights to use this command on this PC	Log-in through a user with sufficient rights (e.g. administrator).
When the TELNET to port 1 command (to assign an IP address to the module) is issued and the TELNET does not respond for a long time	The Ethernet entered address with the ARP command is incorrect	Check the address and send it again
	The sent IP address is not compatible with the current subnet mask of the PC	Check the IP address of the PC and be sure that the subnet mask is compatible with the IP address to send to the module
	The module is not plugged into the network properly	Check that link led of the module is lit, otherwise check carefully both the RJ45 plug and the network cable
The module answers to the PING command, but not to the TELNET command to port 9999	There is an IP address conflict on the network	Turn off the module and PING to the same address: in case of valid answer, there is another device in the network with the same address. Change the address of the module
	The Telnet configuration port (9999) is disabled within the module security settings	Set the module using the serial line

12 Document Revision History

Data	Revision	Firmware	Description
19/11/09	1.00	From 1.00	Initial release
16/03/10	1.01	From 1.01	Added the reading antenna information to data request and inventory commands as parameter flag that could be enabled or disable.
25/05/10	1.02	From 1.06	Added ISO 18000-6C read / write / lock / kill commands. Added the spontaneous mode on RS485 activateable with dip switches. Added the section related to the connection of the antenna cable shield to the ground using the faston connector on the board.
17/06/10	1.03	From 1.06	Deleted the section (previously added) related to the connection of the antenna cable shield to the ground using the faston connector on the board, because it is not applicable to this product.
21/07/10	1.04	From 1.08	Changed the read firmware version command by adding the auxiliary reader (the RFID front-end connected to the **BLUEBOX**) firmware version. Corrected the Ethernet module configuration parameters. Added the document revision history section (this section). New document formatting.
25/01/11	1.05	From 1.11	Changed the introduction section of the document. Changed the technical specifications and added the reading distance as operating feature. Corrected the default value of the filter time. Added the tag type information to data request and inventory commands and spontaneous message on RS232, RS485 and Ethernet interface as parameter flag that could be enabled or disable. Corrected the read status command by adding the RF status flag. Changed the ISO 18000-6C (EPC Class-1 Generation-2) inventory, read, write, lock and kill commands to support the variable length of the UID of the tags. Corrected and added some antennas features. Deleted the mechanical design of the antennas. Corrected the reader code in the Ethernet section. Added the supported transponder appendix with the

Data	Revision	Firmware	Description
			transponder description.
05/12/12	1.06		Auxiliary reader status bit information added. GND crosslink on digital inputs added. Reset command added.

A. Supported Transponders

Supported transponders by **BLUEBOX** are:

- ISO 18000-6B;
- ISO 18000-6C (EPC Class-1 Generation-2).

A.1 ISO 18000-6B

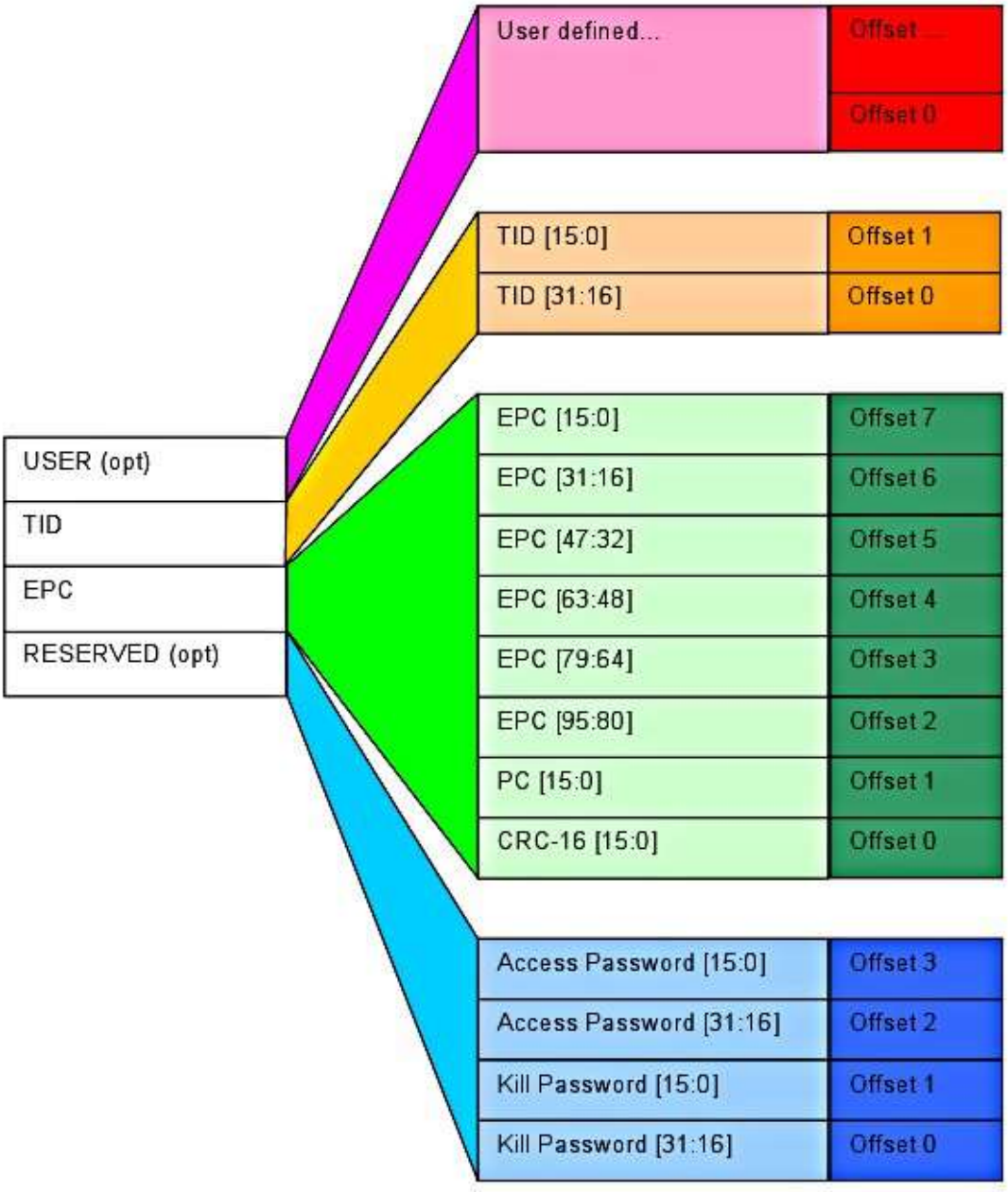
ISO 18000-6B transponders have a memory divided into blocks of 1 byte each, the first 8 bytes are the UID code of the transponder and are not changeable. The memory size is variable and depends on the transponder type, the maximum size is 256 bytes (2 kbits).

A.2 ISO 18000-6C (EPC Class-1 Generation-2)

ISO 18000-6C (EPC Class 1 Generation 2) transponders have a memory divided into 4 banks, even if some banks are optional and therefore not present. Each memory bank is divided into blocks of 16-bits each, the first block in each bank has address zero (00h).

Memory Bank	Description
RESERVED	Used to store the access password (32 bits) and the kill password (32 bits). The kill password is used to destroy the tag so that is no longer usable. It is an optional memory bank.
EPC	Used to store the CRC, PC and EPC. The CRC is 16 bits length and it is the checksum of the PC+EPC field, it is not directly writable. The PC (Protocol Control) is 16 bits length and the 5 most significant bits contain the EPC length information, for example PC = 0x0000 -> PC + EPC = 1 word (2 byte), PC = 0x0800 -> PC + EPC = 2 word (4 byte), PC = 0x1000 -> PC + EPC = 3 word (6 byte). The EPC has a variable length, the maximum length is 96 bits.
TID	Used to store a manufacturer identifier (16 bits). It is suggested to not modify the content of this memory bank.
USER	This memory bank is a User Memory, it is optional and the size depends on the tag.

The figure below shows the tag memory structure.



Access to each memory bank can be protected using the LOCK command to make it accessible only by knowing the password. When a memory bank is not protected is always accessible in both reading and writing (in this case we speak of "Open State"), when it is protected read and write operations are limited (in this case we speak of "Secured State "). The protection of access and kill passwords, allows a further degree of protection.