UHF
RFID System

BLUEBOX INDUSTRIAL UHF LONG RANGE

Profibus

Preface

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

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.

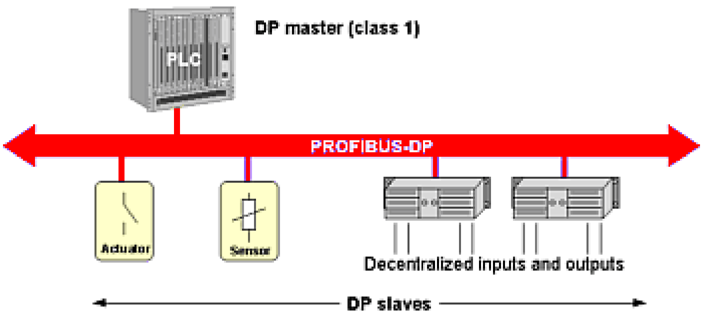
- The liability-prescriptions of the manufacturer in the issue valid at the time of purchase are valid for the device. The manufacturer shall not be held legally responsible for inaccuracies, errors, or omissions in the manual or automatically set parameters for a device or for an incorrect application 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 a read/write RFID device for industrial application (item 5033U) that communicates with a 'host' system (typically a PLC) through a PROFIBUS DP serial line. The **BLUEBOX** acts as a joint through a set of commands between the host system and a RFID tag (or transponder) present near the antenna.

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

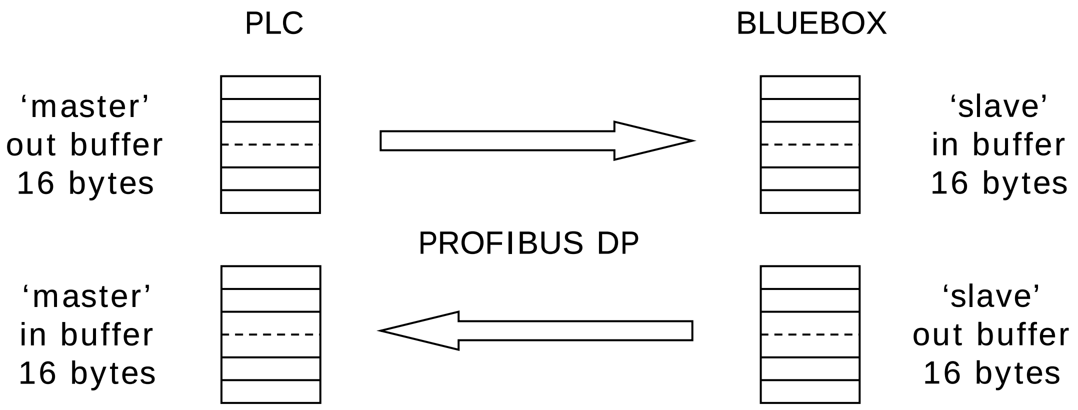


More devices can be connected on the bus, each one has a different address. Once configured, the communication with devices is cyclic.

Therefore the **BLUEBOX** is a remote device ('slave') with a 16-bytes input buffer and a 16-bytes output buffer. Similarly, in order to talk to the **BLUEBOX**, the PLC ('master') has a 16-bytes output buffer (corresponding to the 16-bytes input buffer of the **BLUEBOX**) and a 16-bytes input buffer (corresponding to the 16-bytes output buffer of the **BLUEBOX**). The input buffer of the **BLUEBOX** can only be modified by the 'master' (PLC) while the output buffer of the **BLUEBOX** can only be modified by the 'slave' that is the **BLUEBOX**. The aim of the cyclic communication is to keep up to date the corresponding buffers at the 'master' side and at the 'slave' side.

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

For the developers: the **BLUEBOX** includes the Profibus module **ANYBUS-IC AB6000** of **HMS** (www.anybus.com)**,** a .GSD (Generic Station Description) is provided that defines the characteristics of the device and is necessary for the configuration of the Profibus network.



The **BLUEBOX** can also communicate through the RS232/RS485 serial line as for the items 5031U and 5032U; in this case of use, refer to the B5U001xxxE relative manual for the communication protocol.

Through these communication channels, it is also possible to configure the functional parameters and to upgrade the firmware (only through the RS232/RS485 serial line). Furthermore the **BLUEBOX** is able to handle two digital outputs (relays) and two opto-isolated digital inputs. 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 **BLUEBOX** is equipped with useful spring-cage 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 1 W (30 dBm) software programmable
Antenna	Up to 4 external antennas
Reading distance	10 m[1]
Supported tags	ISO 18000-6B and ISO 18000-6C (EPC Class-1 Generation-2)
Communication interface	RS232 / RS485 Profibus
Signalling	3 LED, Buzzer
Digital inputs	2 opto-isolated inputs voltage 12Vdc ... 24Vdc max current 20mA Debounce 16ms
Digital outputs	2 channels (relay) no/nc contacts max 1A @ 30Vdc max 0.5A @ 125Vac
Dimensions	120 x 220 x 80 mm
Protection classification	IP 65
Operating temperature	-10°C ... +45°C
Connections	Spring-cage terminal blocks (cable section: 0.5 … 1.5 mm²)

[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

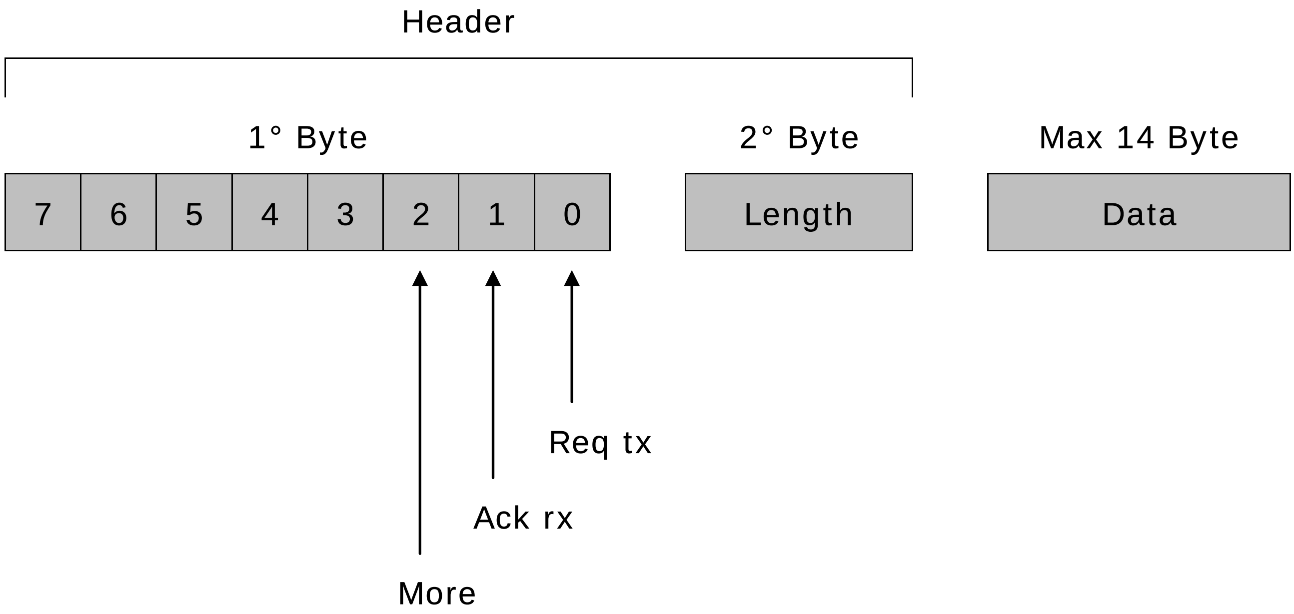
List of configurable RF parameters:

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

There are two types of data packets used in the communication:
Outgoing data packets from the ‘master’ (PLC) to the ‘slave’ in order to send a command or an answer to the **BLUEBOX**
Incoming data packets to the ‘master’ (PLC) from the ‘slave’ (**BLUEBOX**) carrying for example the answer to a command

The outgoing data packets from the ‘master’ assume the following structure:



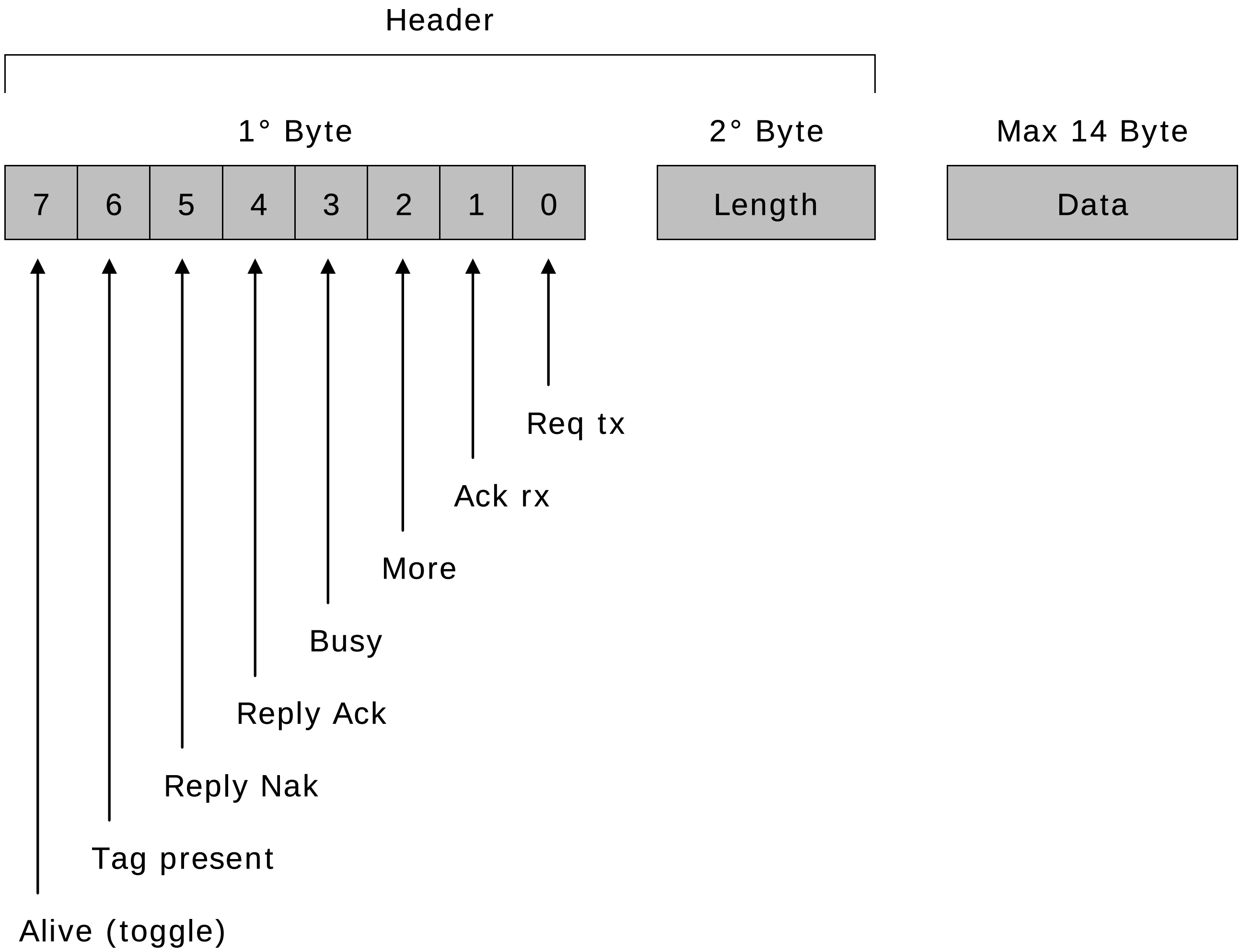
The outgoing data packet consists of a header (first two bytes) and a data buffer of 14 bytes.

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

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

The second byte (Length) of the header specifies the number of data bytes in the data buffer (max data bytes 14).

The incoming data packets to the ‘master’ assume the following structure:



Also the incoming data packet consists of a header (first two bytes) and a data buffer of 14 bytes.

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

- Bit 7: Alive, toggles every second and means that the 'slave' is running correctly;
- Bit 6: Tag present, set to '1' by the 'slave' means that a transponder is present near the antenna (detected by the 'continuous' identification activity);
- Bit 5: Reply Nak, set to '1' by the 'slave' means that an error has occured;
- Bit 4: Reply Ack, set to '1' by the 'slave' means that the received command has been processed;
- Bit 3: Busy, set to '1' by the 'slave' means that the 'slave' is processing the command message from the 'master';
- Bit 2: More, set to '1' by the 'slave' means that the message is composed of several data packets;

- Bit 1: Req tx, reception request (from ‘slave’);
- Bit 0: Ack tx, transmission acknowledge.

The second byte (Length) of the header specifies the number of data bytes in the data buffer (max data bytes 14).

The messages consist of one or more data packets. If the length of the message is shorter than 14 bytes, the message will be composed of only 1 data packet. If the length of the message is bigger than 14 bytes, the message will be composed of more than 1 data packet; in this case the header of all the transmitted data packets, apart the last one, will present at ‘1’ the flag ‘More’ indicating that the message is not completed and another data packet will follow.

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

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

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

The answer of the ‘slave’ to a command message from the ‘master’ can take place through a full answer message or in a short form depending of the type of command. In the case of a short form answer, it take place through the setting to ‘1’ of the flag ‘Reply ack’ or the flag ‘Reply nak’ (in function of the result of the execution of the command).

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

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

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

Example:

The ‘master’ device requests the code of the transponder present near the antenna; for the example, we suppose that the transponder is a ISO 18000-6C (EPC Class-1 Generation-2) with the following hex code: **0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F**.

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

In this case, the message from the ‘master’ command is composed of the byte 0x05 (code of the ‘request data’ command).

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

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

1° Byte								2° Byte	Data	
0	0	0	0	0	0	0	1	01	05	OUT
T	1	0	0	0	0	0	0			IN

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

1° Byte								2° Byte	Data	
0	0	0	0	0	0	0	1	01	05	OUT
T	1	0	0	0	0	0	1			IN

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

1° Byte								2° Byte	Data	
0	0	0	0	0	0	0	0			OUT
T	1	0	0	0	0	0	1			IN

1° Byte								2° Byte	Data	
0	0	0	0	0	0	0	0			OUT
T	1	0	0	0	0	0	0			IN

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

1° Byte								2° Byte	Data	
0	0	0	0	0	0	0	0			OUT
T	1	0	0	1	0	0	0			IN

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

The message consists of 2 data blocks because the code length is 20 bytes and the data buffer can only support at most 14 bytes.

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

1° Byte								2° Byte	Data														
0	0	0	0	0	0	0	0																OUT
T	1	0	0	0	1	1	0	0E	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	IN

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

1° Byte								2° Byte	Data														
0	0	0	0	0	0	1	0																OUT
T	1	0	0	0	1	1	0	0E	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	IN

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

1° Byte								2° Byte	Data	
0	0	0	0	0	0	0	0			OUT
T	1	0	0	0	0	0	1			IN

1° Byte								2° Byte	Data	
0	0	0	0	0	0	0	0			OUT
T	1	0	0	0	0	0	0			IN

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

1° Byte								2° Byte	Data														
0	0	0	0	0	0	0	0																OUT
T	1	0	0	0	0	1	0	02	0E	0F													IN

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

1° Byte								2° Byte	Data														
0	0	0	0	0	0	1	0																OUT
T	1	0	0	0	0	1	0	02	0E	0F													IN

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

1° Byte								2° Byte	Data	
0	0	0	0	0	0	0	0			OUT
T	1	0	0	0	0	0	1			IN

1° Byte								2° Byte	Data	
0	0	0	0	0	0	0	0			OUT
T	1	0	0	0	0	0	0			IN

4.1 Parameters Programming

This command is used to set the communication (not Profibus) and operating parameters of the **BLUEBOX**.

Byte	Value	Description	Note
1	0x2F	Command code	
2	0x..	New address to be setted.	Address to be used with RS232 / RS485 link (no Profibus).
3	0x..	High nibble: Communication baud rate of RS232 / RS485 interface: • 0x0 -> 1200 bps; • 0x1 -> 2400 bps; • 0x2 -> 4800 bps; • 0x3 -> 9600 bps; • 0x4 -> 19200 bps • 0x5 -> 38400 bps. Low nibble: Communication data bits of RS232 / RS485 interface: • 0x7 -> 7 bits; • 0x8 -> 8 bits.	Comm. parameters of RS232 / RS485 link (no Profibus).
4	0x..	High nibble: Communication stop bits of RS232 / RS485 interface: • 0x1 -> 1 bit; • 0x2 -> 2 bits. Low nibble: Communication parity of RS232 / RS485 interface: • 0x0 -> None; • 0x1 -> Even; • 0x2 -> Odd.	Comm. parameters of RS232 / RS485 link (no Profibus).
5	0x00		
6	0x..	Tag type. The single 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.	
7	0x..	Filter time: • Decimal 0 … 99 for time in	

Byte	Value	Description	Note
		seconds (0 … 99 seconds); • Decimal 100 … 199 for time in minutes (0 … 99 minutes).	
8	0x..	Flags. The single 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.	

The answer to this command is given in short form through the ‘reply ack’ / ‘reply nak’ flags (this command doesn’t ‘produce’ data).

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**.

Byte	Value	Description	Note
1	0x3B	Command code.	
2	0x..	Antenna 1 power in 0.1 dBm, high byte.	0x0000 -> Antenna off.
3	0x..	Antenna 1 power in 0.1 dBm, low byte.	
4	0x..	Antenna 1 duration in milliseconds, high byte.	0x0000 -> Antenna off.
5	0x..	Antenna 1 duration in milliseconds, low byte.	
6	0x..	Antenna 2 power in 0.1 dBm, high byte.	0x0000 -> Antenna off.
7	0x..	Antenna 2 power in 0.1 dBm, low byte.	
8	0x..	Antenna 2 duration in milliseconds, high	0x0000 -> Antenna off.

		byte.	
9	0x..	Antenna 2 duration in milliseconds, low byte.	
10	0x..	Antenna 3 power in 0.1 dBm, high byte.	0x0000 -> Antenna off.
11	0x..	Antenna 3 power in 0.1 dBm, low byte.	
12	0x..	Antenna 3 duration in milliseconds, high byte.	0x0000 -> Antenna off.
13	0x..	Antenna 3 duration in milliseconds, low byte.	
14	0x..	Antenna 4 power in 0.1 dBm, high byte.	0x0000 -> Antenna off.
15	0x..	Antenna 4 power in 0.1 dBm, low byte.	
16	0x..	Antenna 4 duration in milliseconds, high byte.	0x0000 -> Antenna off.
17	0x..	Antenna 4 duration in milliseconds, low byte.	

The command consists of 17 data bytes and then it will be transmitted with 2 data packets (14 bytes in the 1st packet and 3 bytes in the 2nd one).

The answer to this command is given in short form through the ‘reply ack’ / ‘reply nak’ flags (this command doesn’t ‘produce’ data).

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 and general parameters of the **BLUEBOX**.

Byte	Value	Description	Note
1	0x31	Command code.	

The answer to this command is given in short form through the ‘reply ack’ / ‘reply nak’ flags (this command doesn’t ‘produce’ data).

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**.

Byte	Value	Description	Note
1	0x2A	Command code.	

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

Byte	Value	Description	Note
1	0x2A	Command code	
2	0x..	High nibble: Communication baud rate of RS232 / RS485 interface: • 0x0 -> 1200 bps; • 0x1 -> 2400 bps; • 0x2 -> 4800 bps; • 0x3 -> 9600 bps; • 0x4 -> 19200 bps • 0x5 -> 38400 bps. Low nibble: Communication data bits of RS232 / RS485 interface: • 0x7 -> 7 bits; • 0x8 -> 8 bits.	Comm. parameters of RS232 / RS485 link (no Profibus).
3	0x..	High nibble: Communication stop bits of RS232 / RS485 interface: • 0x1 -> 1 bit; • 0x2 -> 2 bits. Low nibble: Communication parity of RS232 / RS485 interface: • 0x0 -> None; • 0x1 -> Even; • 0x2 -> Odd.	Comm. parameters of RS232 / RS485 link (no Profibus).
4	0x00		
5	0x..	Tag type. The single bits are dedicated to disable (0 value) or enable (1 value) functions: • Bit 7 … 2: not used;	

Byte	Value	Description	Note
		• Bit 1: C type; • Bit 0: B type.	
6	0x..	Filter time: • Decimal 0 … 99 for time in seconds (0 … 99 seconds); • Decimal 100 … 199 for time in minutes (0 … 99 minutes).	
7	0x..	Flags. The single 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**.

Byte	Value	Description	Note
1	0x3A	Command code.	

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

Byte	Value	Description	Note
1	0x3A	Command code.	
2	0x..	Antenna 1 power in 0.1 dBm, high byte.	0x0000 -> Antenna off.
3	0x..	Antenna 1 power in 0.1 dBm, low byte.	

4	0x..	Antenna 1 duration in milliseconds, high byte.	0x0000 -> Antenna off.
5	0x..	Antenna 1 duration in milliseconds, low byte.	
6	0x..	Antenna 2 power in 0.1 dBm, high byte.	0x0000 -> Antenna off.
7	0x..	Antenna 2 power in 0.1 dBm, low byte.	
8	0x..	Antenna 2 duration in milliseconds, high byte.	0x0000 -> Antenna off.
9	0x..	Antenna 2 duration in milliseconds, low byte.	
10	0x..	Antenna 3 power in 0.1 dBm, high byte.	0x0000 -> Antenna off.
11	0x..	Antenna 3 power in 0.1 dBm, low byte.	
12	0x..	Antenna 3 duration in milliseconds, high byte.	0x0000 -> Antenna off.
13	0x..	Antenna 3 duration in milliseconds, low byte.	
14	0x..	Antenna 4 power in 0.1 dBm, high byte.	0x0000 -> Antenna off.
15	0x..	Antenna 4 power in 0.1 dBm, low byte.	
16	0x..	Antenna 4 duration in milliseconds, high byte.	0x0000 -> Antenna off.
17	0x..	Antenna 4 duration in milliseconds, low byte.	

The answer consists of 17 data bytes and then it will be transmitted with 2 data packets (14 bytes in the 1st packet and 3 bytes in the 2nd one).

4.6 FW Version Reading

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

Byte	Value	Description	Note
1	0x34	Command code.	

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

Byte	Value	Description	Note
1	0x34	Command code.	

2	0x..	Firmware version, 1st ASCII char.	
3	0x..	Firmware version, 2nd ASCII char.	
...	...		
16	0x..	Firmware version, 15th ASCII char.	
17	0x..	Firmware version, 16th ASCII char.	

The answer consists of 17 data bytes and then it will be transmitted with 2 data packets (14 bytes in the 1st packet and 3 bytes in the 2nd one). The 16 bytes means a string of 16 ASCII chars which define the firmware version, for example: 'BRIDGE_UHF_1.01l' indicates that this is this is a **BRIDGE** in **UHF** configuration (**U**ltra **H**igh **F**requency 860 - 960 MHz) with firmware version **1.01s long range**.

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

Byte	Value	Description	Note
1	0x34	Command code.	
2	0x01		

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

Byte	Value	Description	Note
1	0x34	Command code.	
2	0x..	Firmware version, 1st ASCII char.	
3	0x..	Firmware version, 2nd ASCII char.	
...	...		
64	0x..	Firmware version, 63th ASCII char.	
65	0x..	Firmware version, 64th ASCII char.	

The answer consists of 65 data bytes and then it will be transmitted with 4 data packets (14 bytes in the 1st, 2nd, 3rd and 4th packet and 8 bytes in the 5th one).

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 and sends back the result of the operation.

Byte	Value	Description	Note
1	0x05	Command code.	

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

a) case of no transponder

Byte	Value	Description	Note
1	0x00		
2	0x00		
3	0x00		
4	0x00		
5	0x00		

b1) case of one transponder present; antenna and transponder type indication flags are disabled:

Byte	Value	Description	Note
1	0x..	Transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		
...	0x..	Transponder code, last byte.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

b2) case of one transponder present; antenna indication flag is enabled and transponder type indication flag is disabled:

Byte	Value	Description	Note
1	0x..	Transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		
...	0x..	Transponder code, last byte.	
...	0x..	Reading antenna: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

b3) case of one transponder present; antenna indication flag is disabled and transponder type indication flag is enabled:

Byte	Value	Description	Note
1	0x..	Transponder type: • 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC Class-1 Generation-2).	
2	0x..	Transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		

...	0x..	Transponder code, last byte.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

b4) case of one transponder present; antenna and transponder type indication flags are enabled:

Byte	Value	Description	Note
1	0x..	Transponder type: • 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC Class-1 Generation-2).	
2	0x..	Transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		
n+1	0x..	Transponder code, last byte.	
n+2	0x..	Reading antenna: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

c1) case of m transponders present; antenna and transponder type indication flags are disabled:

Byte	Value	Description	Note
1	0x..	1st transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C

			(EPC Class-1 Generation-2).
...	...		
...	0x..	1st transponder code, last byte.	
...	...		
...	0x..	m-th transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		
...	0x..	m-th transponder code, last byte.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

c2) case of m transponders present; antenna indication flag is enabled and transponder type indication flag is disabled:

Byte	Value	Description	Note
1	0x..	1st transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		
...	0x..	1st transponder code, last byte.	
...	0x..	Reading antenna of the 1st tag: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	
...	0x..	m-th transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C

			(EPC Class-1 Generation-2).
...	...		
...	0x..	m-th transponder code, last byte.	
...	0x..	Reading antenna of the m-th tag: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

c3) case of one m transponders present; antenna indication flag is disabled and transponder type indication flag is enabled:

Byte	Value	Description	Note
1	0x..	1st transponder type: • 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC Class-1 Generation-2).	
2	0x..	1st transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		
...	0x..	1st transponder code, last byte.	
...	0x..	m-th transponder type: • 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC Class-1 Generation-2).	
...	0x..	m-th transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).

...	...		
...	0x..	m-th transponder code, last byte.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

c4) case of m transponders present; antenna and transponder type indication flags are enabled:

Byte	Value	Description	Note
1	0x..	1st transponder type: • 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC Class-1 Generation-2).	
2	0x..	1st transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		
n+1	0x..	1st transponder code, last byte.	
n+2	0x..	Reading antenna of the 1st tag: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	
...	0x..	m-th transponder type: • 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC Class-1 Generation-2).	
...	0x..	m-th transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		

Byte	Value	Description	Note
...	0x..	m-th transponder code, last byte.	
...	0x..	Reading antenna of the m-th tag: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

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 code present in the queue. After executing the command, the code must be deleted from the queue, otherwise each time you make a data request from the queue the same code will be returned.

Byte	Value	Description	Note
1	0x06	Command code.	

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

a) case of no transponder

Byte	Value	Description	Note
1	0x00		
2	0x00		
3	0x00		
4	0x00		
5	0x00		

b1) case of one transponder present; antenna and transponder type indication flags are disabled:

Byte	Value	Description	Note
1	0x..	Transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		
...	0x..	Transponder code, last byte.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

b2) case of one transponder present; antenna indication flag is enabled and transponder type indication flag is disabled:

Byte	Value	Description	Note
1	0x..	Transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		
...	0x..	Transponder code, last byte.	
...	0x..	Reading antenna: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

b3) case of one transponder present; antenna indication flag is disabled and transponder type indication flag is enabled:

Byte	Value	Description	Note
1	0x..	Transponder type: • 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC Class-1 Generation-2).	
2	0x..	Transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		
...	0x..	Transponder code, last byte.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

b4) case of one transponder present; antenna and transponder type indication flags are enabled:

Byte	Value	Description	Note
1	0x..	Transponder type: • 0x01: ISO 18000-6B; • 0x02: ISO 18000-6C (EPC Class-1 Generation-2).	
2	0x..	Transponder code, 1st byte.	The UID of the tag code consists of 8 bytes in the case of a transponder ISO 18000-6B and a maximum of 20 bytes in the case of a transponder ISO 18000-6C (EPC Class-1 Generation-2).
...	...		
...	0x..	Transponder code, last byte.	
...	0x..	Reading antenna: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

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

Byte	Value	Description	Note
1	0x07	Command code.	

The answer to this command is given in short form through the 'reply ack' / 'reply nak' flags (this command doesn't 'produce' data).

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.

Byte	Value	Description	Note
1	0x37	Command code.	
2	0x..	Relay to activate: • 0x01 -> Relay 1; • 0x02 -> Relay 2.	
3	0x..	Activation time: • 0x01 … 0x63 (1 … 99 seconds) -> in case of 'impulsive' relay activation; • 0x81 -> 'Continuous' activation; • 0x80 -> Deactivation.	

The answer to this command is given in short form through the 'reply ack' / 'reply nak' flags (this command doesn't 'produce' data).

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.

Byte	Value	Description	Note
1	0x36	Command code.	

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

Byte	Value	Description	Note
1	0x36	Command code.	
2	0x..	BLUEBOX status. High byte whose bits have the following meaning: • Bit 7…6: Not used; • Bit 5: RF status (0=off, 1=on); • Bit 4: 'Continuous' mode status (0=disabled, 1=enabled); • Bit 3…2: Not used; • Bit 1: Relay 2 status (1=active); • Bit 0: Relay 1 status (1=active);	
3	0x..	BLUEBOX status. Low byte whose bits have the following meaning: • Bit 7: Dip switch status 4 (1=off); • Bit 6: Dip switch status 3 (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=active); • Bit 0: Input 1 status (1=active).	

4.11 RF Deactivation

In 'continuous' mode, this command is used to suspend the activity of the RF antennas connected to the **BLUEBOX**; this feature can be useful for example when two **BLUEBOX** interfere because of their proximity allowing alternative activities (see also 'RF Activation' command).

Byte	Value	Description	Note
1	0x38	Command code.	

The answer to this command i given in short form through the 'reply ack' / 'reply nak' flags (this command doesn't 'produce' data).

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.

Byte	Value	Description	Note
1	0x39	Command code.	

The answer to this command is given in short form through the ‘reply ack’ / ‘reply nak’ flags (this command doesn’t ‘produce’ data).

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 tags that are present near the antennas. If the command can be executed, the response time is variable and depends on the number of enabled antennas and the activation time of each one.

Byte	Value	Description	Note
1	0x10	Command code.	

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

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

Byte	Value	Description	Note
1	0x10	Command code.	
2	0x00	Status Ok.	
3	0x..	1st transponder code, 1st byte.	The UID of the tag code consists of 8 bytes.
...	...		
9	0x..	1st transponder code, 8th byte.	
...	...		
...	0x..	m-th transponder code, 1st byte.	The UID of the tag code consists of 8 bytes.

Byte	Value	Description	Note
...	...		
...	0x..	m-th transponder code, 8th byte.	

If the answer consists of more than 14 bytes it will be transmitted with 2 or more data packets.

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

Byte	Value	Description	Note
1	0x10	Command code.	
2	0x00	Status Ok.	
3	0x..	1st transponder code, 1st byte.	The UID of the tag code consists of 8 bytes.
...	...		
9	0x..	1st transponder code, 8th byte.	
10	0x..	Reading antenna for 1st tag: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	
...	...		
...	0x..	m-th transponder code, 1st byte.	The UID of the tag code consists of 8 bytes.
...	...		
...	0x..	m-th transponder code, 8th byte.	
...	...	Reading antenna for m-th tag: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

b) if some error is occurred during the transaction

Byte	Value	Description	Note
1	0x10	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Note
1	0x10	Command code.	
2	0x01	No tag present.	

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

Byte	Value	Description	Note
1	0x10	Command code.	
2	0x..	Selection command code: • 0x00: GROUP_SELECT_EQ • 0x01: GROUP_SELECT_NE • 0x02: GROUP_SELECT_GT • 0x03: GROUP_SELECT_LT	
3	0x..	Address of the 1st byte of the tags to be compared with the bytes of the following data field according to the following mask field.	
4	0x..	Comparison mask. The single bits have the following meaning: • Bit 7: Enables /disables the comparison of byte 1 of the following data field with the 1st specified byte of the tags; • Bit 6: Enables /disables the comparison of byte 2 of the following data field with the 2nd specified byte of the tags; • ... • Bit 0: enables /disables the comparison of byte 8 of the following data field with the 8th specified byte of the tags g.	

5	0x..	1st byte to be compared to the data of the tags.	
...	...		
12		8th byte to be compared to the data of the tags.	

Note that when the short form command is used the following values are assumed: 0x01 for selection command code, 0x00 for address, 0xC0 for mask, 0x00 … 0x00 for data.

4.14 Read Data from B Type Tags

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

Byte	Value	Description	Note
1	0x11	Command code.	
2	0x..	1st transponder code, 1st byte.	The UID of the tag code consists of 8 bytes.
...	...		
9	0x..	1st transponder code, 8th byte.	
10	0x..	Start address of the first block to read.	
11	0x..	Number of blocks to read (1 … 8).	

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

a) if the addressed tag is present

Byte	Value	Description	Note
1	0x11	Command code.	
2	0x00	Status Ok	
3	0x..	1st block data, 1st byte.	Every data block is 2 bytes length.
...	...		

10	0x..	1st block data, 8th byte.	
...	...		
...	0x..	n-th block data, 1st byte	
...	0x..	n-th block data, 8th byte	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

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

Byte	Value	Description	Note
1	0x11	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Note
1	0x11	Command code.	
2	0x01	No tag present.	

4.15 Write Data to Type B Tags

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

Byte	Value	Description	Note
1	0x12	Command code.	
2	0x..	1st transponder code, 1st byte.	The UID of the tag code consists of 8 bytes.
...	...		
9	0x..	1st transponder code, 8th byte.	
10	0x..	Address of the first byte to read.	
11	0x..	Number of bytes to read (1 … 32).	

...	0x..	Data, 1st byte to write.	
...	...		
...	0x..	Data, n-th byte to write.	

If the command consists of more than 14 data bytes it must be transmitted with 2 or more data packets.

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

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

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

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

Byte	Value	Description	Note
1	0x12	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Note
1	0x12	Command code.	
2	0x01	No tag present.	

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.

Byte	Value	Description	Note
1	0x18	Command code.	

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

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

Byte	Value	Description	Note
1	0x18	Command code.	
2	0x00	Status Ok.	
3	0x..	1st transponder code, 1st byte.	The transponder code length is 20 bytes max.
...	...		
n+2	0x..	1st transponder code, n-th byte.	
...	...		
...	0x..	m-th transponder code, 1st byte.	The transponder code length is 20 bytes max.
...	...		
...	0x..	m-th transponder code, n-th byte.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

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

Byte	Value	Description	Note
1	0x18	Command code.	
2	0x00	Status Ok.	
3	0x..	1st transponder code, 1st byte.	The transponder code length is 20 bytes max.

Byte	Value	Description	Note
...	...		
n+2	0x..	1st transponder code, n-th byte.	
n+3	0x..	Reading antenna of the 1st tag: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	
...	...		
...	0x..	m-th transponder code, 1st byte.	The transponder code length is 20 bytes max.
...	...		
...	0x..	m-th transponder code, n-th byte.	
...	0x..	Reading antenna of the m-th tag: • 0x01 -> Antenna 1; • 0x02 -> Antenna 2; • 0x03 -> Antenna 3; • 0x04 -> Antenna 4.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

b) if some error is occurred during the transaction

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

c) if no tag is present

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

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.

Byte	Value	Description	Note
1	0x19	Command code.	
2	0x..	Transponder code, 1st byte.	The transponder code length is 20 bytes max.
...	...		
n+1	0x..	Transponder code, n-th byte.	
n+2	0x..	Tag access password, 1° byte.	The tag access password is 4 bytes length.
...	...		
n+5	0x..	Tag access password, 4° byte.	
n+6	0x..	Memory bank: • 0x00: Reserved. • 0x01: TID. • 0x02: EPC. • 0x03: User.	
n+7	0x..	Memory address of the 1st byte of the 1st memory block to read, 1st byte.	The memory address is 4 bytes length.
...	...		
n+10	0x..	Memory address of the 1st byte of the 1st memory block to read, 4th byte.	
n+11	0x..	Number of blocks to read (1 … 64).	

If the command consists of more than 14 data bytes it must be transmitted with 2 or more data packets.

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

a) if the addressed tag is present

Byte	Value	Description	Note

Byte	Value	Description	Note
1	0x19	Command code.	
2	0x00	Status Ok.	
3	0x..	1st data block, 1st byte.	Every data block is 2 bytes length.
4	0x..	1st data block, 2nd byte.	
...	...		
...	0x..	n-th data block, 1st byte.	
...	0x..	n-th data block, 2nd byte.	

If the answer consists of more than 14 data bytes it will be transmitted with 2 or more data packets.

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

Byte	Value	Description	Note
1	0x19	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Note
1	0x19	Command code.	
2	0x01	No tag present.	

4.18 Write Data into Type C Tags

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

Byte	Value	Description	Note
1	0x1A	Command code.	
2	0x..	Transponder code, 1st byte.	The transponder code length is 20 bytes max.

Byte	Value	Description	Note
...	...		
n+1	0x..	Transponder code, n-th byte.	
n+2	0x..	Tag access password, 1° byte.	The tag access password is 4 bytes length.
...	...		
n+5	0x..	Tag access password, 4° byte.	
n+6	0x..	Memory bank: • 0x00: Reserved. • 0x01: TID. • 0x02: EPC. • 0x03: User.	
n+7	0x..	Memory address of the 1st byte of the 1st memory block to write, 1st byte.	The memory address is 4 bytes length.
...	...		
n+10	0x..	Memory address of the 1st byte of the 1st memory block to write, 4th byte.	
n+11	0x..	Number of blocks to write (1 … 64).	
n+12	0x..	1st data block, 1st byte.	Every data block is 2 bytes length.
n+13	0x..	1st data block, 2nd byte.	
...	...		
...	0x..	n-th data block, 1st byte.	
...	0x..	n-th data block, 2nd byte.	

If the command consists of more than 14 data bytes it must be transmitted with 2 or more data packets.

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

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

Byte	Value	Description	Note

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

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

Byte	Value	Description	Note
1	0x1A	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Note
1	0x1A	Command code.	
2	0x01	No tag present.	

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.

Byte	Value	Description	Note
1	0x1B	Command code.	
2	0x..	Transponder code, 1st byte.	The transponder code length is 20 bytes max.
...	...		
n+1	0x..	Transponder code, n-th byte.	
n+2	0x..	Tag access password, 1° byte.	The tag access password is 4 bytes length.
...	...		
n+5	0x..	Tag access password, 4° byte.	
n+6	0x..	Kill password lock property. ASCII	

Byte	Value	Description	Note
		character: • 0x00: Accessible from all states; • 0x01: Permanently accessible from all states and may never be locked; • 0x02: Accessible only from the secured state; • 0x03: Not accessible from any state; • 0x04: No change.	
n+7	0x..	Access password lock property. ASCII character: • 0x00: Accessible from all states; • 0x01: Permanently accessible from all states and may never be locked; • 0x02: Accessible only from the secured state; • 0x03: Not accessible from any state; • 0x04: No change.	
n+8	0x..	EPC memory bank lock property. ASCII character: • 0x00: Writable from all states; • 0x01: Permanently writable from all states and may never be locked; • 0x02: Writable only from the secured state; • 0x03: Not writable from any state; • 0x04: No change.	
n+9	0x..	TID memory bank lock property. ASCII character: • 0x00: Writable from all states; • 0x01: Permanently writable from all states and may never be locked; • 0x02: Writable only from the secured state; • 0x03: Not writable from any state; • 0x04: No change.	

Byte	Value	Description	Note
n+10	0x..	User memory bank lock property. ASCII character: • 0x00: Writable from all states; • 0x01: Permanently writable from all states and may never be locked; • 0x02: Writable only from the secured state; • 0x03: Not writable from any state; • 0x04: No change.	

If the command consists of more than 14 data bytes it must be transmitted with 2 or more data packets.

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

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

Byte	Value	Description	Note
1	0x1B	Command code.	
2	0x00	Status Ok.	

b) if some error is occurred during the transaction

Byte	Value	Description	Note
1	0x1B	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Note
1	0x1B	Command code.	
2	0x01	No tag present.	

4.20 Kill Type C Tags

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

Byte	Value	Description	Note
1	0x1C	Command code.	
2	0x..	Transponder code, 1st byte.	The transponder code length is 20 bytes max.
...	...		
n+1	0x..	Transponder code, n-th byte.	
n+2	0x..	Tag kill password, 1° byte.	The tag kill password is 4 bytes length.
...	...		
n+5	0x..	Tag kill password, 4° byte.	

If the command consists of more than 14 data bytes it must be transmitted with 2 or more data packets.

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

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

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

b) if some error is occurred during the transaction

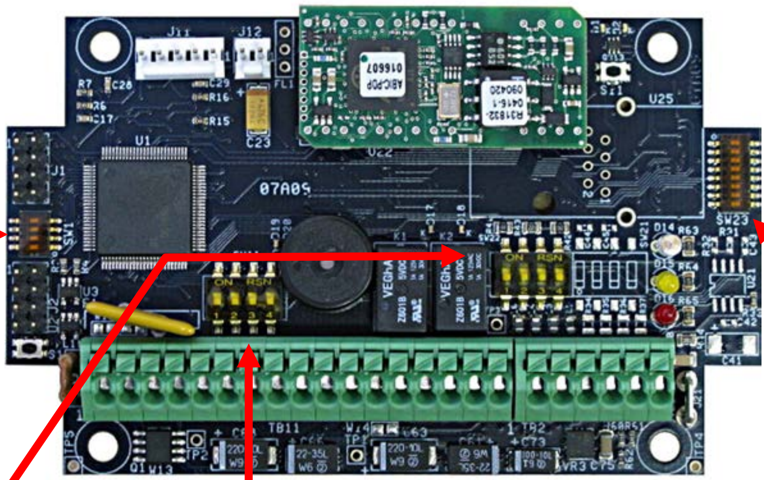
Byte	Value	Description	Note
1	0x1C	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Note
1	0x1C	Command code.	
2	0x01	No tag present.	

5 Hardware Settings: Dip Switch

SW1 – Configuration	
Dip 1	On: force 255, 19200, 8, n, 1.
Dip 2	On: enables ‘spontaneous’ mode on RS232 serial line.
Dip 3	On: enables ‘spontaneous’ mode on RS485 serial line.
Dip 4	Unused.

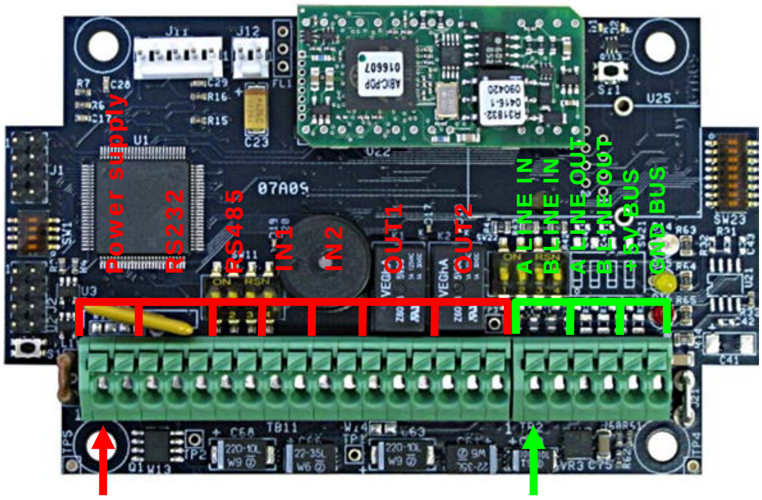


SW1 – Profibus Terminations		
Dip 1	Off	On
Dip 2	Off	On
Dip 3	Off	On
Dip 4	Off	On
Term.	Not ins.	Inserted

SW11 – RS485	
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	Serial line selection, On = RS485, Off = RS232.

SW2 – Profibus Address									
Dip 1	On	Off	On	Off	…	Off	On	Off	On
Dip 2	On	On	Off	Off	…	Off	On	On	Off
Dip 3	On	On	On	On	…	On	Off	Off	Off
Dip 4	On	On	On	On	…	Off	Off	Off	Off
Dip 5	On	On	On	On	…	Off	Off	Off	Off
Dip 6	On	On	On	On	…	Off	Off	Off	Off
Dip 7	On	On	On	On	…	Off	Off	Off	Off
Dip 8	On	On	On	On	…	On	On	On	On
Address	0	1	2	3	…	123	124	125	126

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

Profibus connections:

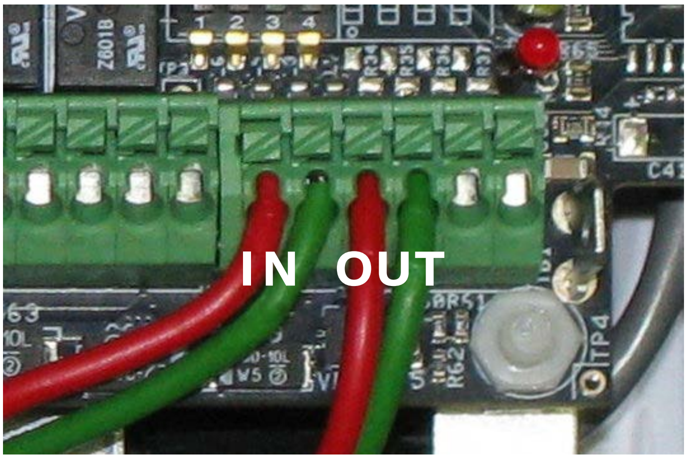
Pin	No.	Description
A LINE IN	1	RXD/TXD-N
B LINE IN	2	RXD/TXD-P
A LINE OUT	3	RXD/TXD-N
B LINE OUT	4	RXD/TXD-P
+5V BUS	5	+5V
GND BUS	6	GND

Notes:

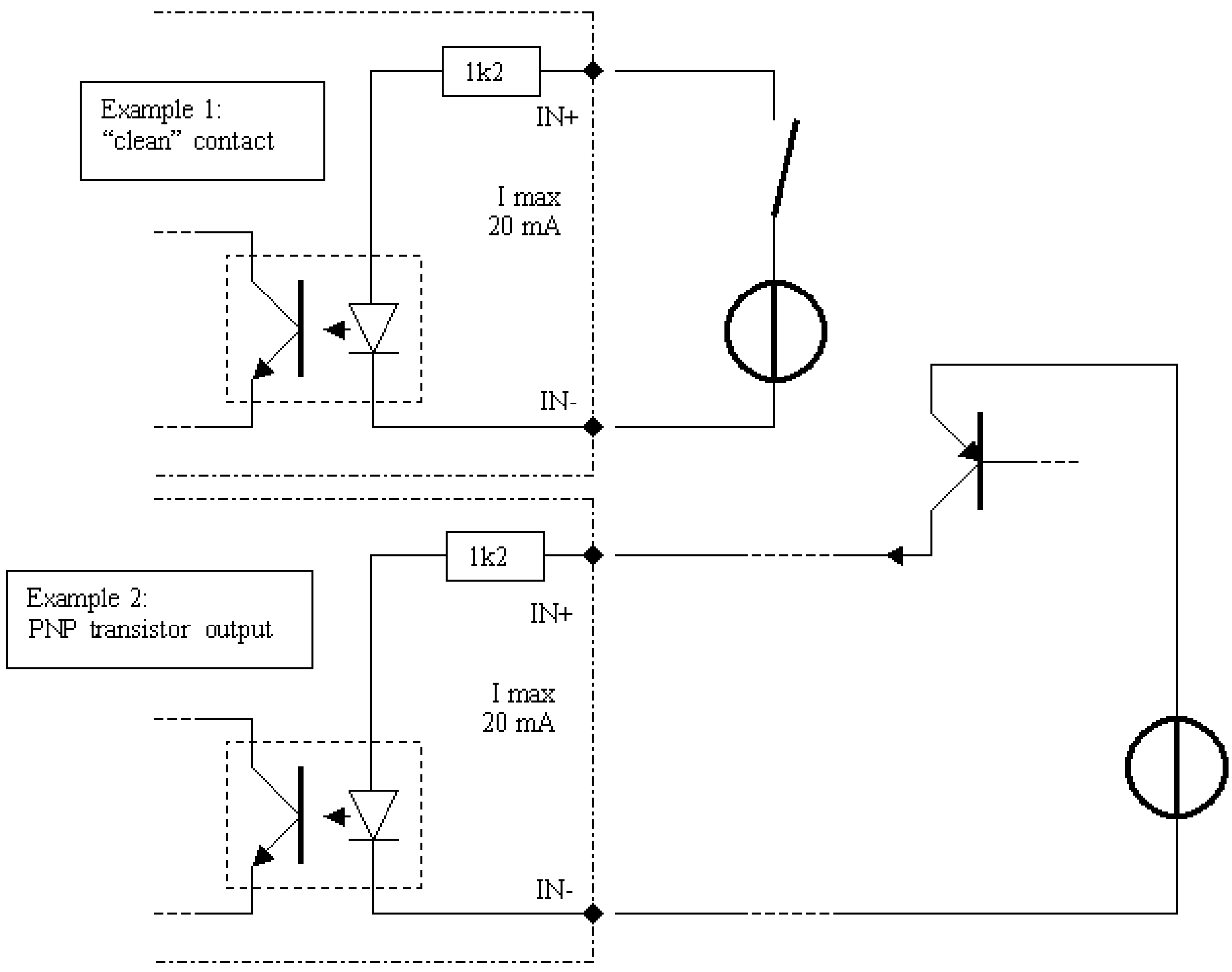
- +5V BUS e GND BUS can be used for external bus termination, max current available 100 mA;
- using standard Profibus cables, the A line is characterized by the green wire while the B line is characterized by the red wire;
- in order to maintain shield continuity (if shield is used), the shield of the incoming cable must be connected to the shield of the outgoing cable.

Example of wiring:

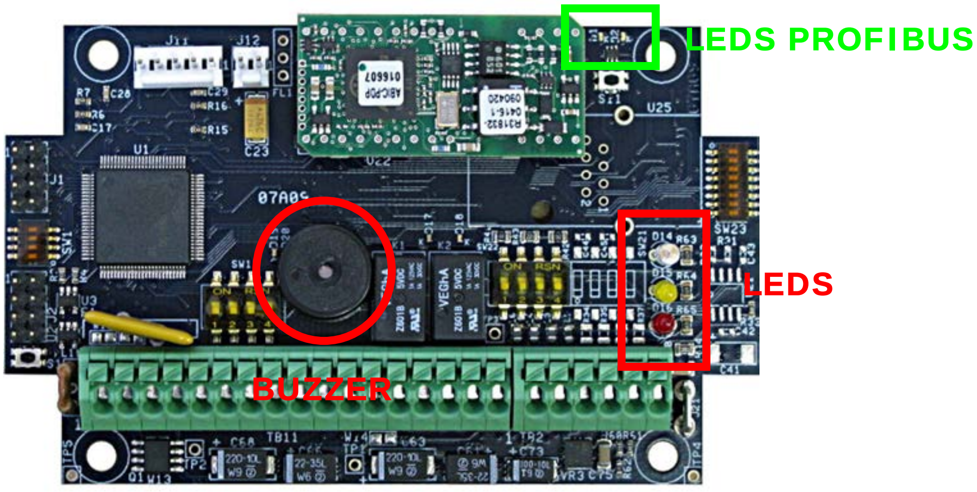
Example of Profibus network wiring using standard cable: the following picture depicts the case of an ‘intermediate’ device in the Profibus network, in case of a final device, it is necessary to enable the local bus termination (refer to chapter 5) or to realize external bus termination using +5V BUS e GND BUS.



7 Digital inputs wiring examples



8 Status Indications: Led and Buzzer



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

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.

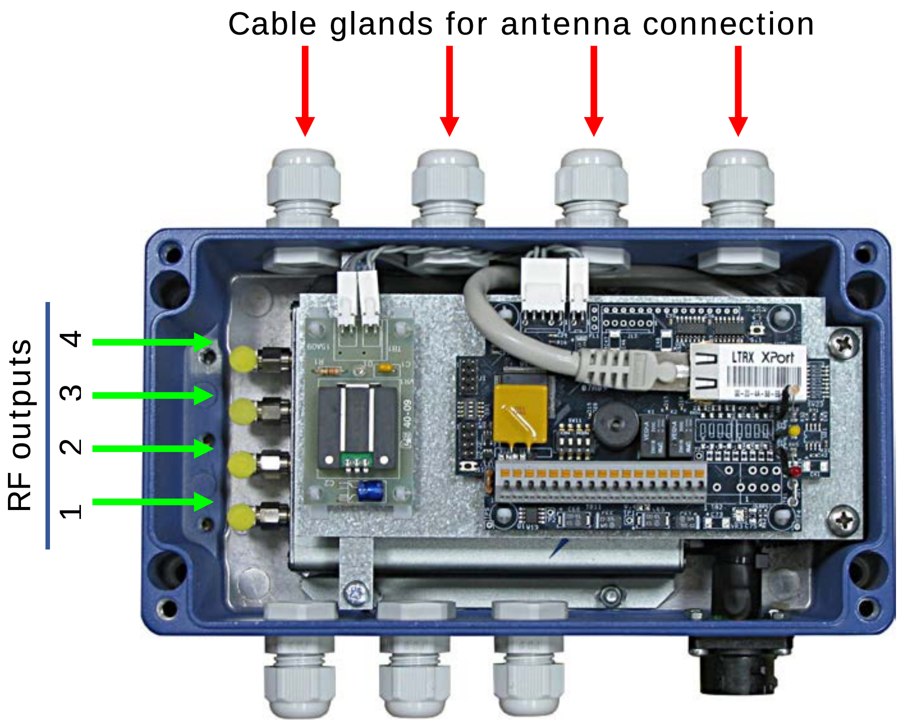
LEDs and buzzer behaviour during the **BLUEBOX** start up: LEDs will be switched on during the whole initialization phase (about 5 seconds), at the end of the initialization phase, the buzzer will be switched on for 0.5 seconds, afterwards the LEDs and the buzzer will work as described above.

Led Profibus: see the table below.

Led	Status	Description
Green (D2)	Off On Flashing	Node off line Node in ‘data exchange’ mode Node in ‘clear’ mode
Red (D1)	Off On Flashing	No errors Error during initialization Configuration error

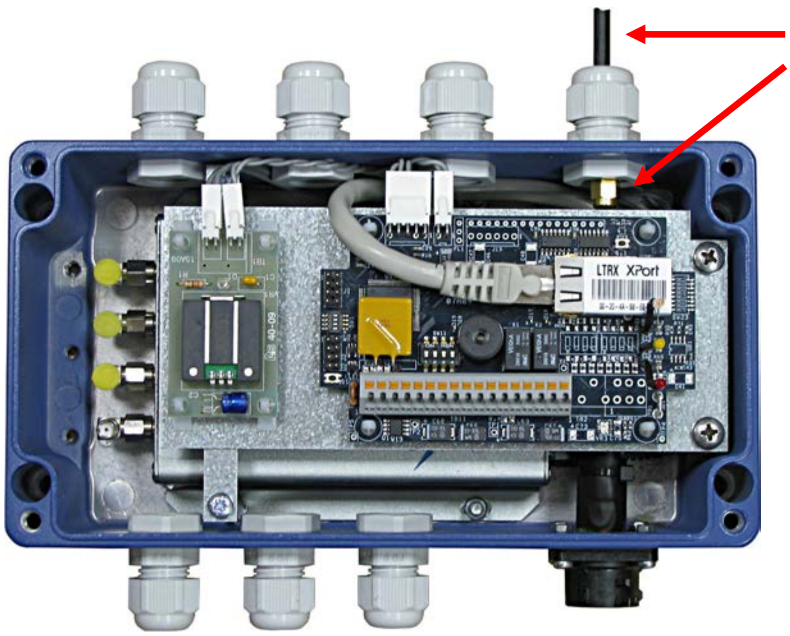
9 Antennas

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).

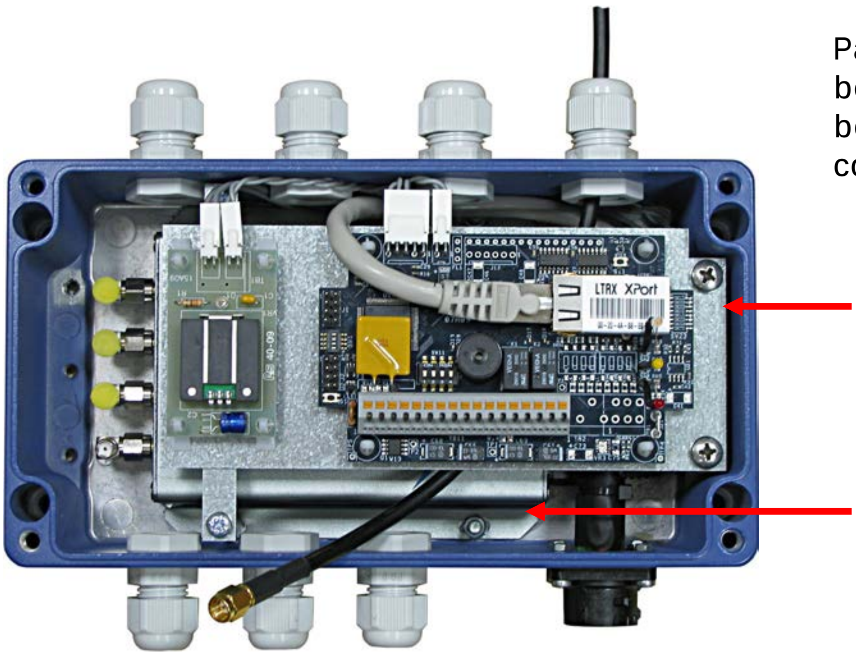


The read range of an RFID system always depends on various factors like antenna size, tag size, tag IC type, orientation between tag and reader antenna, position of the tag versus the reader antenna, noise environment, metallic environment, RF power, etc. Therefore all data about read ranges can only 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**.



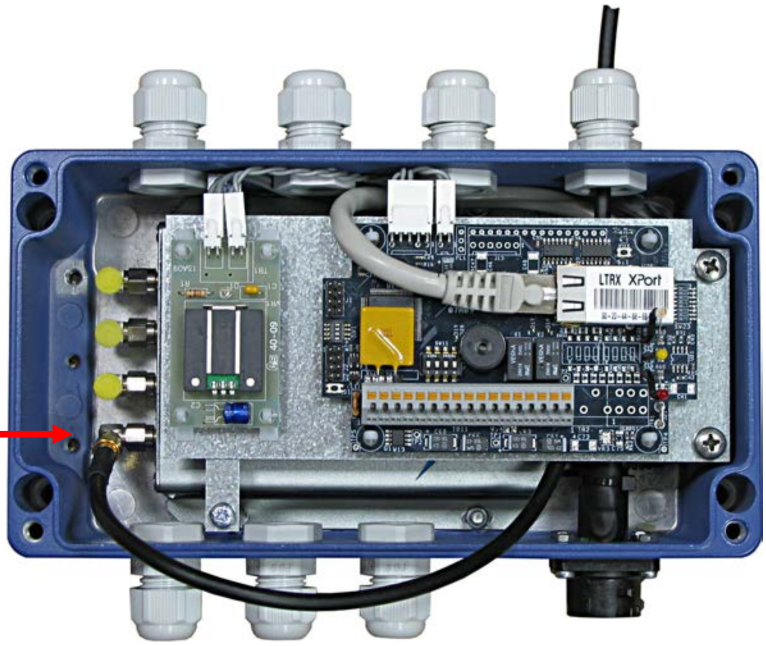
Pass the antenna cable through the cable gland



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

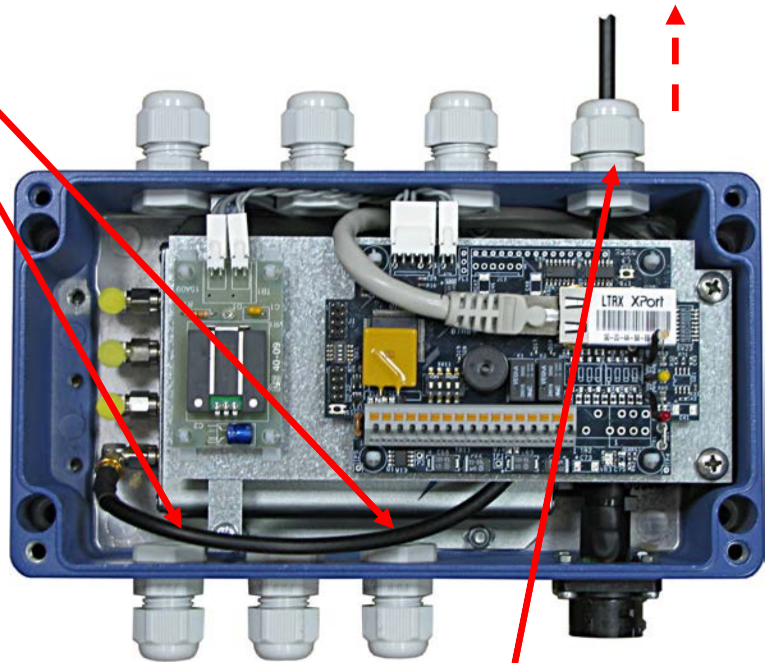
Electronic board support

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

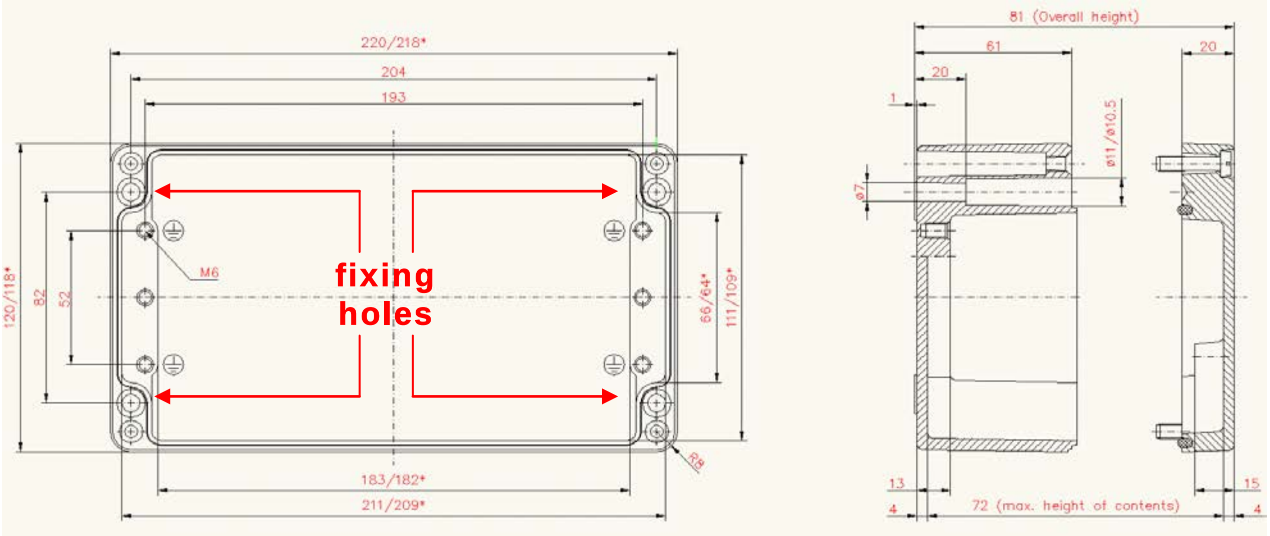
Operating frequency	860 MHz – 960 MHz
Polarisation	Linear
Impedance	50 Ohm
Connector	SMA, male

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 Document Revision History

Date	Revision	Firmware	Description
19/11/09	1.00	From 1.00	First 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.
17/06/10	1.03	From 1.06	Some corrections to the document.
25/01/11	1.04	From 1.11	Changed the introduction section of the document. Changed the technical specifications and added the antenna power, operating frequency and reading distance as operating features. 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. Changed the read firmware version command by adding the auxiliary reader (the RFID front-end connected to the **BLUEBOX**) firmware version. 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. Added the document revision history section (this section). New document formatting.
11/03/11	1.05	From 1.11	Corrected the Profibus address dip switch settings table in section 5.

A. Supported Transponders

Supported transponders by **BLUEBOX** are:

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

A.1 ISO 18000-6B

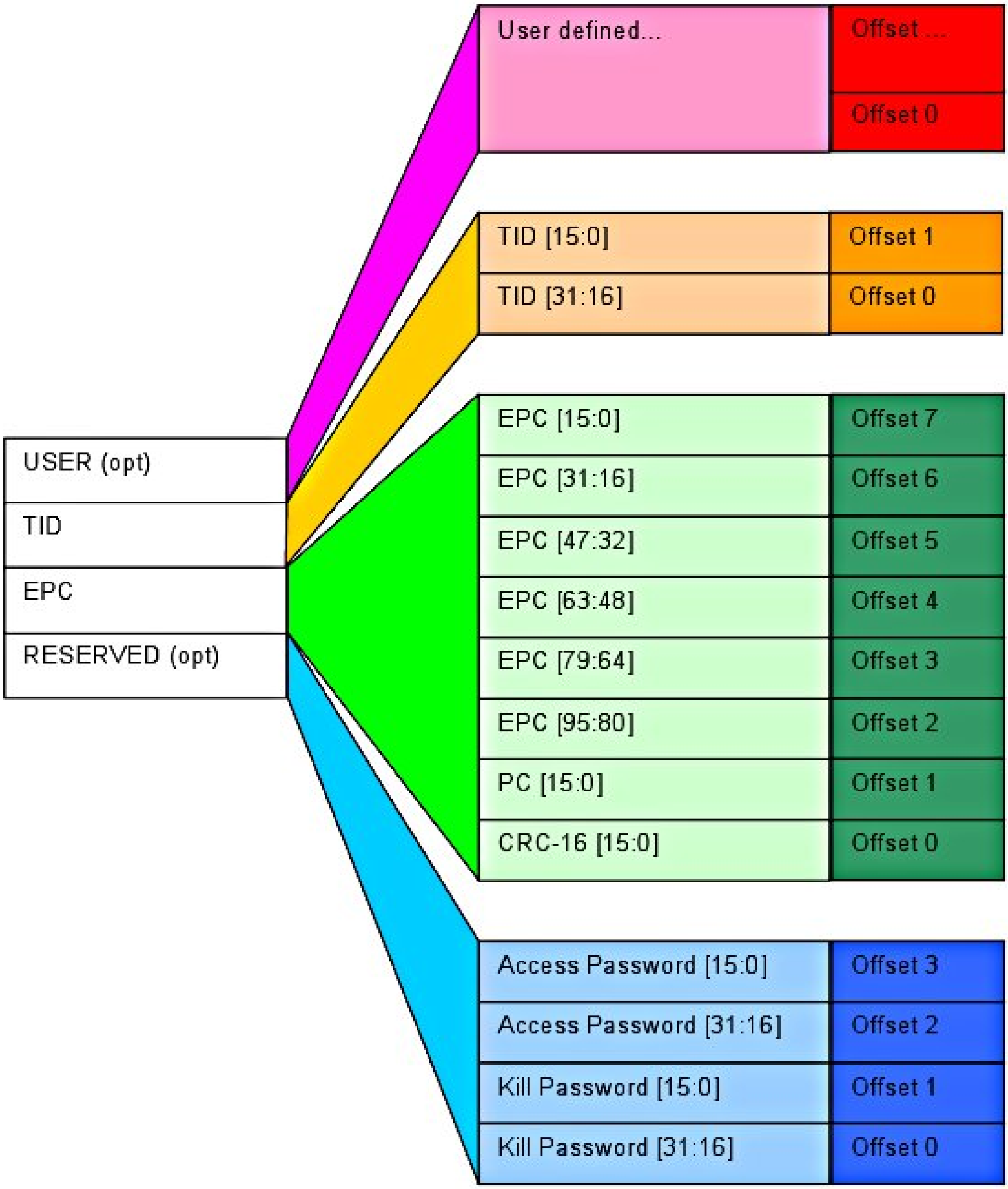
ISO 18000-6B transponders have a memory divided into blocks of 1 byte each, the first 8 bytes are the UID code of the transponder and are not changeable. The memory size is variable and depends on the transponder type, the maximum size is 256 bytes (2 kbits).

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.

B. '.GSD' File

```
;==============================================================
; Profibus Device Database of HMS Industrial Networks.
;
; Model :       Anybus-IC DP-V0
; Description : Anybus-IC DP-V0 slave
; Language :    English
; Date :        16 March 2007
; Author :      HMS Industrial Networks
;
; Revision log:
;
; 2006-10-02:   Updated revision information. Cleared old
;               revision history.
; 2007-03-16:   Updated revision information.
; 2007-03-16:   Updated SW revision information.
; 2007-09-18:   Updated SW revision information.
; 2008-01-16:   Updated Model_Name.
; 2008-03-27:   Updated SW revision information.
; 2008-05-15:   Updated SW revision information.
; 2008-06-12:   Updated SW revision information.
; 2008-10-23:   Updated SW revision information and added Max_User_Prm_Data_Len.
; 2008-11-20:   Updated SW revision information and removed Max_User_Prm_Data_Len.
;
;==============================================================

#Profibus_DP

GSD_Revision        = 3

; Device identification
Vendor_Name         = "HMS Industrial Networks"
Model_Name          = "Anybus-IC PDP"
Revision            = "Version 2.12"
Ident_Number        = 0x1810
Protocol_Ident      = 0             ; DP protocol
Station_Type        = 0             ; Slave device
FMS_supp            = 0             ; FMS not supported
Hardware_Release    = "Version 2.3"
Software_Release    = "Version 2.12"

;Used bitmap
Bitmap_Device = "ABIC_DE"
Bitmap_Diag   = "ABIC_DI"
Bitmap_SF     = "ABIC_SF"

; Supported baudrates
9.6_supp            = 1
19.2_supp           = 1
45.45_supp          = 1
93.75_supp          = 1
187.5_supp          = 1
500_supp            = 1
1.5M_supp           = 1
3M_supp             = 1
6M_supp             = 1
12M_supp            = 1

; Maximum responder time for supported baudrates
MaxTsdr_9.6         = 15
MaxTsdr_19.2        = 15
MaxTsdr_45.45       = 15
MaxTsdr_93.75       = 15
MaxTsdr_187.5       = 15
```

```
MaxTsdr_500         = 15
MaxTsdr_1.5M        = 25
MaxTsdr_3M          = 50
MaxTsdr_6M          = 100
MaxTsdr_12M         = 200

; Supported hardware features
Redundancy          = 0       ; not supported
Repeater_Ctrl_Sig   = 2       ; TTL
24V_Pins            = 0       ; not connected
Implementation_Type = "NP30"

; Supported DP features
Freeze_Mode_supp    = 1       ; supported
Sync_Mode_supp      = 1       ; supported
Auto_Baud_supp      = 1       ; supported
Set_Slave_Add_supp  = 1       ; supported

; Maximum polling frequency
Min_Slave_Intervall = 1       ; 100 us

; Maximum supported sizes
Modular_Station     = 1       ; modular
Max_Module          = 24
Max_Input_Len       = 144
Max_Output_Len      = 144
Max_Data_Len        = 288
Modul_Offset        = 1

Fail_Safe           = 1       ; Data telegram without data in state CLEAR accepted

Slave_Family        = 0
Max_Diag_Data_Len   = 6

; Definition of modules
Module = "IN/OUT:   1 Byte" 0x30
1
EndModule
;
Module = "IN/OUT:   2 Byte ( 1 word)" 0x70
2
EndModule
;
Module = "IN/OUT:   4 Byte ( 2 word)" 0x71
3
EndModule
;
Module = "IN/OUT:   8 Byte ( 4 word)" 0x73
4
EndModule
;
Module = "IN/OUT:  16 Byte ( 8 word)" 0x77
5
EndModule
;
Module = "IN/OUT:  32 Byte (16 word)" 0x7F
6
EndModule
;
Module = "INPUT:    1 Byte" 0x10
7
EndModule
;
Module = "INPUT:    2 Byte ( 1 word)" 0x50
8
EndModule
;
Module = "INPUT:    4 Byte ( 2 word)" 0x51
9
```

```
EndModule
;
Module = "INPUT:   8 Byte ( 4 word)" 0x53
10
EndModule
;
Module = "INPUT:  16 Byte ( 8 word)" 0x57
11
EndModule
;
Module = "INPUT:  32 Byte (16 word)" 0x5F
12
EndModule
;
Module = "OUTPUT:   1 Byte" 0x20
13
EndModule
;
Module = "OUTPUT:   2 Byte ( 1 word)" 0x60
14
EndModule
;
Module = "OUTPUT:   4 Byte ( 2 word)" 0x61
15
EndModule
;
Module = "OUTPUT:   8 Byte ( 4 word)" 0x63
16
EndModule
;
Module = "OUTPUT:  16 Byte ( 8 word)" 0x67
17
EndModule
;
Module = "OUTPUT:  32 Byte (16 word)" 0x6F
18
EndModule
```