BLUEBOX RFID System

COMMUNICATION PROTOCOL

Serial / Ethernet

Preface

Kronotech S.r.l. (Kronotech) reserves the right to make changes to its products or services or to discontinue any product or service at any time without notice. Kronotech 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, Kronotech assumes no liability and is not responsible for customer applications or product or software design or performance relating to systems or applications incorporating Kronotech products. In addition, Kronotech 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 Kronotech. Kronotech 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. Kronotech 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. Kronotech 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 subsidaries in any country. This document may be downloaded onto a computer, stored and duplicated as necessary to support the use of the related Kronotech products. Any other type of duplication, circulation or storage on data carriers in any manner not authorized by Kronotech 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.

This manual applies to the following devices:

Description:	**Order Number:**	
Read / write LF 125kHz RFID OEM device with one external antenna. 5Vdc power supply, serial TTL 0-5V communication interface.	1021L	
Read / write LF 125kHz RFID OEM device with one external antenna. 5Vdc power supply, serial RS232 communication interface.	1031L	
Read / write LF 125kHz RFID OEM device with one external antenna. 12Vdc power supply, serial RS232 communication interface.	1041L	
Read / write HF 13.56MHz RFID OEM device with one external antenna. 5Vdc power supply, serial TTL 0-5V communication interface.	1021H	
Read / write HF 13.56MHz RFID OEM device with one external antenna. 5Vdc power supply, serial RS232 communication interface.	1031H	
Read / write HF 13.56MHz RFID OEM device with one external antenna. 12Vdc power supply, serial RS232 communication interface.	1041H	
Read / write HF 13.56MHz RFID OEM device with one external antenna. 5Vdc power supply, serial TTL 0-5V communication interface.	1051H	
Read / write HF 13.56MHz RFID OEM device with one external antenna. 5Vdc power supply, serial TTL 0-5V communication interface.	1021N	

Description:	Order Number:
Read / write HF 13.56MHz RFID OEM device with one external antenna. 5Vdc power supply, serial RS232 communication interface.	1031N
Read / write 100mW UHF RFID OEM device with one external antenna. Serial TTL 0-5V communication interface. EU1 (865 MHz … 868 MHz) version.	1021U
Read / write 100mW UHF RFID OEM device with one external antenna. Serial TTL 0-5V communication interface. FCC (902 MHz … 928 MHz) version.	1021U-FCC
Read / write 100mW UHF RFID OEM device with one external antenna. Serial TTL 0-5V communication interface. Brazil (902 MHz … 928 MHz) version.	1021U-BRA
Read / write 100mW UHF RFID OEM device with one external antenna with soldering strips. Serial TTL 0-5V communication interface. EU1 (865 MHz … 868 MHz) version.	1021U-S
Read / write 100mW UHF RFID OEM device with one external antenna with soldering strips. Serial TTL 0-5V communication interface. FCC (902 MHz … 928 MHz) version.	1021U-S-FCC
Read / write 100mW UHF RFID OEM device with one external antenna with soldering strips. Serial TTL 0-5V communication interface. Brazil (902 MHz … 928 MHz) version.	1021U-S-BRA
Read / write 100mW UHF RFID OEM device with one external antenna. Serial RS232 communication interface. EU1 (865 MHz … 868 MHz) version.	1031U
Read / write 100mW UHF RFID OEM device with one external antenna. Serial RS232 communication interface. FCC (902 MHz … 928 MHz) version.	1031U-FCC
Read / write 100mW UHF RFID OEM device with one external antenna. Serial RS232 communication interface. Brazil (902 MHz … 928 MHz) version.	1031U-BRA
Read / write 100mW UHF RFID OEM device with one external antenna with soldering strips. Serial RS232 communication interface. EU1 (865 MHz … 868 MHz) version.	1031U-S
Read / write 100mW UHF RFID OEM device with one external antenna with soldering strips. Serial RS232 communication interface. FCC (902 MHz … 928 MHz) version.	1031U-S-FCC

Description:	Order Number:
Read / write 100mW UHF RFID OEM device with one external antenna with soldering strips. Serial RS232 communication interface. Brazil (902 MHz … 928 MHz) version.	1031U-S-BRA
Read / write 500mW UHF RFID OEM device with one external antenna. Serial TTL 0-5V communication interface. EU1 (865 MHz … 868 MHz) version.	1061U
Read / write 500mW UHF RFID OEM device with one external antenna. Serial TTL 0-5V communication interface. FCC (902 MHz … 928 MHz) version.	1061U-FCC
Read / write 500mW UHF RFID OEM device with one external antenna. Serial TTL 0-5V communication interface. Brazil (902 MHz … 928 MHz) version.	1061U-BRA
Read / write 500mW UHF RFID OEM device with one external antenna with soldering strips. Serial TTL 0-5V communication interface. EU1 (865 MHz … 868 MHz) version.	1061U-S
Read / write 500mW UHF RFID OEM device with one external antenna with soldering strips. Serial TTL 0-5V communication interface. FCC (902 MHz … 928 MHz) version.	1061U-S-FCC
Read / write 500mW UHF RFID OEM device with one external antenna with soldering strips. Serial TTL 0-5V communication interface. Brazil (902 MHz … 928 MHz) version.	1061U-S-BRA
Read / write 500mW UHF RFID OEM device with one external antenna. Serial RS232 communication interface. EU1 (865 MHz … 868 MHz) version.	1071U
Read / write 500mW UHF RFID OEM device with one external antenna. Serial RS232 communication interface. FCC (902 MHz … 928 MHz) version.	1071U-FCC
Read / write 500mW UHF RFID OEM device with one external antenna. Serial RS232 communication interface. Brazil (902 MHz … 928 MHz) version.	1071U-BRA
Read / write 500mW UHF RFID OEM device with one external antenna with soldering strips. Serial RS232 communication interface. EU1 (865 MHz … 868 MHz) version.	1071U-S

Description:	Order Number:	
Read / write 500mW UHF RFID OEM device with one external antenna with soldering strips. Serial RS232 communication interface. FCC (902 MHz … 928 MHz) version.	1071U-S-FCC	
Read / write 500mW UHF RFID OEM device with one external antenna with soldering strips. Serial RS232 communication interface. Brazil (902 MHz … 928 MHz) version.	1071U-S-BRA	
Read / write 1W UHF RFID OEM device with up to 4 external antennas. Serial TTL 0-5V communication interface. EU1 (865 MHz … 868 MHz) version.	1041U	
Read / write 1W UHF RFID OEM device with up to 4 external antennas. Serial TTL 0-5V communication interface. FCC (902 MHz … 928 MHz) version.	1041U-FCC	
Read / write 1W UHF RFID OEM device with up to 4 external antennas. Serial TTL 0-5V communication interface. Brazil (902 MHz … 928 MHz) version.	1041U-BRA	
Read / write 1W UHF RFID OEM device with up to 4 external antennas. Serial RS232 communication interface. EU1 (865 MHz … 868 MHz) version.	1051U	
Read / write 1W UHF RFID OEM device with up to 4 external antennas. Serial RS232 communication interface. FCC (902 MHz … 928 MHz) version.	1051U-FCC	
Read / write 1W UHF RFID OEM device with up to 4 external antennas. Serial RS232 communication interface. Brazil (902 MHz … 928 MHz) version.	1051U-BRA	
Read / write 125 kHz RFID device with integrated antenna. USB communication interface (Virtual COM + HID Keyboard).	3122L	
Read / write 13.56 MHz RFID device with integrated antenna. USB communication interface (Virtual COM + HID Keyboard).	3122H	
Read / write 13.56 MHz RFID device with integrated antenna. USB communication interface (Virtual COM + HID Keyboard).	3122N	
Read / write UHF RFID device with integrated antenna. USB (Virtual COM + HID Keyboard) communication interface. EU1 (865 MHz … 868 MHz) version.	3122U	
Read / write 125 kHz RFID panel reader with integrated antenna. Serial RS232 communication interface.	3221L	

Description:	Order Number:
Read / write 125 kHz RFID panel reader with integrated antenna. USB (Virtual COM) communication interface.	3222L
Read / write 125 kHz RFID panel reader with integrated antenna. USB (Virtual COM and HID Keyboard) communication interface.	3223L
Read / write 13.56 MHz RFID panel reader with integrated antenna. Serial RS232 communication interface.	3221N
Read / write 13.56 MHz RFID panel reader with integrated antenna. USB (Virtual COM) communication interface.	3222N
Read / write 13.56 MHz RFID panel reader with integrated antenna. USB (Virtual COM and HID Keyboard) communication interface.	3223N
Read / write 125 kHz RFID device with integrated antenna. Serial RS232/RS485 communication interface.	5121L
Read / write 125 kHz RFID device with one external antenna. Serial RS232/RS485 communication interface.	5131L
Read / write 13.56 MHz RFID device with integrated antenna. Serial RS232/RS485 communication interface.	5121H
Read / write 13.56 MHz RFID device with one external antenna. Serial RS232/RS485 communication interface.	5131H
Read / write UHF RFID device with integrated antenna. Serial RS232/RS485 communication interface. EU1 (865 MHz … 868 MHz) version.	5121U
Read / write UHF RFID device with one external antenna. Serial RS232/RS485 communication interface. EU1 (865 MHz … 868 MHz) version.	5131U
Read / write LF RFID device with integrated antenna. Serial RS232/RS485 communication interface.	5221L
Read / write LF RFID device with one external antenna. Serial RS232/RS485 communication interface.	5231L
Read / write LF RFID device with two external antennas. Serial RS232/RS485 communication interface.	5241L
Read / write LF RFID device with integrated antenna. Ethernet 10-100M communication interface.	5222L

NO PRODUCT IMAGE

NO PRODUCT IMAGE

Description:	Order Number:	
Read / write LF RFID device with one external antenna. Ethernet 10-100M communication interface.	5232L	
Read / write LF RFID device with two external antennas. Ethernet 10-100M communication interface.	5242L	
Read / write HF RFID device with integrated antenna. Serial RS232/RS485 communication interface.	5221H	NO PRODUCT IMAGE
Read / write HF RFID device with one external antenna. Serial RS232/RS485 communication interface.	5231H	
Read / write HF RFID device with two external antennas. Serial RS232/RS485 communication interface.	5241H	
Read / write HF RFID device with integrated antenna. Ethernet 10-100M communication interface.	5222H	NO PRODUCT IMAGE
Read / write HF RFID device with one external antenna. Ethernet 10-100M communication interface.	5232H	
Read / write HF RFID device with two external antennas. Ethernet 10-100M communication interface.	5242H	
Read / write UHF RFID device with integrated antenna. Serial RS232/RS485 communication interface. EU1 (865 MHz … 868 MHz) version.	5221U-S	NO PRODUCT IMAGE
Read / write UHF RFID device with one external antenna. Serial RS232/RS485 communication interface. EU1 (865 MHz … 868 MHz) version.	5237U-S	
Read / write UHF RFID device with up to four external antennas. Serial RS232/RS485 communication interface. EU1 (865 MHz … 868 MHz) version.	5231U	
Read / write UHF RFID device with integrated antenna. Ethernet 10-100M communication interface. EU1 (865 MHz … 868 MHz) version.	5222U-S	NO PRODUCT IMAGE
Read / write UHF RFID device with one external antenna. Ethernet 10-100M communication interface. EU1 (865 MHz … 868 MHz) version.	5238U-S	
Read / write UHF RFID device with up to four external antennas. Ethernet 10-100M communication interface. EU1 (865 MHz … 868 MHz) version.	5232U	
Read / write RFID UHF device with integrated antenna. Serial RS232 communication interface. EU (865 MHz … 868 MHz) version.	5224U	

Description:	Order Number:
Read / write RFID UHF device with integrated antenna. Serial RS485 communication interface. EU (865 MHz … 868 MHz) version.	5225U
Mid Range read / write UHF RFID (EU) device with integrated antenna. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (Amphenol RJF connector). EU1 (865 MHz … 868 MHz) version.	5325U
Mid Range read / write UHF RFID (EU) device with integrated antenna and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (Amphenol RJF connector). EU1 (865 MHz … 868 MHz) version.	5325U-RTC
Mid Range read / write UHF RFID (US) device with integrated antenna. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (Amphenol RJF connector). FCC (902 MHz … 928 MHz) version.	5325U-FCC
Mid Range read / write UHF RFID (US) device with integrated antenna and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (Amphenol RJF connector). FCC (902 MHz … 928 MHz) version.	5325U-RTC-FCC
Long Range read / write UHF RFID device with 1 external antenna. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (Amphenol RJF connector). EU1 (865 MHz … 868 MHz) version.	5335U
Long Range read / write UHF RFID device with 1 external antenna and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (Amphenol RJF connector). EU1 (865 MHz … 868 MHz) version.	5335U-RTC
Long Range read / write UHF RFID device with up to 2 external antennas. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (Amphenol RJF connector). EU1 (865 MHz … 868 MHz) version.	5345U
Long Range read / write UHF RFID device with up to 2 external antennas and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (Amphenol RJF connector). EU1 (865 MHz … 868 MHz) version.	5345U-RTC

Description:	Order Number:
Mid Range read / write UHF RFID (EU) device with integrated antenna. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5326U 
Mid Range read / write UHF RFID (EU) device with integrated antenna and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5326U-RTC
Mid Range read / write UHF RFID (US) device with integrated antenna. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). FCC (902 MHz … 928 MHz) version.	5326U-FCC
Mid Range read / write UHF RFID (US) device with integrated antenna and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). FCC (902 MHz … 928 MHz) version.	5326U-RTC-FCC
Long Range read / write UHF RFID device with 1 external antenna. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5336U
Long Range read / write UHF RFID device with 1 external antenna and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5336U-RTC
Long Range read / write UHF RFID device with up to 2 external antennas. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5346U
Long Range read / write UHF RFID device with up to 2 external antennas and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5346U-RTC

Description:	Order Number:
Long Range read / write UHF RFID device with up to 2 external antennas. With Bluetooth communication interface and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5346U-BL
Mid Range read / write UHF RFID (EU) device with integrated antenna. With serial RS232/RS485 communication interface and Wiegand communication interface. EU1 (865 MHz … 868 MHz) version.	5327U
Mid Range read / write UHF RFID (US) device with integrated antenna. With serial RS232/RS485 communication interface and Wiegand communication interface. FCC (902 MHz … 928 MHz) version.	5327U-FCC
Long Range read / write UHF RFID device with 1 external antenna. With serial RS232/RS485 communication interface and Wiegand communication interface. EU1 (865 MHz … 868 MHz) version.	5337U
Long Range read / write UHF RFID device with up to 2 external antennas. With serial RS232/RS485 communication interface and Wiegand communication interface. EU1 (865 MHz … 868 MHz) version.	5347U
Mid Range read / write UHF RFID (EU) device with integrated antenna. With CAN bus and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5328U
Mid Range read / write UHF RFID (EU) device with integrated antenna and RTC (Real Time Clock). With CAN bus and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5328U-RTC
Mid Range read / write UHF RFID (US) device with integrated antenna. With CAN bus and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). FCC (902 MHz … 928 MHz) version.	5328U-FCC
Mid Range read / write UHF RFID (US) device with integrated antenna and RTC (Real Time Clock). With CAN bus and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). FCC (902 MHz … 928 MHz) version.	5328U-RTC-FCC
Long Range read / write UHF RFID device with 1 external antenna. With CAN bus and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5338U

Description:	Order Number:
Long Range read / write UHF RFID device with 1 external antenna with RTC (Real Time Clock). With CAN bus and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5338U-RTC
Long Range read / write UHF RFID device with up to 2 external antennas. With CAN bus and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5348U
Long Range read / write UHF RFID device with up to 2 external antennas with RTC (Real Time Clock). With CAN bus and Ethernet 10-100M communication interface (M12 4 poles D-coded female connector). EU1 (865 MHz … 868 MHz) version.	5348U-RTC
Mid Range read / write UHF RFID device with integrated antenna. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface. Grey white (RAL 9002) case color. EU1 (865 MHz … 868 MHz) version.	5426U NO PRODUCT IMAGE
Mid Range read / write UHF RFID device with integrated antenna and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface. Grey white (RAL 9002) case color. EU1 (865 MHz … 868 MHz) version.	5426U-RTC
Mid Range read / write UHF RFID device with integrated antenna. With serial RS232/RS485 communication interface and Wiegand communication interface. Grey white (RAL 9002) case color. EU1 (865 MHz … 868 MHz) version.	5427U
Mid Range read / write UHF RFID device with integrated antenna. With CAN bus and Ethernet 10-100M communication interface. Grey white (RAL 9002) case color. EU1 (865 MHz … 868 MHz) version.	5428U
Mid Range read / write UHF RFID device with integrated antenna and RTC (Real Time Clock). With CAN bus and Ethernet 10-100M communication interface. Grey white (RAL 9002) case color. EU1 (865 MHz … 868 MHz) version.	5428U-RTC
Mid Range read / write UHF RFID device with integrated antenna. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz … 868 MHz) version.	5426U-G NO PRODUCT IMAGE

Description:	Order Number:
Mid Range read / write UHF RFID device with integrated antenna and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz … 868 MHz) version.	5426U-RTC-G
Mid range read / write UHF RFID device with integrated antenna. With serial RS232/RS485 and Wiegand communication interface. Grey (RAL 7045) case color. EU1 (865 MHz … 868 MHz) version.	5427U-G
Mid Range read / write UHF RFID device with integrated antenna. With CAN bus and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz … 868 MHz) version.	5428-G
Mid Range read / write UHF RFID device with integrated antenna and RTC (Real Time Clock). With CAN bus and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz … 868 MHz) version.	5428-RTC-G
Long range read / write UHF RFID device with integrated antenna. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface. Grey white (RAL 9002) case color. EU1 (865 MHz … 868 MHz) version.	5526U
Long range read / write UHF RFID device with integrated antenna and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface. Grey white (RAL 9002) case color. EU1 (865 MHz … 868 MHz) version.	5526U-RTC
Long range read / write UHF RFID device with integrated antenna. With serial RS232/RS485 communication interface and Wiegand communication interface. Grey white (RAL 9002) case color. EU1 (865 MHz … 868 MHz) version.	5527U
Long range read / write UHF RFID device with integrated antenna. With CAN bus and Ethernet 10-100M communication interface. Grey white (RAL 9002) case color. EU1 (865 MHz … 868 MHz) version.	5528U
Long range read / write UHF RFID device with integrated antenna and RTC (Real Time Clock). With CAN bus and Ethernet 10-100M communication interface. Grey white (RAL 9002) case color. EU1 (865 MHz … 868 MHz) version.	5528U-RTC
Long range read / write UHF RFID device with	5526U-G

NO PRODUCT IMAGE

Description:	Order Number:
integrated antenna. With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz … 868 MHz) version.	
Long range read / write UHF RFID device with integrated antenna and RTC (Real Time Clock). With serial RS232/RS485 communication interface and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz … 868 MHz) version.	5526U-RTC-G
Long range read / write UHF RFID device with integrated antenna. With serial RS232/RS485 communication interface and Wiegand communication interface. Grey (RAL 7045) case color. EU1 (865 MHz … 868 MHz) version.	5527U-G
Long range read / write UHF RFID device with integrated antenna. With CAN bus and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz … 868 MHz) version.	5528U-G
Long range read / write UHF RFID device with integrated antenna and RTC (Real Time Clock). With CAN bus and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz … 868 MHz) version.	5528U-RTC-G
Read / write 500mW UHF RFID RTU (Ready to Use) device with one external antenna. Serial RS232 / RS485 communication interface. EU1 (865 MHz … 868 MHz) version.	5721U 

and (from) hardware and firmware versions:

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

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

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

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

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

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

DESKTOP READER HF		
Order Number	**Hardware Version**	**Firmware Version**
3122H	2 + 1	2.31a + 1.23

DESKTOP READER HF		
Order Number	**Hardware Version**	**Firmware Version**
3122H	2 + 2	2.31a + 2.07

DESKTOP READER NFC		
Order Number	**Hardware Version**	**Firmware Version**
3122N	2 + 1	2.31a + 1.11

DESKTOP READER UHF		
Order Number	**Hardware Version**	**Firmware Version**
3122U	2 + 1	2.31a + 1.54

PANEL READER LF		
Order Number	**Hardware Version**	**Firmware Version**
3221L	3	3.17d
3121L	4	4.06d
3222L	3	3.17d
3222L	4	4.06d
3223L	1 + 3	1.00 + 3.17d
3223L	1 + 4	1.00 + 4.06d

PANEL READER NFC		
Order Number	**Hardware Version**	**Firmware Version**
3221N	1	1.11
3221N	2	2.07
3222N	1	1.11
3222N	2	2.07
3223N	1 + 1	1.00 + 1.11

PANEL READER NFC		
Order Number	**Hardware Version**	**Firmware Version**
3223N	1 + 2	1.00 + 2.07

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

BASIC READER HF		
Order Number	**Hardware Version**	**Firmware Version**
5121H	2 + 1	2.21 + 1.23
5121H	2 + 2	2.21 + 2.07
5131H	2 + 1	2.21 + 1.23
5131H	2 + 2	2.21 + 2.07

BASIC READER UHF		
Order Number	**Hardware Version**	**Firmware Version**
5121U	2 + 1	2.21 + 1.54
5131U	2 + 1	2.21 + 1.54

ADVANT READER LF		
Order Number	**Hardware Version**	**Firmware Version**
5221L	1 + 3	2.01 + 3.17d
5222L	1 + 3	2.01 + 3.17d
5231L	1 + 3	2.01 + 3.17d
5232L	1 + 3	2.01 + 3.17d

ADVANT READER LF		
Order Number	**Hardware Version**	**Firmware Version**
5241L	1 + 3	2.01 + 3.17d
5242L	1 + 3	2.01 + 3.17d

ADVANT READER HF		
Order Number	**Hardware Version**	**Firmware Version**
5221H	1 + 1	2.10 + 1.23
5222H	1 + 1	2.10 + 1.23
5231H	1 + 1	2.10 + 1.23
5232H	1 + 1	2.10 + 1.23
5241H	1 + 1	2.10 + 1.23
5242H	1 + 1	2.10 + 1.23

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

CX READER UHF		
Order Number	**Hardware Version**	**Firmware Version**

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

CX READER UHF

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

CX E READER UHF

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

RTU READER UHF

Order Number	Hardware Version	Firmware Version
5721U	1	1.54M

Table of Contents

1 Introduction

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

2 Protocol Specifications

The 'master/slave' serial / ethernet protocol expects that the **BLUEBOX** (as 'slave') after the reception of a message sent to it by the 'host' (as 'master'), sends back an answer message after a minimum time of about 10 ms. For the communication through the serial line interface, by default, the **BLUEBOX** will apply the following parameters: 'master/slave' protocol address 255, baud rate 19200 bps, 8 data bits, parity none and 1 stop bit. These parameters can be modified as specified in the 'General Parameters Programming' protocol command. For the communication through the ethernet interface, by default, the **BLUEBOX** will apply the following parameters: 'master/slave' protocol address 255, IP address 192.168.4.200, port 3000, subnet mask 255.255.255.0 and gateway address 0.0.0.0. These parameters can be modified as specified in the 'General Parameters Programming' protocol command for the 'master/slave' protocol address and in the 'Configuration Parameters Programming' protocol command.

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

SOH	Character 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).

2.1 Device Reset

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

The 'master' sends the following command:

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

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

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

Otherwise it answers:

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

2.2 Read Device Serial Number

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

The 'master' sends the following command:

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

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

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

Otherwise it answers:

SOH <add h> <add l> STX '2' 'A' '0' '1' <sn 1 h> <sn 1 l> … <sn i h> <sn i l> … <sn 6 h> <sn 6 l> ETX <bcc> CR

Where:

i	1 … 6.
<sn i h> <sn i l>	i-th byte of the SN of the device. ASCII encoded byte.

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

2.3 Read Ethernet MAC Address

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

The 'master' sends the following command:

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

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

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

Otherwise it answers:

SOH <add h> <add l> STX '2' 'A' '0' '2' <mac 1 h> <mac 1 l> … <mac i h> <mac i l> … <mac 6 h> <mac 6 l> ETX <bcc> CR

Where:

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

2.4 Read Bluetooth MAC Address

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

The 'master' sends the following command:

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

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

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

Otherwise it answers:

SOH <add h> <add l> STX '2' 'A' '0' '6' <mac 1 h> <mac 1 l> … <mac i h> <mac i l> … <mac 6 h> <mac 6 l> ETX <bcc> CR

Where:

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

2.5 Read Firmware Version

This command is used to get the firmware version of the **BLUEBOX**.

The 'master' sends the following command:

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

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

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

Otherwise it answers:

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 a string of 16 ASCII characters that defines the version.

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

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

SOH <add h> <add l> STX '3' '4' '0' '1' (for module 2: **'2'**) **ETX <bcc> CR**

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

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

Otherwise it answers:

SOH <add h> <add l> STX '3' '4' '0' '1' (for module 2: **'2') <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 64 h> <vf 64 l>	ASCII coding of the byte 64 of the string

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

2.6 Firmware Upgrade

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

The 'master' sends the following command:

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

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

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

Otherwise it answers:

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

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

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

SOH <add h> <add l> STX '3' '3' '0' '1' (for module 2: **'2'**) **ETX <bcc> CR**

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

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

Otherwise it answers:

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

With a successful reply and so the firmware upgrade procedure is started, once the **BLUEBOX** is able to receive the image file, it sends the following string

Upload Firmware?

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

Do not power off the **BLUEBOX** during the firmware upgrade procedure, it shall reboot itself at the end of the firmware upgrade.

2.7 Read Temperature

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

The 'master' sends the following command:

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

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

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

Otherwise it answers:

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

SOH <add h> <add l> STX '3' 'A' <temp 1 h> <temp 1 l> <temp 2 h> <temp 2 l> ETX <bcc> CR

Where:

<temp 1 h> <temp 1 l>	Integer value of the temperature in °C. ASCII encoded byte.
<temp 2 h> <temp 2 l>	Fractional value of the temperature. ASCII encoded byte. Bits 7, 6, 5 encode the fractional value in steps of 0.125 °C: • 00000000b → .000 °C • 00100000b → .125 °C • ... • 11100000b → .875°C

28h, E0h -> 40.875 °C

b) for other devices:

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

Where:

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

2.8 Read Date/Time

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

The 'master' sends the following command:

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

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

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

Otherwise it answers:

SOH <add h> <add l> STX '2' '8' <year 1 h> <year 1 l> <year 2 h> <year 2 l> <mon h> <mon l> <day h> <day l> <hou h> <hou l><min h> <min l> <sec h> <sec l> ETX <bcc> CR

Where:

<year 1 h> <year 1 l>	Year value thousands and hundereds. ASCII encoded byte. BCD encoded byte.
<year 2 h> <year 2 l>	Year value tens and units. ASCII encoded byte. BCD encoded byte.
<mon h> <mon l>	Month value tens and units. ASCII encoded byte. BCD encoded byte.
<day h> <day l>	Day value tens and units. ASCII encoded byte. BCD encoded byte.
<hou h> <hou l>	Hour value tens and units. ASCII encoded byte. BCD encoded byte.
<min h> <min l>	Minute value tens and units. ASCII encoded byte. BCD encoded byte.
<sec h> <sec l>	Second value tens and units. ASCII encoded byte. BCD encoded byte.

2.9 Write Date/Time

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

The 'master' sends the following command:

SOH <add h> <add l> STX '2' '9' <year 1 h> <year 1 l> <year 2 h> <year 2 l> <mon h> <mon l> <day h> <day l> <hou h> <hou l> <min h> <min l> <sec h> <sec l> ETX <bcc> CR

Where:

<year 1 h> <year 1 l>	Year value thousands and hundereds. ASCII encoded byte. BCD encoded byte.
<year 2 h> <year 2 l>	Year value tens and units. ASCII encoded byte. BCD encoded byte.
<mon h> <mon l>	Month value tens and units. ASCII encoded byte. BCD encoded byte.
<day h> <day l>	Day value tens and units. ASCII encoded byte. BCD encoded byte.
<hou h> <hou l>	Hour value tens and units. ASCII encoded byte. BCD encoded byte.
<min h> <min l>	Minute value tens and units. ASCII encoded byte. BCD encoded byte.
<sec h> <sec l>	Second value tens and units. ASCII encoded byte. BCD encoded byte.

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

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

Otherwise it answers:

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

2.10 Write ROM General Parameters

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

The 'master' sends the following command:

SOH <adda h> <adda l> STX '2' 'F' <param 1 h> <param 1 l> … <param i h> <param i l> … <param 7 h> <param 7 l> ETX <bcc> CR

Where:

i	1…7

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

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

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

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

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

After the command execution, the **BLUEBOX** performs a device reset to apply the change in RAM as well.

2.11 Write RAM Configuration Parameters

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

The 'master' sends the following command:

SOH <add h> <add l> STX '3' 'F' <page h> <page l> <param 1 h> <param 1 l> … <param i h> <param i l> … <param n h> <param n l> ETX <bcc> CR

Where:

<page h> <page l>	The configuration page (0x00 … 0x0F, 0x80 … 0x87, 0xC0 … 0xCF). ASCII encoded byte.
i	1…n
n	The configuration parameters array size in bytes: • 7 if configuration page is 0x00 … 0x0F; • 14 if configuration page is 0x80 … 0x87; • Variable size (max 240 bytes) null terminated string if configuration page is 0xC0 … 0xCF.
<param i h> <param i l>	i-th byte of the configuration parameters (see the reader user manual for details). ASCII encoded byte.

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

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

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

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

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

2.12 Write ROM Configuration Parameters

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

The 'master' sends the following command:

SOH <add h> <add l> STX '3' 'D' <page h> <page l> <param 1 h> <param 1 l> … <param i h> <param i l> … <param n h> <param n l> ETX <bcc> CR

Where:

<page h> <page l>	The configuration page (0x00 … 0x0F, 0x80 … 0x87, 0xC0 … 0xCF). ASCII encoded byte.
i	1…n
n	The configuration parameters array size in bytes: • 7 if configuration page is 0x00 … 0x0F; • 14 if configuration page is 0x80 … 0x87; • Variable size (max 240 bytes) null terminated string if configuration page is 0xC0 … 0xCF.
<param i h> <param i l>	i-th byte of the configuration parameters (see the reader user manual for details). ASCII encoded byte.

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

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

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

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

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

2.13 Write ROM Default Parameters

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

The 'master' sends the following command:

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

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

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

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

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

After the command execution, the **BLUEBOX** perfoms a device reset to apply the change in RAM as well.

2.14 Read RAM General Parameters

This command is used to read from RAM the values of the 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' <param 1 h> <param 1 l> … <param i h> <param i l> … <param 7 h> <param 7 l> ETX <bcc> CR

Where:

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

2.15 Read RAM Configuration Parameters

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

The 'master' sends the following command:

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

Where:

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

If the addressed **BLUEBOX** 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' 'C' <param 1 h> <param 1 l> … <param i h> <param i l> … <param n h> <param n l> ETX <bcc> CR

Where:

i	1…n
n	The configuration parameters array size in bytes: • 7 if configuration page is 0x00 … 0x0F; • 14 if configuration page is 0x80 … 0x87; • Variable size (max 240 bytes) null terminated string if configuration page is 0xC0 … 0xCF.
<param i h> <param i l>	i-th byte of the configuration parameters (see the reader user manual for details). ASCII encoded byte.

2.16 Read ROM Configuration Parameters

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

The 'master' sends the following command:

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

Where:

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

If the addressed **BLUEBOX** 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' 'E' <param 1 h> <param 1 l> … <param i h> <param i l> … <param n h> <param n l> ETX <bcc> CR

Where:

i	1…n
n	The configuration parameters array size in bytes: • 7 if configuration page is 0x00 … 0x0F; • 14 if configuration page is 0x80 … 0x87; • Variable size (max 240 bytes) null terminated string if configuration page is 0xC0 … 0xCF.

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

2.17 Sleep Mode

This command is used to manage the sleep mode of the **BLUEBOX**.

The 'master' sends the following command:

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

Where:

<mode h> <mode l>	The sleep mode. ASCII encoded byte: • 0x00: To exit from sleep mode. • 0x01: To enter in sleep mode. Any command on serial interface wakes up the device, after reply the device stays active. • 0x02: To enter in sleep mode. Any command on serial interface wakes up the device, after reply the device enters in sleep mode again.

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

2.18 'RF Reading' Test

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

The 'master' sends the following command:

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

Where:

<onoff h> <onoff l>	To activate/deactivate the 'RF reading' test mode. ASCII encoded byte: • 0x00: To deactivate 'RF reading' test mode; • 0x01: To activate 'RF reading' test 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

The 'RF reading' test mode setting is stored in non volatile memory and its status is kept at every restart of the **BLUEBOX**.

2.19 'RF Power' Test

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

The 'master' sends the following command:

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

Where:

<antenna>	Antenna to use for test. ASCII character: • '1' -> Antenna 1. • '2' -> Antenna 2. • '3' -> Antenna 3. • '4' -> Antenna 4.
<channel h> <channel l>	RF channel to use for test. ASCII encoded byte:

	• 0x01 … 0x0A if ETSI region is selected; • 0x01 … 0x32 if FCC region is selected.

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

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

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

a) if a tag has been identified

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

Where:

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

b) if no tag has been found

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

2.20 'RF Sensitivity' Test

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

The 'master' sends the following command:

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

Where:

<antenna>	Antenna to use for test. ASCII character: • '1' -> Antenna 1. • '2' -> Antenna 2. • '3' -> Antenna 3.

	• '4' -> Antenna 4.
<channel h> <channel l>	RF channel to use for test. ASCII encoded byte: • 0x01 … 0x0A if ETSI region is selected; • 0x01 … 0x32 if FCC region is selected.

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

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

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

a) if a tag has been identified

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

Where:

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

b) if no tag has been found

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

2.21 Read Reflected Power

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

The 'master' sends the following command:

SOH <add h> <add l> STX 'F' 'E' '0' <antenna> <freq 1 h> <freq 1 l> <freq 2 h> <freq 2 l> <freq 3 h> <freq 3 l> ETX <bcc> CR

Where:

<antenna>	Antenna to use for test. ASCII character: • '1' -> Antenna 1. • '2' -> Antenna 2. • '3' -> Antenna 3. • '4' -> Antenna 4.

<freq 1 h> <freq 1 l> <freq 2 h> <freq 2 l> <freq 3 h> <freq 3 l>	The frequency to test in MHz in the range 840 – 960 MHz. 3-bytes ASCII encoded value.

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 'F' 'E' <I-ch h> <Ich l> <Qch h> <Qch l> <G h> <G l> ETX <bcc> CR

Where:

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

And the reflected power is calculated as follows

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

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

2.22 Read RSSI Power

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

The 'master' sends the following command:

SOH <add h> <add l> STX 'F' 'D' '0' <antenna> <freq 1 h> <freq 1 l> <freq 2 h> <freq 2 l> <freq 3 h> <freq 3 l> ETX <bcc> CR

Where:

<antenna>	Antenna to use for test. ASCII character:

	• '1' -> Antenna 1. • '2' -> Antenna 2. • '3' -> Antenna 3. • '4' -> Antenna 4.
<freq 1 h> <freq 1 l> <freq 2 h> <freq 2 l> <freq 3 h> <freq 3 l>	The frequency to test in MHz in the range 840 – 960 MHz. 3-bytes ASCII encoded value.

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 'F' 'D' <I-ch h> <Ich l> <Qch h> <Qch l> <G h> <G l> ETX <bcc> CR

Where:

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

And the RSSI power is calculated as follows

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

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

2.23 Digital Output Activation

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

The 'master' sends the following command:

SOH <add h> <add l> STX '3' '7' <chn h> < chn l> <dur h> <dur l> ETX <bcc> CR

Where:

<chn h> <chn l>	Output to activate. ASCII encoded byte: • 0x01 -> Output 1; • 0x02 -> Output 2.
<dur h> <dur l>	Activation time. ASCII encoded byte: • 0x01 … 0x63 (1 ... 99 seconds) -> 'Impulsive' output 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

2.24 Status Reading

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

The 'master' sends the following command:

SOH <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' <status hh> <status hl> <status lh> <status ll> ETX <bcc> CR

Where:

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

2.25 RF Deactivation

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

The 'master' sends the following command:

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

If the addressed **BLUEBOX** 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

2.26 RF Activation

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

The 'master' sends the following command:

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

If the addressed **BLUEBOX** 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

2.27 Antennas Auto-Tuning

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

The 'master' sends the following command:

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

If the addressed **BLUEBOX** 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

2.28 Buffer Data Request

This command sends back the code of the eventual transponder that is present in the buffer. When 'continuous' mode is enabled, the reply is immediate because the **BLUEBOX** sends back the data 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:

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

Otherwise,

a) for LF devices with only 1 antenna, it answers,

a1) if the antenna have identified a transponder:

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

Where:

i	1 … n
n	Number of bytes of the tag code identified by the antenna: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE

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

a2) if no transponder is identified by the antenna:

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

b) for LF devices with 2 antennas, it answers,

b1) if both antennas have identified a transponder:

SOH <add h> <add l> STX <code ant 1 1 h> <code ant 1 1 l> … <code ant 1 i h> <code ant 1 i l> … <code ant 1 n h> <code ant 1 n l> '-' <code ant 2 1 h> <code ant 2 1 l> … <code ant 2 j h> <code ant 2 j l> … <code ant 2 m h> <code ant 2 m l> ETX <bcc> CR

Where:

i	1 … n
n	Number of bytes of the tag code identified by antenna 1: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE
<code ant 1 i h> <code ant 1 i l>	i-th byte of the code of the tag identified by antenna 1. ASCII encoded byte
j	1 … m
m	Number of bytes of the tag code identified by antenna 2: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE
<code ant 2 i h> <code ant 2 i l>	i-th byte of the code of the tag identified by antenna 2. ASCII encoded byte

b2) if only antenna 1 have identified a transponder:

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

Where:

i	1 … n.
n	Number of bytes of the tag code identified by antenna 1: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE
<code ant 1 i h> <code ant 1 i l>	i-th byte of the code of the tag identified by antenna 1. ASCII encoded byte

b3) if only antenna 2 have identified a transponder:

SOH <add h> <add l> STX '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '-' <code ant 2 1 h> <code ant 2 1 l> … <code ant 2 j h> <code ant 2 j l> … <code ant 2 m h> <code ant 2 m l> ETX <bcc> CR

Where:

j	1 … m
m	Number of bytes of the tag code identified by antenna 2: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE
<code ant 2 i h> <code ant 2 i l>	i-th byte of the code of the tag identified by antenna 1. ASCII encoded byte

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

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

c) for HF devices with only 1 antenna it answers,

c1) if the antenna have identified a transponder:

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

Where:

<type h> <type l>	Transponder type.

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

c2) if no transponder is identified by the antenna:

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

d) for HF devices with 2 antennas, it answers,

d1) if both antennas have identified a transponder:

SOH <add h> <add l> STX <type ant 1 h> <type ant 1 l> <UID ant 1 1 h> <UID ant 1 1 l> … <UID ant 1 i h> <UID ant 1 i l> … <UID ant 1 n h> <UID ant 1 n l> '-' <type ant 2 h> <type ant 2 l> <UID ant 2 1 h> <UID ant 2 1 l> … <UID ant 2 j h> <UID ant 2 j l> … <UID ant 2 m h> <UID ant 2 m l> ETX <bcc> CR

Where:

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

d2) if only antenna 1 have identified a transponder:

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

Where:

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

d3) if only antenna 2 have identified a transponder:

SOH <add h> <add l> STX '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '-' <type ant 2 h> <type ant 2 l> <UID ant 2 1 h> <UID ant 2 1 l> … <UID ant 2 j h> <UID ant 2 j l> … <UID ant 2 m h> <UID ant 2 m l> ETX <bcc> CR

Where:

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

d4) if none of the antennas have identified a transponder:

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

e) for UHF devices, it answers with

SOH <add h> <add l> STX <tag 1 h> <tag 1 l> <ID 1 1 h> <ID 1 1 l> … <ID 1 i h> <ID 1 i l> … <ID 1 m h> <ID 1 m l> <1st RSSI Q 1 h> <1st RSSI Q 1 l> <1st RSSI I 1 h> <1st RSSI I 1 l> <last RSSI Q 1 h> <last RSSI Q 1 l> <last RSSI I 1 h> <last RSSI I 1 L> <max RSSI Q 1 h> <max RSSI Q 1 l> <max RSSI I 1 h> <max RSSI I 1 l> <ant 1 h> <ant 1 l> <dir 1 h> <dir 1 l> <1st tm 1 1 h> <1st tm 1 1 l> … <1st tm 1 7 h> <1st tm 1 7 l> <last tm 1 1 h> <last tm 1 1 l> … <last tm 1 7 h> <last tm 1 7 l> <rdcount 1 hh> <rdcount 1 hl> <rdcount 1 lh> <rdcount 1 ll> '-' … '-' <tag j h> <tag j l> <ID j 1 h> <ID j 1 l> … <ID j i h> <ID j i l> … <ID j m h> <ID j m l> <1st RSSI Q j h> <1st RSSI Q j l> <1st RSSI I j h> <1st RSSI I j l> <last RSSI Q j h> <last RSSI Q j l> <last RSSI I j h> <last RSSI I j l> <max RSSI Q j h> <max RSSI Q j l> <max RSSI I j h> <max RSSI I j l> <ant j h> <ant j l> <dir j h> <dir j l> <1st tm j 1 h> <1st tm j 1 l> … <1st tm j 7 h> <1st tm j 7 l> <last tm j 1 h> <last tm j 1 l> … <last tm j 7 h> <last tm j 7 l>

<rdcount j hh> <rdcount j hl> <rdcount j lh> <rdcount j ll> '-' ... '-' <tag n h> <tag n l> <ID n 1 h> <ID n 1 l> ... <ID n i h> <ID n i l> ... <ID n m h> <ID n m l> <1st RSSI Q n h> <1st RSSI Q n l> <1st RSSI I n h> <1st RSSI I n l> <last RSSI Q n h> <last RSSI Q n l> <last RSSI I n h> <last RSSI I n l> <max RSSI Q n h> <max RSSI Q n l> <max RSSI I n h> <max RSSI I n l> <ant n h> <ant n l> <dir n h> <dir n l> <1st tm n 1 h> <1st tm n 1 l> ... <1st tm n 7 h> <1st tm n 7 l> <last tm n 1 h> <last tm n 1 l> ... <last tm n 7 h> <last tm n 7 l> <rdcount n hh> <rdcount n hl> <rdcount n lh> <rdcount n ll> ETX <bcc> CR

Where:

i	1 … m.
m	ID length.
J	1 … n.
n	Number of identified tags.
<tag j h> <tag j l>	Transponder type for the j-th identified tag **(optional parameter present only if the tag type information flag in the general parameters is active, see the reader user manual for more info)**. ASCII encoded byte: • 0x02: ISO 18000-63 (EPC Class-1Generation-2).
<ID j i h> <ID j i l>	i-th byte of the ID of the j-th identified tag. ASCII encoded byte..
<1st RSSI Q j h> <1st RSSI Q j l> <1st RSSI I j h> <1st RSSI I j l>	First seen RSSI Q and I values in dB for the j-th identified tag **(optional parameter present only if the RSSI information flag in the RF configuration parameters is active, see the reader user manual for more info)**. ASCII encoded bytes.
<last RSSI Q j h> <last RSSI Q j l> <last RSSI I j h> <last RSSI I j l>	Last seen RSSI Q and I values in dB for the j-th identified tag **(optional parameter present only if the RSSI information flag in the RF configuration parameters is active, see the reader user manual for more info)**. ASCII encoded bytes.
<max RSSI Q j h> <max RSSI Q j l> <max RSSI I j h> <max RSSI I j l>	Max seen RSSI Q and I values in dB for the j-th identified tag **(optional parameter present only if the max RSSI information flag in the RF configuration parameters is active, see the reader user manual for more info)**. ASCII encoded bytes.
<ant j h> <ant j l>	Reading antenna for the j-th identified tag **(optional parameter present only if the reading antenna**

	information flag in the general parameters is active, see the reader user manual for more info). ASCII character: • 0x01 -> Antenna 1. • 0x02 -> Antenna 2. • 0x03 -> Antenna 3. • 0x04 -> Antenna 4.
<dir j h> <dir j l>	Gate crossing direction for the j-th identified tag **(optional parameter present only if 'gate' mode is active, see the reader user manual for more info)**. ASCII character: • 0x01 -> Crossing from input 1 to input 2. • 0x02 -> Crossing from input 2 to input 1.
<1st tm j 1 h> <1st tm j 1 l> … <1st tm j 7 h> <1st tm j 7 l>	First seen timestamp for the j-th identified tag **(optional parameter present only if the reading timestamp information flag in the general parameters is active, see the reader user manual for more info)**. ASCII encoded byte array of the BCD encoded timestamp with the format yyyyMMddhhmmss.
<last tm j 1 h> <last tm j 1 l> … <last tm j 7 h> <last tm j 7 l>	Last seen timestamp for the j-th identified tag **(optional parameter present only if the reading timestamp information flag in the general parameters is active, see the reader user manual for more info)**. ASCII encoded byte array of the BCD encoded timestamp with the format yyyyMMddhhmmss.
<rdcount j hh> <rdcount j hl> <rdcount j lh> <rdcount j ll>	The tag read count for the j-th identified tag **(optional parameter present only if the tag read count information flag in the RF parameters is active, see the reader user manual for more info)**. ASCII encoded word.
'-'	Separator 0x5F.

2.29 Queue Data Request

In 'continuous' mode, when the **BLUEBOX** finds a 'new' transponder, it inserts its code in the FIFO queue. This command sends back the first present code in the queue.

The 'master' sends the following command:

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

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

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

Otherwise,

a) for LF devices with only 1 antenna, it answers:

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

Where:

i	1 … n
n	Number of bytes of the tag code: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE
<code i h> <code i l>	i-th byte of the code of the identified tag. ASCII encoded byte

b) for LF devices with 2 antennas, it answers:

SOH <add h> <add l> STX <code 1 h> <code 1 l> … <code i h> <code i l> … <code n h> <code n l> <ant h> <ant l> ETX <bcc> CR

Where:

i	1 … n
n	Number of bytes of the tag code: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE
<code i h> <code i l>	i-th byte of the code of the identified tag. ASCII encoded byte
<ant h> <ant l>	The antenna number which have identified the tag: ASCII encoded byte: • 0x01: Antenna 1 • 0x02: Antenna 2

c) for HF devices with only 1 antenna it answers:

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

Where:

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

d) for HF devices with 2 antennas, it answers:

SOH <add h> <add l> STX <type h> <type l> <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID n h> <UID n l> <ant h> <ant l> ETX <bcc> CR

Where:

<type h> <type l>	Transponder type.
i	1 … n (the UID length).
<UID i h> <UID i l>	i-th byte of the UID code of the identified tag. ASCII encoded byte.
<ant h> <ant l>	The antenna number which have identified the tag: ASCII encoded byte: • 0x01: Antenna 1 • 0x02: Antenna 2

e) for UHF devices, it answers:

SOH <add h> <add l> STX <tag h> <tag l> <ID 1 h> <ID 1 l> … <ID i h> <ID i l> … <ID m h> <ID m l> <RSSI Q h> <RSSI Q l> <RSSI I h> <RSSI I l> <ant h> <ant l> <dir h> <dir l> <tm 1 h> <tm 1 l> … <tm 7 h> <tm 7 l> ETX <bcc> CR

Where:

i	1 … m.
m	ID length.
<tag h> <tag l>	Transponder type of the identified tag **(optional parameter present only if the tag type information flag in the general parameters is active, see the reader user manual for more info)**. ASCII encoded byte:

	• 0x02: ISO 18000-63 (EPC Class-1Generation-2).
<ID I h> <ID I l>	i-th byte of the ID of the identified tag. ASCII encoded byte.
<RSSI Q h> <RSSI Q l> <RSSI I h> <RSSI I l>	RSSI Q and I channel info in dB of the identified tag. **(optional parameter present only if the RSSI information flag in the RF configuration parameters is active, see the reader user manual for more info)**. ASCII encoded bytes.
<ant h> <ant l>	Reading antenna for the identified tag **(optional parameter present only if the reading antenna information flag in the general parameters is active, see the reader user manual for more info)**. ASCII character: • 0x01 -> Antenna 1. • 0x02 -> Antenna 2. • 0x03 -> Antenna 3. • 0x04 -> Antenna 4.
<dir h> <dir l>	Gate crossing direction for the identified tag **(optional parameter present only if 'gate' mode is active, see the reader user manual for more info)**. ASCII character: • 0x01 -> Crossing from input 1 to input 2. • 0x02 -> Crossing from input 2 to input 1.
<tm 1 h> <tm 1 l> … <tm 7 h> <tm 7 l>	Timestamp for the identified tag **(optional parameter present only if the reading timestamp information flag in the general parameters is active, see the reader user manual for more info)**. ASCII encoded byte array of the BCD encoded timestamp with the format yyyyMMddhhmmss.

If the queue is empty, the **BLUEBOX** will answer with:

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

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

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

2.30 Read Number of Records

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

The 'master' sends the following command:

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

If the addressed **BLUEBOX** 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 '0' '0' <num hh> <num hl> <num lh> <num ll> ETX <bcc> CR

Where:

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

2.31 Reset Record Database

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

The 'master' sends the following command:

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

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

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

Otherwise it answers:

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

2.32 Read Current Record

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

The 'master' sends the following command:

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

If the addressed **BLUEBOX** 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 '0' '1' <idt hh> <idt hl> <idt lh> <idt ll> <record 1 h> <record 1 l> … <record n h> <record n l> ETX <bcc> CR

Where:

<idt hh> <idt hl> <idt lh> <idt ll>	Index of the record in the database. ASCII encoded word.
n	The record size in bytes. Must be a power of 2 minus 1 bytes of header and checksum (for example 15 or 31 or 63 …).
<record 1 h> <record 1 l>	1st byte of the record. ASCII encoded byte.
…	
<record n h> <record n l>	Last byte of the record. ASCII encoded byte.

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

2.33 Unqueue Current Record

This command is used to unque the first unread record in the database of the **BLUEBOX**.

The 'master' sends the following command:

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

Where:

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

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

The field <idt> in the command is the index of the record to unqueue received with the message '0' '1'.

2.34 Re-Read an Unqueued Record

This command is used to get an already read and unqueued record from the database of the **BLUEBOX**.

The 'master' sends the following command:

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

Where:

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

If the addressed **BLUEBOX** 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 '0' '3' <num hh> <num hl> <num lh> <num ll> <record 1 h> <record 1 l> … <record n h> <record n l> ETX <bcc> CR

Where:

<num hh> <num hl> <num lh> <num ll>	The index of the already read record starting from the first unread record in the database. ASCII encoded word. 0001 to read the last read record, …, 0000 not allowed.
n	The record size in bytes. Must be a power of 2 minus 1 bytes of header and checksum (for example 15 or 31 or 63 …).
<record 1 h> <record 1 l>	1st byte of the record. ASCII encoded byte.
…	
<record n h> <record n l>	Last byte of the record. ASCII encoded byte.

2.35 Start Continuous Read Records

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

The 'master' sends the following command:

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

If the addressed **BLUEBOX** 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 '5' '0' <idt hh> <idt hl> <idt lh> <idt ll> <num hh> <num hl> <num lh> <num ll> ETX <bcc> CR

SOH <add h> <add l> STX '5' '0' <id hh> <id hl> <id lh> <id ll> <record 1 h> <record 1 l> … <record n h> <record n l> ETX <bcc> CR

< … >

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

Where:

<idt hh> <idt hl> <idt lh> <idt ll>	Index of the first unread record in the database. ASCII encoded word.
<num hh> <num hl> <num lh> <num ll>	Number of unread records in the database. ASCII encoded word.
n	The record size in bytes. Must be a power of 2 minus 1 bytes of header and checksum (for example 15 or 31 or 63 …).
<record 1 h> <record 1 l>	1st byte of the record. ASCII encoded byte.
…	
<record n h> <record n l>	Last byte of the record. ASCII encoded byte.
< … >	The records to read.

In case of no unread records, it answers with:

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

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

2.36 Stop Continuous Read Records

This command is used to stop the continuous reception of the records and to unqueue the records from the database of the **BLUEBOX**.

The 'master' sends the following command:

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

Where:

<idt hh> <idt hl>	Index of the first unread record in the database. ASCII

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

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

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

Otherwise it answers:

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

2.37 Write Data to a EM4305 Transponder

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

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '1' '9' with '6' '9'):

SOH <add h> <add l> STX '1' '9' (for antenna 2: **'6' '9'**) **<code 1 h> … <code n l> … <code i h> <code i l> … <code n h> <code n l> ETX <bcc> CR**

Where:

i	1 … n.
n	Number of bytes of the tag code: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE
<code i h> <code i l>	i-th byte of the code of the tag. ASCII encoded byte.

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

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

Otherwise it answers:

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

2.38 Read ID Code of an EM4305 Transponder

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '1' '8' with '6' '8'):

SOH <add h> <add l> STX '1' '8' (for antenna 2: **'6' '8'**) **ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' '8' (for antenna 2: **'6' '8'**) **'0' '0' <code 1 h> <code 1 l> … <code i h> <code i l> … <code 4 h> <code 4 l> ETX <bcc> CR**

Where:

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

b) if errors occurred:

SOH <add h> <add l> STX '1' '8' (for antenna 2: **'6' '8'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '1' '8' (for antenna 2: **'6' '8'**) **'0' '1' ETX <bcc> CR**

2.39 Write Data to a T5557 Transponder

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

- **T5557 BLUEBOX SHORT,** the code is constituted by 40 bits divided into 10 nibbles (UNIQUE compatible) giving 5 bytes
- **T5557 BLUEBOX MEDIUM,** the code is constituted by 80 bits divided into 20 nibbles giving 10 bytes
- **T5557 BLUEBOX LARGE,** the code is constituted by 160 bits divided into 40 nibbles giving 20 bytes

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '1' 'D' with '6' 'D'):

SOH <add h> <add l> STX '1' 'D' (for antenna 2: **'6' 'D'**) **<code 1 h> … <code n l> … <code i h> <code i l> … <code n h> <code n l> ETX <bcc> CR**

Where:

i	1 … n.
n	Number of bytes of the tag code: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE
<code i h> <code i l>	i-th byte of the code of the tag. ASCII encoded byte

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

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

Otherwise it answers:

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

2.40 Read ID Code of a T5557 Transponder

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '1' 'C' with '6' 'C'):

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'6' 'C'**) **ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'6' 'C'**) **'0' '0' <code 1 h> <code 1 l> … <code i h> <code i l> … <code 8 h> <code 8 l> ETX <bcc> CR**

Where:

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

b) if errors occurred:

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'6' 'C'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'6' 'C'**) **'0' '1' ETX <bcc> CR**

2.41 Write Data to a Q5 Transponder

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

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '2' '1' with '7' '1'):

SOH <add h> <add l> STX '2' '1' (for antenna 2: **'7' '1'**) **<code 1 h> … <code n l> … <code i h> <code i l> … <code n h> <code n l> ETX <bcc> CR**

Where:

i	1 … n.
n	Number of bytes of the tag code: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE
<code i h> <code i l>	i-th byte of the code of the tag. ASCII encoded byte

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

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

Otherwise it answers:

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

2.42 Read ID Code of a Q5 Transponder

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '2' '0' with '7' '0'):

SOH <add h> <add l> STX '2' '0' (for antenna 2: **'7' '0'**) **ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '2' '0' (for antenna 2: **'7' '0'**) **'0' '0' <code 1 h> <code 1 l> … <code i h> <code i l> … <code 5 h> <code 5 l> ETX <bcc> CR**

Where:

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

b) if errors occurred:

SOH <add h> <add l> STX '2' '0' (for antenna 2: **'7' '0'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '2' '0' (for antenna 2: **'7' '0'**) **'0' '1' ETX <bcc> CR**

2.43 Write Data to a HITAG S Transponder

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

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '2' '3' with '7' '3'):

SOH <add h> <add l> STX '2' '3' (for antenna 2: **'7' '3'**) **<code 1 h> … <code n l> … <code i h> <code i l> … <code n h> <code n l> ETX <bcc> CR**

Where:

i	1 … n
n	Number of bytes of the tag code: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM

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

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

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

b) if errors occurred:

SOH <add h> <add l> STX '2' '3' (for antenna 2: **'7' '3'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '2' '3' (for antenna 2: **'7' '3'**) **'0' '1' ETX <bcc> CR**

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

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '2' '2' with '7' '2'):

SOH <add h> <add l> STX '2' '2' (for antenna 2: **'7' '2'**) **ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '2' '2' (for antenna 2: **'7' '2'**) **'0' '0' <type h> <type l> <code 1 h> <code 1 l> … <code i h> <code i l> … <code 4 h> <code 4 l> ETX <bcc> CR**

Where:

i	1 … 4
<type h> <type l>	The tag type. ASCII encoded byte: • 0x01 -> HITAG S 256 • 0x02 -> HITAG S 2048

	• 0x03 -> HITAG 1
<code i h> <code i l>	i-th byte of the code of the tag. ASCII encoded byte.

b) if errors occurred:

SOH <add h> <add l> STX '2' '2' (for antenna 2: **'7' '2'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '2' '2' (for antenna 2: **'7' '2'**) **'0' '1' ETX <bcc> CR**

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

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '2' '4' with '7' '4'):

SOH <add h> <add l> STX '2' '4' (for antenna 2: **'7' '4'**) **<code 1 h> <code 1 l> … <code i h> <code i l> … <code 4 h> <code 4 l> <pag h> <pag l> ETX <bcc> CR**

Where:

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

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '2' '4' (for antenna 2: **'7' '4'**) **'0' '0' <data 1 h> <data 1 l> … <data i h> <data i l> … <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

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

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '2' '4' (for antenna 2: **'7' '4'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '2' '4' (for antenna 2: **'7' '4'**) **'0' '1' ETX <bcc> CR**

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

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '2' '5' with '7' '5'):

SOH <add h> <add l> STX '2' '5' (for antenna 2: **'7' '5'**) **<code 1 h> <code 1 l> … <code i h> <code i l> … <code 4 h> <code 4 l> <pag h> <pag l> <data 1 h> <data 1 l> … <data j h> <data j l> … <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

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

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

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '2' '5' (for antenna 2: **'7' '5'**) **'0' '0' ETX <bcc> CR**

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '2' '5' (for antenna 2: **'7' '5'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '2' '5' (for antenna 2: **'7' '5'**) **'0' '1' ETX <bcc> CR**

2.47 Read ID Code of a HITAG 2 Transponder

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '4' '8' with '9' '8'):

SOH <add h> <add l> STX '4' '8' (for antenna 2: **'9' '8'**) **ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '8' (for antenna 2: **'9' '8'**) **'0' '0' <code 1 h> <code 1 l> … <code i h> <code i l> … <code 4 h> <code 4 l> ETX <bcc> CR**

Where:

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

b) if errors occurred:

SOH <add h> <add l> STX '4' '8' (for antenna 2: **'9' '8'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '4' '8' (for antenna 2: **'9' '8'**) **'0' '1' ETX <bcc> CR**

2.48 Read a Page of a HITAG 2 Transponder

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '4' 'A' with '9' 'A'):

SOH <add h> <add l> STX '4' 'A' (for antenna 2: **'9' 'A'**) **<code 1 h> <code 1 l> … <code i h> <code i l> … <code 4 h> <code 4 l> <pwd 1 h> < pwd 1 l> … < pwd i h> < pwd i l> … < pwd 4 h> < pwd 4 l> <pag h> <pag l> ETX <bcc> CR**

Where:

i	1 … 4
<code i h> <code i l>	i-th byte of the ID code of the tag. ASCII encoded byte
<pwd i h> <pwd i l>	i-th byte of the password of the tag. ASCII encoded byte
<pag h> <pag l>	Page to be read. ASCII encoded byte (0x00 … 0x7).

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' 'A' (for antenna 2: **'9' 'A'**) **'0' '0' <data 1 h> <data 1 l> … <data i h> <data i l> … <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

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

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '4' 'A' (for antenna 2: **'9' 'A'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '4' 'A' (for antenna 2: **'9' 'A'**) **'0' '1' ETX <bcc> CR**

2.49 Write a Page of a HITAG 2 Transponder

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

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '4' 'B' with '9' 'B'):

SOH <add h> <add l> STX '4' 'B' (for antenna 2: **'9' 'B'**) **<code 1 h> <code 1 l> … <code i h> <code i l> … <code 4 h> <code 4 l> <pwd 1 h> < pwd 1 l> … < pwd i h> < pwd i l> … < pwd 4 h> < pwd 4 l> <pag h> <pag l> <data 1 h> <data 1 l> … <data j h> <data j l> … <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

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

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

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' 'B' (for antenna 2: **'9' 'B'**) **'0' '0' ETX <bcc> CR**

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '4' 'B' (for antenna 2: **'9' 'B'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '4' 'B' (for antenna 2: **'9' 'B'**) **'0' '1' ETX <bcc> CR**

2.50 'Reset' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to reset the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder.

If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '4' '0' with '6' '0'):

SOH <add h> <add l> STX '4' '0' (for antenna 2: **'6' '0'**) **ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '0' (for antenna 2: **'6' '0'**) **'0' '0' ETX <bcc> CR**

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '4' '0' (for antenna 2: **'6' '0'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '4' '0' (for antenna 2: **'6' '0'**) **'0' '1' ETX <bcc> CR**

2.51 'Login' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to log in the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder.

If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '4' '1' with '6' '1'):

SOH <add h> <add l> STX '4' '1' (for antenna 2: **'6' '1'**) **<pw 1 h> <pw 1 l> … <pw i h> <pw i l> … <pw 4 h> <pw 4 l> ETX <bcc> CR**

Where:

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

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '1' (for antenna 2: **'6' '1'**) **'0' '0' ETX <bcc> CR**

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '4' '1' (for antenna 2: **'6' '1'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '4' '1' (for antenna 2: **'6' '1'**) **'0' '1' ETX <bcc> CR**

2.52 'Write Password' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to set the password of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder.

If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '4' '2' with '6' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'6' '2'**) **<pw a 1 h> <pw a 1 l> … <pw a i h> <pw a i l> … <pw a 4 h> <pw a 4 l> <pw n 1 h> <pw n 1 l> … <pw n i h> <pw n i l> … <pw n 4 h> <pw n 4 l> ETX <bcc> CR**

Where:

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

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'6' '2'**) **'0' '0' ETX <bcc> CR**

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'6' '2'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'6' '2'**) **'0' '1' ETX <bcc> CR**

2.53 'Standard Read' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to get the data relative to 'standard read' mode of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder.

If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '4' '3' with '6' '3'):

SOH <add h> <add l> STX '4' '3' (for antenna 2: **'6' '3'**) **ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '3' (for antenna 2: **'6' '3'**) **'0' '0' <data 1 1 h> <data 1 1 l> … <data i 1 h> <data i 1 l> … <data 4 1 h> <data 4 1 l> … <data 1 j h> <data 1 j l> … <data i j h> <data i j l> … <data 4 j**

h> <data 4 j l> … <data 1 n h> <data 1 n l> … <data i n h> <data i n l> … <data 4 n h> <data 4 n l> ETX <bcc> CR

Where:

i	1 … 4
j	1 … n
n	Number of pages read
<data i j h> <data i j l>	i-th byte of the j-th page read. ASCII encoded byte

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '4' '3' (for antenna 2: **'6' '3'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '4' '3' (for antenna 2: **'6' '3'**) **'0' '1' ETX <bcc> CR**

2.54 'Selective Read' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to read the data relative to 1 or more long word/s of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder.

If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '4' '4' with '6' '4'):

SOH <add h> <add l> STX '4' '4' (for antenna 2: **'6' '4'**) **<add u h> <add u l> <add p h> <add p l> ETX <bcc> CR**

Where:

<add p h> <add p l>	Address of the first page to be read. ASCII encoded byte
<add u h> <add u l>	Address of the last page to be read. ASCII encoded byte

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '4' (for antenna 2: **'6' '4'**) **'0' '0' <data 1 1 h> <data 1 1 l> … <data i 1 h> <data i 1 l> … <data 4 1 h> <data 4 1 l> … <data 1 j h> <data 1 j l> … <data i j h> <data i j l> … <data 4 j h> <data 4 j l> … <data 1 n h> <data 1 n l> … <data i n h> <data i n l> … <data 4 n h> <data 4 n l> ETX <bcc> CR**

Where:

i	1 … 4
j	1 … n
n	Number of read pages
<data i j h> <data i j l>	i-th byte of the j-th read page. ASCII encoded byte

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '4' '4' (for antenna 2: **'6' '4'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '4' '4' (for antenna 2: **'6' '4'**) **'0' '1' ETX <bcc> CR**

2.55 'Write Word' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to write the data relative to a long word of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder.

If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '4' '5' with '6' '5'):

SOH <add h> <add l> STX '4' '5' (for antenna 2: **'6' '5'**) **<add w h> <add w l> <data 1 h> <data 1 l> … <data i h> <data i l> … <data 4 h> data 4 l> ETX <bcc> CR**

Where:

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

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '5' (for antenna 2: **'6' '5'**) **'0' '0' ETX <bcc> CR**

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '4' '5' (for antenna 2: **'6' '5'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '4' '5' (for antenna 2: **'6' '5'**) **'0' '1' ETX <bcc> CR**

2.56 'Write Several Words' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to write the data relative to more long words of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder.

If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the

TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '4' '6' with '6' '6'):

SOH <add h> <add l> STX '4' '6' (for antenna 2: **'6' '6'**) **<add w h> <add w l> <data 1 1 h> <data 1 1 l> … <data i 1 h> <data i 1 l> … <data 4 1 h> <data 4 1 l> … <data 1 j h> <data 1 j l> … <data i j h> <data i j l> … <data 4 j h> <data 4 j l> … <data 1 n h> <data 1 n l> … <data i n h> <data i n l> … <data 4 n h> <data 4 n l> ETX <bcc> CR**

Where:

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

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '6' (for antenna 2: **'6' '6'**) **'0' '0' ETX <bcc> CR**

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '4' '6' (for antenna 2: **'6' '6'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '4' '6' (for antenna 2: **'6' '6'**) **'0' '1' ETX <bcc> CR**

2.57 'Read After Write Word' Command for TITAN Transponder

If the **BLUEBOX** 'continuous' mode is disabled, this command allows to write and read back the data relative to a long word of the TITAN transponder. Refer to the related datasheet to get more information about the TITAN transponder.

If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as a command involving the TITAN transponder is used; it will be resumed automatically when another type of command will be executed.

The 'master' sends the following command (for devices with 2 antennas, this command is valid to work with the antenna nr 1; to work with the antenna nr 2, replace '4' '7' with '6' '7'):

SOH <add h> <add l> STX '4' '7' (for antenna 2: **'6' '7'**) **<add w h> <add w l> <data 1 h> <data 1 l> … <data i h> <data i l> … <data 4 h> data 4 l> ETX <bcc> CR**

Where:

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

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '7' (for antenna 2: **'6' '7'**) **'0' '0' <data 1 h> <data 1 l> … <data i h> <data i l> … <data 4 h> data 4 l> ETX <bcc> CR**

Where:

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

b) if a transponder is present but errors occurred:

SOH <add h> <add l> STX '4' '7' (for antenna 2: **'6' '7'**) **'0' '2' ETX <bcc> CR**

c) if a transponder is not present:

SOH <add h> <add l> STX '4' '7' (for antenna 2: **'6' '7'**) **'0' '1' ETX <bcc> CR**

2.58 ISO 15693 Transponders 'Inventory' Command

This command is used to get the UID code of the identified ISO 15693 transponders that are present near the antenna/s. For devices with 2 antennas, the command code 0x10 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x90 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' '0' with '9' '0'):

SOH <add h> <add l> STX '1' '0' (for antenna 2: **'9' '0'**) **ETX <bcc> CR**

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

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

Otherwise it answers,

a) if at least one tag is present:

SOH <add h> <add l> STX '1' '0' (for antenna 2: **'9' '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**

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.

b) if some error is occurred during the transaction:

SOH <add h> <add l> STX '1' '0' (for antenna 2: **'9' '0'**) **'0' '2' ETX <bcc> CR**

c) if no tag is present:

SOH <add h> <add l> STX '1' '0' (for antenna 2: **'9' '0'**) **'0' '1' ETX <bcc> CR**

2.59 Read a Data Block of an ISO 15693 Transponder

This command is used to get a data block of a known (UID) ISO 15693 transponder. Note that the number of bytes of a block and the number of blocks depends on the transponder type; for example, the **NXP I CODE SLI** transponder is organized in blocks of 4 bytes, the **Fujitsu MB89R118** transponder is organized in blocks of 8 bytes, for more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x11 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x91 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' '1' with '9' '1'):

SOH <add h> <add l> STX '1' '1' (for antenna 2: **'9' '1'**) **<UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 8 h> <UID 8 l> <blk h> <blk 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.
<blk h> <blk l>	Address of the block to read. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' '1' (for antenna 2: **'9' '1'**) **'0' '0' <data 1 h> <data 1 l>… <data i h> <data i l>… <data n h> <data n l> ETX <bcc> CR**

Where:

i	1 … n.
n	Number of bytes on the block read.
<data i h> <data i l>	i-th byte of the block read from tag. ASCII encoded byte.

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

SOH <add h> <add l> STX '1' '1' (for antenna 2: **'9' '1'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' '1' (for antenna 2: **'9' '1'**) **'0' '1' ETX <bcc> CR**

2.60 Write a Data Block of an ISO 15693 Transponder

This command is used to write a data block of a known (UID) ISO 15693 transponder. Note that the number of bytes of a block depends on the transponder type; for example, the **NXP I CODE SLI** transponder is organized in blocks of 4 bytes, the **Fujitsu MB89R118** transponder is organized in blocks of 8 bytes, for more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x12 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x92 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' '2' with '9' '2'):

SOH <add h> <add l> STX '1' '2' (for antenna 2: **'9' '2'**) **<UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 8 h> <UID 8 l> <blk h> <blk 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 encoded byte.
<pag h> <pag l>	Address of the page to write. ASCII encoded byte.

j	1 … n.
n	Number of bytes of the block to write.
<data j h> <data j l>	j-th byte of the page to write. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' '2' (for antenna 2: **'9' '2'**) **'0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '1' '2' (for antenna 2: **'9' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' '2' (for antenna 2: **'9' '2'**) **'0' '1' ETX <bcc> CR**

2.61 Lock a Data Block of an ISO 15693 Transponder

This command is used to lock a data block of a known (UID) ISO 15693 transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x13 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x93 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' '3' with '9' '3'):

SOH <add h> <add l> STX '1' '3' (for antenna 2: **'9' '3'**) **<UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID 8 h> <UID 8 l> <blk h> <blk 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.
<pag h> <pag l>	Address of the block to lock. ASCII encoder byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' '3' (for antenna 2: **'9' '3'**) **'0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '1' '3' (for antenna 2: **'9' '3'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' '3' (for antenna 2: **'9' '3'**) **'0' '1' ETX <bcc> CR**

2.62 ISO 15693 Transponder 'Get System Info' Command

This command is used to get the system info data block of a known (UID) ISO 15693 transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x14 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x94 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' '4' with '9' '4'):

SOH <add h> <add l> STX '1' '4' (for antenna 2: **'9' '4'**) **<UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID 8 h> <UID 8 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.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' '4' (for antenna 2: **'9' '4'**) **'0' '0' <flg h> <flg l> <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID 8 h> <UID 8 l> <dsf h> <dsf l> <afi h> <afi l> <msbs h> <msbs l> <msnb h> <msnb l> <icr h> <icr l> ETX <bcc> CR**

Where:

<flg h> <flg l>	Info Flags of the tag. ASCII encoded byte. Single bits are dedicated to specify the presence of the following fields (0 → absent, 1 → present): • Bit 7…4: Not used; • Bit 3: IC Reference (1 byte); • Bit 2: Memory Size (2 bytes); • Bit 1: AFI (1 byte); • Bit 0: DSFID (1 byte).
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
i	1 … 8
<dsf h> <dsf l>	DSFID of the tag. ASCII encoded byte. Field present only if bit 0 of Info Flags is set.
<afi h> <afi l>	AFI of the tag. ASCII encoded byte. Field present only if bit 1 of Info Flags is set.
<msbs h> <msbs l>	Memory Size – Block Size in bytes. ASCII encoded byte. 0x00 (1 byte) ... 0x1F (32 bytes). Field present only if bit 2 of Info Flags is set.
<msnb h> <msnb l>	Memory Size – Number of Blocks. ASCII encoded byte. 0x00 (1 block) ... 0xFF (256 blocks). Field present only if bit 2 of Info Flags is set.
<icr h> <icr l>	IC Reference. ASCII encoded byte. Field present only if bit 3 of Info Flags is set.

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '1' '4' (for antenna 2: **'9' '4'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' '4' (for antenna 2: **'9' '4'**) **'0' '1' ETX <bcc> CR**

2.63 ISO 15693 Transponder 'General Protocol' Command

This command allows to send any ISO 15693 general format protocol command (flags field, command code field, parameters fields, application data fields) to a ISO 15693 transponder and to receive, in case of successfull operation, the response of the transponder (flag field, parameters fields, data fields). For more details see the specific transponder data sheet and ISO 15693 protocol. If the 'continuous' mode is enabled, it will be suspended by the execution of this command and will be suspended as long as this command is used; it will be resumed automatically when another type of command will be executed. For devices with 2 antennas, the command code 0x15 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x95 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' '5' with '9' '5'):

SOH <add h> <add l> STX '1' '5' (for antenna 2: **'9' '5'**) **<data 1 h> <data 1 l>… <data i h> <data i l>… <data n h> <data n l> ETX <bcc> CR**

Where:

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

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' '5' (for antenna 2: **'9' '5'**) **'0' '0' <data 1 h> <data 1 l>… <data i h> <data i l>… <data n h> <data n l> ETX <bcc> CR**

Where:

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

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '1' '5' (for antenna 2: **'9' '5'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' '5' (for antenna 2: **'9' '5'**) **'0' '1' ETX <bcc> CR**

2.64 ISO 14443A Transponder 'Inventory' Command

This command is used to get the UID code of a ISO 14443A transponder - **MIFARE Ultralight, MIFARE 1k (UID 4), MIFARE 4k (UID 4), MIFARE 1k (UID 7), MIFARE 4k (UID 7), MIFARE Desfire, MIFARE PLUS 2k, MIFARE Plus 4k, NTAG213/215/216** - that is present near the antenna. For devices with 2 antennas, the command code 0x18 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x98 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' '8' with '9' '8'):

SOH <add h> <add l> STX '1' '8' (for antenna 2: **'9' '8'**) **ETX <bcc> CR**

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

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

Otherwise it answers,

a) if at least one tag is present:

SOH <add h> <add l> STX '1' '8' (for antenna 2: **'9' '8'**) **'0' '0' <type 1 h> <type 1 l> <UID 1 1 h> <UID 1 1 l>… <UID 1 i h> <UID 1 i l>…**

<UID 1 n h> <UID 1 n l> … <type j h> <type j l> <UID j 1 h> <UID j 1 l>… <UID j i h> <UID j i l>… <UID j n h> <UID j n l> … <type m h> <type m l> <UID m 1 h> <UID m 1 l>… <UID m i h> <UID m i l>… <UID m n h> <UID m n l> ETX <bcc> CR

Where:

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

b) if some error is occurred during the transaction:

SOH <add h> <add l> STX '1' '8' (for antenna 2: **'9' '8'**) **'0' '2' ETX <bcc> CR**

c) if no tag is present:

SOH <add h> <add l> STX '1' '8' (for antenna 2: **'9' '8'**) **'0' '1' ETX <bcc> CR**

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

This command is used to get a data block (16 bytes) of a known (UID) **MIFARE 1k/4k (UID 4)** transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x19 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x99 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' '9' with '9' '9'):

SOH <add h> <add l> STX '1' '9' (for antenna 2: **'9' '9'**) **<UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 4 h> <UID 4 l> <kt h> <kt l> <key 1 h> <key 1 l> … <key j h> <key j l> … <key 6 h> <key 6 l> <blk h> <blk l> ETX <bcc> CR**

Where:

i	1 … 4.

<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
<kt h> <kt l>	Key type to access the tag's memory. ASCII encoded byte. • 0x00: Key A; • 0x01: Key B.
j	1 … 6.
<key j h> <key j l>	j-th byte of the key. ASCII encoded byte.
<blk h> <blk l>	Memory block to read. ASCII encoded byte. • 0x00 … 0x3F for MIFARE 1k; • 0x00 … 0xFF for MIFARE 4k.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' '9' (for antenna 2: **'9' '9'**) **'0' '0' <data 1 h> <data 1 l> … <data i h> <data i l> … <data 16 h> <data 16 l> ETX <bcc> CR**

Where:

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

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

SOH <add h> <add l> STX '1' '9' (for antenna 2: **'9' '9'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' '9' (for antenna 2: **'9' '9'**) **'0' '1' ETX <bcc> CR**

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

This command is used to write a data block (16 bytes) of a known (UID) **MIFARE 1k/4k (UID 4)** transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x1A is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x9A is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' 'A' with '9' 'A'):

SOH <add h> <add l> STX '1' 'A' (for antenna 2: **'9' 'A'**) **<UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID 4 h> <UID 4 l> <kt h> <kt l> <key 1 h> <key 1 l> … <key j h> <key j l> … <key 6 h> <key 6 l> <blk h> <blk l> <data 1 h> <data 1 l> … <data k h> <data k l> … <data 16 h> <data 16 l> ETX <bcc> CR**

Where:

i	1 … 4.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
<kt h> <kt l>	Key type to access the tag's memory. ASCII encoded byte. • 0x00: Key A; • 0x01: Key B.
j	1 … 6.
<key j h> <key j l>	j-th byte of the key. ASCII encoded byte.
<blk h> <blk l>	Memory block to write. ASCII encoded byte. • 0x00 … 0x3F for MIFARE 1k; • 0x00 … 0xFF for MIFARE 4k.
k	1 … 16.
<data i h> <data i l>	i-th byte to write in the tag. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' 'A' (for antenna 2: **'9' 'A'**) **'0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '1' 'A' (for antenna 2: **'9' 'A'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' 'A' (for antenna 2: **'9' 'A'**) **'0' '1' ETX <bcc> CR**

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

This command is used to get a data block (16 bytes) of a known (UID) **MIFARE 1k/4k (UID 7)** transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x1D is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x9D is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' 'D' with '9' 'D'):

SOH <add h> <add l> STX '1' 'D' (for antenna 2: **'9' 'D'**) **<UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 7 h> <UID 7 l> <kt h> <kt l> <key 1 h> <key 1 l> … <key j h> <key j l> … <key 6 h> <key 6 l> <blk h> <blk l> ETX <bcc> CR**

Where:

i	1 … 7.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
<kt h> <kt l>	Key type to access the tag's memory. ASCII encoded byte. • 0x00: Key A; • 0x01: Key B.
j	1 … 6.
<key j h> <key j l>	j-th byte of the key. ASCII encoded byte.
<blk h> <blk l>	Memory block to read. ASCII encoded byte. • 0x00 … 0x3F for MIFARE 1k; • 0x00 … 0xFF for MIFARE 4k.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' 'D' (for antenna 2: '9' 'D') '0' '0' <data 1 h> <data 1 l> … <data i h> <data i l> … <data 16 h> <data 16 l> ETX <bcc> CR

Where:

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

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

SOH <add h> <add l> STX '1' 'D' (for antenna 2: '9' 'D') '0' '2' ETX <bcc> CR

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' 'D' (for antenna 2: '9' 'D') '0' '1' ETX <bcc> CR

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

This command is used to write a data block (16 bytes) of a known (UID) **MIFARE 1k/4k (UID 7)** transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x1E is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x9E is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' 'E' with '9' 'E'):

SOH <add h> <add l> STX '1' 'E' (for antenna 2: '9' 'E') <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID 7 h> <UID 7 l> <kt h> <kt l> <key 1 h> <key 1 l> … <key j h> <key j l> … <key 6 h> <key 6 l> <blk h> <blk l> <data 1 h> <data 1 l> … <data k h> <data k l> … <data 16 h> <data 16 l> ETX <bcc> CR

Where:

i	1 … 7.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
<kt h> <kt l>	Key type to access the tag's memory. ASCII encoded byte. • 0x00: Key A; • 0x01: Key B.
j	1 … 6.
<key j h> <key j l>	j-th byte of the key. ASCII encoded byte.
<blk h> <blk l>	Memory block to write. ASCII encoded byte. • 0x00 … 0x3F for MIFARE 1k; • 0x00 … 0xFF for MIFARE 4k.
k	1 … 16.
<data i h> <data i l>	i-th byte to write in the tag. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' 'E' (for antenna 2: **'9' 'E'**) **'0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '1' 'E' (for antenna 2: **'9' 'E'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' 'E' (for antenna 2: **'9' 'E'**) **'0' '1' ETX <bcc> CR**

2.69 Read a Data Page of a MIFARE Ultralight Transponder

This command is used to get a data page (4 bytes) of a known (UID) **MIFARE Ultralight** transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x1B is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x9B is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' 'B' with '9' 'B'):

SOH <add h> <add l> STX '1' 'B' (for antenna 2: **'9' 'B'**) **<UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 7 h> <UID 7 l> <page h> <page l> ETX <bcc> CR**

Where:

i	1 … 7.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
<page h> <page l>	Data page to read. ASCII encoded byte (0x00 … 0x0F).

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' 'B' (for antenna 2: **'9' 'B'**) **'0' '0' <data 1 h> <data 1 l> … <data i h> <data i l> … <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

i	1 … 4.
<data i h> <data i l>	i-th byte read from the 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' 'B' (for antenna 2: **'9' 'B'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' 'B' (for antenna 2: **'9' 'B'**) **'0' '1' ETX <bcc> CR**

2.70 Write a Data Page of a MIFARE Ultralight Transponder

This command is used to write a data page (4 bytes) of a known (UID) **MIFARE Ultralight** transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x1C is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x9C is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' 'C' with '9' 'C'):

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'9' 'C'**) **<UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID 7 h> <UID 7 l> <page h> <page l> <data 1 h> <data 1 l> … <data j h> <data j l> … <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

i	1 … 7.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
<page h> <page l>	Data page to write. ASCII encoded byte (0x00 … 0x0F).
j	1 … 4.
<data i h> <data i l>	i-th byte to write in the tag. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'9' 'C'**) **'0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'9' 'C'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'9' 'C'**) **'0' '1' ETX <bcc> CR**

2.71 Read a Data Page of a NTAG213/215/216 Transponder

This command is used to get a data page (4 bytes) of a known (UID) **NTAG213/215/216** transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x1B is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x9B is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' 'B' with '9' 'B'):

SOH <add h> <add l> STX '1' 'B' (for antenna 2: **'9' 'B'**) **<UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 7 h> <UID 7 l> <page h> <page l> ETX <bcc> CR**

Where:

i	1 … 7.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
<page h> <page l>	Data page to read. ASCII encoded byte. • 0x00 … 0x2C for NTAG213; • 0x00 … 0x86 for NTAG215; • 0x00 … 0xE6 for NTAG216.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' 'B' (for antenna 2: **'9' 'B'**) **'0' '0' <data 1 h> <data 1 l> … <data i h> <data i l> … <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

i	1 … 4.
<data i h> <data i l>	i-th byte read from the 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' 'B' (for antenna 2: **'9' 'B'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' 'B' (for antenna 2: **'9' 'B'**) **'0' '1' ETX <bcc> CR**

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

This command is used to write a data page (4 bytes) of a known (UID) **NTAG213/215/216** transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x1C is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0x9C is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '1' 'C' with '9' 'C'):

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'9' 'C'**) **<UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID 7 h> <UID 7 l> <page h> <page l> <data 1 h> <data 1 l> … <data j h> <data j l> … <data 4 h> <data 4 l> ETX <bcc> CR**

Where:

i	1 … 7.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.
<page h> <page l>	Data page to write. ASCII encoded byte. • 0x00 … 0x2C for NTAG213; • 0x00 … 0x86 for NTAG215; • 0x00 … 0xE6 for NTAG216.
j	1 … 4.

<data i h> <data i l>	i-th byte to write in the tag. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'9' 'C'**) **'0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'9' 'C'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '1' 'C' (for antenna 2: **'9' 'C'**) **'0' '1' ETX <bcc> CR**

2.73 ISO 14443A-4 Transponder 'RATS' Command

This command allows to select and send a RATS (Request for Answer To Select) command to a ISO 14443A transponder – **MIFARE Desfire, MIFARE PLUS 2k, MIFARE Plus 4k** – to switch, in case of successfull operation, from ISO 14443A-3 level to ISO 14443A-4 level. If the 'continuous' mode is enabled, it will be suspended, and the RF field left on, by the execution of this command and will be suspended as long as this command is used; it will be resumed automatically when another type of command will be executed excpect of the generic ISO 14443A-4 command. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x40 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC0 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '0' with 'C' '0'):

SOH <add h> <add l> STX '4' '0' (for antenna 2: **'C' '0'**) **<UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID m h> <UID m l> ETX <bcc> CR**

Where:

i	1 … m.
m	UID length. n = 7 MIFARE Desfire n = 7 MIFARE Plus 2k/4k.
<UID i h> <UID i l>	i-th byte of the UID of the tag. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '0' (for antenna 2: **'C' '0'**) **'0' '0' <rats 1 h> <rats 1 l>… <rats i h> <rats i l>… <rats n h> <rats n l> ETX <bcc> CR**

Where:

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

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '0' (for antenna 2: **'C' '0'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '0' (for antenna 2: **'C' '0'**) **'0' '1' ETX <bcc> CR**

2.74 ISO 14443A-4 Transponder 'Generic Command'

This command allows to send any ISO 14443A-4 general format protocol command to a ISO 14443A-4 transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2

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

The 'master' sends the following command (when using it with the antenna 2, replace '4' '1' with 'C' '1'):

SOH <add h> <add l> STX '4' '1' (for antenna 2: **'C' '1'**) **<data 1 h> <data 1 l>… <data i h> <data i l>… <data n h> <data n l> ETX <bcc> CR**

Where:

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

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '1' (for antenna 2: **'C' '1'**) **'0' '0' <data 1 h> <data 1 l>… <data i h> <data i l>… <data n h> <data n l> ETX <bcc> CR**

Where:

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

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '1' (for antenna 2: **'C' '1'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '1' (for antenna 2: **'C' '1'**) **'0' '1' ETX <bcc> CR**

2.75 MIFARE DESFire Transponder 'Generic Command'

This command allows to send a MIFARE DESFire general format protocol command as described below to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **<data 1 h> <data 1 l>… <data i h> <data i l>… <data n h> <data n l> ETX <bcc> CR**

Where:

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

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' <data 1 h> <data 1 l>… <data i h> <data i l>… <data n h> <data n l> ETX <bcc> CR**

Where:

i	1 … n.

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

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.1 MIFARE DESFire Transponder 'Authenticate' Command

This command allows to send an Authenticate command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' 'A' <keyno h> <keyno l> <keyver h> <keyver l> <keytype h> <keytype l> <key 1 h> <key 1 l> … <key i h> <key i l> … <key n h> <key n l> ETX <bcc> CR**

Where:

<keyno h> <keyno l>	The key number, depends on the currently selected application. See the transponder data sheet for details.. ASCII encoded byte.
<keyver h> <keyver l>	The key version. ASCII encoded byte.
<keytype h> <keytype l>	The key type. ASCII encoded byte: • 0x03: Single DES • 0x04: 2 Key TDES
i	1 … n.

n	Number of bytes of the key to use.
<key i h> <key i l>	i-th byte of the key to use. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.2 MIFARE DESFire Transponder 'AuthenticateISO' Command

This command allows to send an AuthenticateISO command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'1' 'A' <keyno h> <keyno l> <keyver h> <keyver l> <keytype h> <keytype l> <key 1 h> <key 1 l> … <key i h> <key i l> … <key n h> <key n l> ETX <bcc> CR**

Where:

<keyno h> <keyno l>	The key number, depends on the currently selected application. See the transponder data sheet for details.. ASCII encoded byte.
<keyver h> <keyver l>	The key version. ASCII encoded byte.
<keytype h> <keytype l>	The key type. ASCII encoded byte: • 0x03: Single DES • 0x04: 2 Key TDES • 0x05: 3 Key TDES
i	1 … n.
n	Number of bytes of the key to use.
<key i h> <key i l>	i-th byte of the key to use. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.3 MIFARE DESFire Transponder 'AuthenticateAES' Command

This command allows to send an AuthenticateAES command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices

with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'A' 'A' <keyno h> <keyno l> <keyver h> <keyver l> <keytype h> <keytype l> <key 1 h> <key 1 l> … <key i h> <key i l> … <key n h> <key n l> ETX <bcc> CR**

Where:

<keyno h> <keyno l>	The key number, depends on the currently selected application. See the transponder data sheet for details.. ASCII encoded byte.
<keyver h> <keyver l>	The key version. ASCII encoded byte.
<keytype h> <keytype l>	The key type. ASCII encoded byte: • 0x00: AES 128
i	1 … n.
n	Number of bytes of the key to use.
<key i h> <key i l>	i-th byte of the key to use. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.4 MIFARE DESFire Transponder 'FreeMem' Command

This command allows to send a FreeMem command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'6' 'E' ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' <memsize 1 h> <memsize 1 l> <memsize 2 h> <memsize 2 l> <memsize 3 h> memsize 3 l> ETX <bcc> CR**

Where:

<memsize 1 h> <memsize 1 l> … <memsize 3 h> <memsize 3 l>	Size of the free memory, LSB first. ASCII encoded bytes.

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.5 MIFARE DESFire Transponder 'Format' Command

This command allows to send a Format command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'F' 'C' ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.6 MIFARE DESFire Transponder 'GetVersion' Command

This command allows to send a GetVersion command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'6' '0' ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' <vinfo 1 h> <vinfo 1 l> … <vinfo i h> <vinfo i l> … <vinfo n h> <vinfo n l> ETX <bcc> CR**

Where:

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

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.7 MIFARE DESFire Transponder 'ChangeKey' Command

This command allows to send a ChangeKey command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'C' '4' <keyno h> <keyno l> <oldkver h> <oldkver l> <oldktype h> <oldktype l> <oldkey 1 h> <oldkey 1 l> … <oldkey i h> <oldkey i l> … <oldkey n h> <oldkey n l> <newkver h> <newkver l> <newktype h> <newktype l> <newkey 1 h> <newkey 1 l> … <newkey j h> <newkey j l> … <newkey m h> <newkey m l> ETX <bcc> CR**

Where:

<keyno h> <keyno l>	The key number, depends on the currently selected application. See the transponder data sheet for details. ASCII encoded byte.
<oldkver h> <oldkver l>	The old key version. ASCII encoded byte.
<oldktype h> <oldktype l>	The old key type. ASCII encoded byte: • 0x00: AES 128 • 0x03: Single DES • 0x04: 2 Key TDES • 0x05: 3 Key TDES
i	1 … n.
n	Number of bytes of the old key.
<oldkey i h> <oldkey i l>	i-th byte of the old key. ASCII encoded byte.
<newkver h> <newkver l>	The new key version. ASCII encoded byte.
<newktype h> <newktype l>	The new key type. ASCII encoded byte: • 0x00: AES 128 • 0x03: Single DES • 0x04: 2 Key TDES • 0x05: 3 Key TDES
i	1 … n.
n	Number of bytes of the old key.
<newkey i h> <newkey i l>	i-th byte of the new key. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.8 MIFARE DESFire Transponder 'ChangeKeySettings' Command

This command allows to send a ChangeKeySettings command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'5' '4' <keyset h> <ketset l> ETX <bcc> CR**

Where:

<keyset h> <keyset l>	The key setting, depends on the selected application. See the transponder data sheet for more details. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.9 MIFARE DESFire Transponder 'CreateApplication' Command

This command allows to send a CreateApplication command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'C' 'A' <aid 1 h> <aid 1 l> <aid 2 h> <aid 2 l> <aid 3 h> <aid 3 l> <keyset1 h> <keyset1 l> <keyset2 h> <keyset2 l> ETX <bcc> CR**

Where:

<aid 1 h> <aid 1 l> … <aid 3 h> <aid 3 l>	The application identifier, LSB first. ASCII encoded bytes.
<keyset1 h> <keyset1 l>	The key setting 1. See the transponder data sheet for more details. ASCII encoded byte.
<keyset2 h> <keyset2 l>	The key setting 2. See the transponder data sheet for more details. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.10 MIFARE DESFire Transponder 'DeleteApplication' Command

This command allows to send a DeleteApplication command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'D' 'A' <aid 1 h> <aid 1 l> <aid 2 h> <aid 2 l> <aid 3 h> <aid 3 l> ETX <bcc> CR**

Where:

<aid 1 h> <aid 1 l> … <aid 3 h> <aid 3 l>	The application identifier, LSB first. ASCII encoded bytes.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.11 MIFARE DESFire Transponder 'SelectApplication' Command

This command allows to send a SelectApplication command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'5' 'A' <aid 1 h> <aid 1 l> <aid 2 h> <aid 2 l> <aid 3 h> <aid 3 l> ETX <bcc> CR**

Where:

<aid 1 h> <aid 1 l> … <aid 3 h> <aid 3 l>	The application identifier, LSB first. ASCII encoded bytes.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.12 MIFARE DESFire Transponder 'CreateStdDataFile' Command

This command allows to send a CreateStdDataFile command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'C' 'D' <fileid h> <fileid l> <comm h> <comm l> <rights 1 h> <rights 1 l> <rights 2 h> <rights 2 l> <fsize 1 h> <fsize 1 l> <fsize 2 h> <fsize 2 l> <fsize 3 h> <fsize 3 l> ETX <bcc> CR**

Where:

<fileid h> <fileid l>	The file identifier. ASCII encoded byte.
<comm h> <comm l>	The communication mode. ASCII encoded bytes: • 0x00: Plain • 0x10: MAC • 0x30: ENC

<rights 1 h> <rights 1 l> <rights 2 h> <rights 2 l>	The file access rights, LSB first. See the set access conditions table in the transponder data sheet for details. ASCII encoded bytes.
<fsize 1 h> <fsize 1 l> … <fsize 3 h> <fsize 1 l>	The file size in bytes, LSB first. ASCII encoded bytes.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.13 MIFARE DESFire Transponder 'CreateBackupDataFile' Command

This command allows to send a CreateBackupDataFile command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'C' 'B' <fileid h> <fileid l> <comm h> <comm l> <rights 1 h> <rights 1 l> <rights 2 h>**

<rights 2 l> <fsize 1 h> <fsize 1 l> <fsize 2 h> <fsize 2 l> <fsize 3 h> <fsize 3 l> ETX <bcc> CR

Where:

<fileid h> <fileid l>	The file identifier. ASCII encoded byte.
<comm h> <comm l>	The communication mode. ASCII encoded bytes: • 0x00: Plain • 0x10: MAC • 0x30: ENC
<rights 1 h> <rights 1 l> <rights 2 h> <rights 2 l>	The file access rights, LSB first. See the set access conditions table in the transponder data sheet for details. ASCII encoded bytes.
<fsize 1 h> <fsize 1 l> … <fsize 3 h> <fsize 1 l>	The file size in bytes, LSB first. ASCII encoded bytes.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.14 MIFARE DESFire Transponder 'CreateValueFile' Command

This command allows to send a CreateValueFile command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO

14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'C' 'C' <fileid h> <fileid l> <comm h> <comm l> <rights 1 h> <rights 1 l> <rights 2 h> <rights 2 l> <low 1 h> <low 1 l> … <low i h> <low i l> … <low 4 h> <low 4 l> <up 1 h> <up 1 l> … <up i h> <up i l> … <up 4 h> <up 4 l> <val 1 h> < val 1 l> … < val i h> < val i l> … < val 4 h> < val 4 l> <lim h> <lim l> ETX <bcc> CR**

Where:

<fileid h> <fileid l>	The file identifier. ASCII encoded byte.
<comm h> <comm l>	The communication mode. ASCII encoded bytes: • 0x00: Plain • 0x10: MAC • 0x30: ENC
<rights 1 h> <rights 1 l> <rights 2 h> <rights 2 l>	The file access rights, LSB first. See the set access conditions table in the transponder data sheet for details. ASCII encoded bytes.
i	1 … 4
<low i h> <low i l>	Lower limit of the file, LSB first. ASCII encoded bytes.
<up i h> <up i l>	Upper limit of the file, LSB first. ASCII encoded bytes.
<val i h> <val i l>	Current value of the file, LSB first. ASCII encoded bytes.
<lim h> <lim l>	LimitedCredit and GetValue commands enable. See the transponder data sheet for details. ASCII encoded bytes.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.15 MIFARE DESFire Transponder 'DeleteFile' Command

This command allows to send a DeleteFile command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'D' 'F' <fileid h> <fileid l> ETX <bcc> CR**

Where:

<fileid h> <fileid l>	The file identifier. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.16 MIFARE DESFire Transponder 'ReadData' Command

This command allows to send a ReadData command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'B' 'D' <fileid h> <fileid l> <omm h> <comm l> <offset 1 h> <offset 1 l> <offset 2 h> <offset 2 l> <offset 3 h> <offset 3 l> <length 1 h> <length 1 l> <length 2 h> <length 2 l> <length 3 h> <length 3 l> ETX <bcc> CR**

Where:

<fileid h> <fileid l>	The file identifier. ASCII encoded byte.
<comm h> <comm l>	The communication mode. ASCII encoded bytes: • 0x00: Plain • 0x10: MAC • 0x30: ENC
<offset 1 h> <offset 1 l> … <offset 3 h> <offset 3 l>	The starting position for the read operation, LSB first. ASCII encoded bytes.
<length 1 h> < length 1 l> … < length 3 h> < length 3 l>	The number of bytes to read, LSB first. ASCII encoded bytes.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' <data 1 h> <data 1 l> … <data i h> <data i l> … <data n h> <data n l> ETX <bcc> CR**

Where:

i	1 … n
n	Number of bytes read.
<data i h> <data i l>	i-th byte read. ASCII encoded byte.

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.17 MIFARE DESFire Transponder 'WriteData' Command

This command allows to send a WriteData command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'3' 'D' <fileid h> <fileid l> <omm h> <comm l> <offset 1 h> <offset 1 l> <offset 2 h> <offset 2 l> <offset 3 h> <offset 3 l> <length 1 h> <length 1 l> <length 2 h> <length 2 l> <length 3 h> <length 3 l> <data 1 h>**

<data 1 l> … <data i h> <data i l> … <data n h> <data n l> ETX <bcc> CR

Where:

<fileid h> <fileid l>	The file identifier. ASCII encoded byte.
<comm h> <comm l>	The communication mode. ASCII encoded bytes: • 0x00: Plain • 0x10: MAC • 0x30: ENC
<offset 1 h> <offset 1 l> … <offset 3 h> <offset 3 l>	The starting position for the read operation, LSB first. ASCII encoded bytes.
<length 1 h> < length 1 l> … < length 3 h> < length 3 l>	The number of bytes to read, LSB first. ASCII encoded bytes.
i	1 … n
n	Number of bytes to write.
<data i h> <data i l>	i-th byte to write. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.18 MIFARE DESFire Transponder 'GetValue' Command

This command allows to send a GetValue command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'6' 'C' <fileid h> <fileid l> <omm h> <comm l> ETX <bcc> CR**

Where:

<fileid h> <fileid l>	The file identifier. ASCII encoded byte.
<comm h> <comm l>	The communication mode. ASCII encoded bytes: • 0x00: Plain • 0x10: MAC • 0x30: ENC

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' <val 1 h> < val 1 l> … < val i h> < val i l> … < val 4 h> < val 4 l> ETX <bcc> CR**

Where:

<fileid h> <fileid l>	The file identifier. ASCII encoded byte.

<comm h> <comm l>	The communication mode. ASCII encoded bytes: • 0x00: Plain • 0x10: MAC • 0x30: ENC
i	1…4
<val i h> <val i l>	Current value of the file, LSB first. ASCII encoded bytes.

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.19 MIFARE DESFire Transponder 'Credit' Command

This command allows to send a Credit command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' 'C' <fileid h> <fileid l> <omm h> <comm l> <data 1 h> < data 1 l> … < data i h> < data i l> … < data 4 h> < data 4 l> ETX <bcc> CR**

Where:

<fileid h> <fileid l>	The file identifier. ASCII encoded byte.
<comm h> <comm l>	The communication mode. ASCII encoded bytes: • 0x00: Plain • 0x10: MAC • 0x30: ENC
i	1…4

<data i h> <data i l>	The value to be credited, LSB first. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.20 MIFARE DESFire Transponder 'LimitedCredit' Command

This command allows to send a LimitedCredit command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'1' 'C' <fileid h> <fileid l> <omm h> <comm l> <data 1 h> < data 1 l> … < data i h> < data i l> … < data 4 h> < data 4 l> ETX <bcc> CR**

Where:

<fileid h> <fileid l>	The file identifier. ASCII encoded byte.
<comm h> <comm l>	The communication mode. ASCII encoded bytes:

	• 0x00: Plain • 0x10: MAC • 0x30: ENC
i	1…4
<data i h> <data i l>	The value to be credited, LSB first. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.21 MIFARE DESFire Transponder 'Debit' Command

This command allows to send a Debit command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'D' 'C' <fileid h> <fileid l> <omm h> <comm l> <data 1 h> < data 1 l> … < data i h> < data i l> … < data 4 h> < data 4 l> ETX <bcc> CR**

Where:

<fileid h> <fileid l>	The file identifier. ASCII encoded byte.
<comm h> <comm l>	The communication mode. ASCII encoded bytes: • 0x00: Plain • 0x10: MAC • 0x30: ENC
i	1…4
<data i h> <data i l>	The value to be debited, LSB first. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.22 MIFARE DESFire Transponder 'CommitTransaction' Command

This command allows to send a CommitTransaction command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'C' '7' ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.75.23 MIFARE DESFire Transponder 'AbortTransaction' Command

This command allows to send a AbortTransaction command to a **MIFARE DESFire** transponder and to receive, in case of successfull operation, the response of the transponder. The transponder must be switched to ISO 14443A-4 level before using the RATS command described before. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x42 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '2' with 'C' '2'):

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'A' '7' ETX <bcc> CR**

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '0' '0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '4' '2' (for antenna 2: **'C' '2'**) **'0' '1' ETX <bcc> CR**

2.76 ISO 14443B Transponder 'Inventory' Command

This command is used to get the UID code of a ISO 14443B transponder – **SR176, SRI512** - that is present near the antenna. For devices with 2 antennas, the command code 0x20 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xA0 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '2' '0' with 'A' '0'):

SOH <add h> <add l> STX '2' '0' (for antenna 2: **'A' '0'**) **ETX <bcc> CR**

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

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

Otherwise it answers,

a) if a tag is present:

SOH <add h> <add l> STX '2' '0' (for antenna 2: **'A' '0'**) **'0' '0' <type 1 h> <type 1 l> <UID 1 1 h> <UID 1 1 l>… <UID 1 i h> <UID 1 i l>… <UID 1 n h> <UID 1 n l> … <type j h> <type j l> <UID j 1 h> <UID j 1 l>… <UID j i h> <UID j i l>… <UID j n h> <UID j n l> … <type m h> <type m l> <UID m 1 h> <UID m 1 l>… <UID m i h> <UID m i l>… <UID m n h> <UID m n l> ETX <bcc> CR**

Where:

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

b) if some error is occurred during the transaction:

SOH <add h> <add l> STX '2' '0' (for antenna 2: **'A' '0'**) **'0' '2' ETX <bcc> CR**

c) if no tag is present:

SOH <add h> <add l> STX '2' '0' (for antenna 2: **'A' '0'**) **'0' '1' ETX <bcc> CR**

2.77 Read a Data Block of a SR 176 Transponder

This command is used to get a data block (2 bytes) of a known (UID) **SR 176** transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x21 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xA1 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '2' '1' with 'A' '1'):

SOH <add h> <add l> STX '2' '1' (for antenna 2: 'A' '1') <UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID 8 h> <UID 8 l> <blk h> <blk 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.
<blk h> <blk l>	Data block to read. ASCII encoded byte (0x00 … 0x0F).

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '2' '1' (for antenna 2: **'A' '1'**) **'0' '0' <data 1 h> <data 1 l> … <data i h> <data i l> … <data 2 h> <data 2 l> ETX <bcc> CR**

Where:

i	1 … 2.
<data i h> <data i l>	i-th byte read from the 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 '2' '1' (for antenna 2: **'A' '1'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present

SOH <add h> <add l> STX '2' '1' (for antenna 2: **'A' '1'**) **'0' '1' ETX <bcc> CR**

2.78 Write a Data Block of a SR 176 Transponder

This command is used to write a data block (2 bytes) of a known (UID**) SR 176** transponder. For more details see the specific transponder data sheet. For devices with 2 antennas, the command code 0x22 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xA2 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '2' '2' with 'A' '2'):

SOH <add h> <add l> STX '2' '2' (for antenna 2: 'A' '2') <UID 1 h> <UID 1 l> … <UID i h> <UID i l> … <UID 8 h> <UID 8 l> <blk h> <blk l> <data 1 h> <data 1 l> <data 2 h> <data 2 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.

<blk h> <blk l>	Data block to write. ASCII encoded byte (0x00 … 0x0F).
k	1 … 2.
<data i h> <data i l>	i-th byte to write in the tag. ASCII encoded byte.

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

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

Otherwise it answers,

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

SOH <add h> <add l> STX '2' '2' (for antenna 2: **'A' '2'**) **'0' '0' ETX <bcc> CR**

b) if the addressed tag is present but errors occurred:

SOH <add h> <add l> STX '2' '2' (for antenna 2: **'A' '2'**) **'0' '2' ETX <bcc> CR**

c) if the addressed tag is not present:

SOH <add h> <add l> STX '2' '2' (for antenna 2: **'A' '2'**) **'0' '1' ETX <bcc> CR**

2.79 PicoPass Transponders 'Inventory' Command

This command is used to get the UID code of the identified PicoPass transponders that are present near the antenna/s. For devices with 2 antennas, the command code 0x48 is used to work with the antenna nr 1, like for devices with 1 antenna, while the command code 0xC8 is used to work with antenna nr 2.

The 'master' sends the following command (when using it with the antenna 2, replace '4' '8' with 'C' '8'):

SOH <add h> <add l> STX '4' '8' (for antenna 2: **'C' '8'**) **ETX <bcc> CR**

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

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

Otherwise it answers,

a) if at least one tag is present:

SOH <add h> <add l> STX '4' '8' (for antenna 2: **'C' '8'**) **'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**

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.

b) if some error is occurred during the transaction:

SOH <add h> <add l> STX '4' '8' (for antenna 2: **'C' '8'**) **'0' '2' ETX <bcc> CR**

c) if no tag is present:

SOH <add h> <add l> STX '4' '8' (for antenna 2: **'C' '8'**) **'0' '1' ETX <bcc> CR**

2.80 ISO 18000-63 Transponder 'Inventory' Command

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

The 'master' sends the following command:

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

If the addressed **BLUEBOX** 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:

SOH <add h> <add l> STX '1' '8' '0' '0' <ID 1 1 h> <ID 1 1 l> … <ID 1 i h> <ID 1 i l> … <ID 1 16 h> <ID 1 16 l> '0' <ant 1> … <ID j 1 h>

<ID j 1 l>… <ID j i h> <ID j i l>… <ID j 16 h> <ID j 16 l> '0' <ant j> … <ID n 1 h> <ID n 1 l> … <ID n i h> <ID n i l> … <ID n 16 h> <ID n 16 l> '0' <ant n> ETX <bcc> CR

Where:

i	1 … m.
m	ID length.
j	1 … n.
n	Number of identified tags.
<ID j i h> <ID j i l>	i-th byte of the ID of the j-th identified tag. ASCII encoded byte.
<ant j>	Reading antenna for the j-th identified tag **(optional parameter present only if the reading antenna information flag in the general parameters is active, see the reader user manual for more info)**. 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

c) if no tag is present

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

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

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

If the addressed **BLUEBOX** 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:

SOH <add h> <add l> STX '1' '8' '0' '0' <ID 1 1 h> <ID 1 1 l> … <ID 1 i h> <ID 1 i l> … <ID 1 16 h> <ID 1 16 l> '0' <ant 1> … <ID j 1 h>

<ID j 1 l>… <ID j i h> <ID j i l>… <ID j 16 h> <ID j 16 l> '0' <ant j> … <ID n 1 h> <ID n 1 l> … <ID n i h> <ID n i l> … <ID n 16 h> <ID n 16 l> <RSSI Q h> <RSSI Q l> <RSSI I h> <RSSI I l> '0' <ant n> ETX <bcc> CR

Where:

i	1 … m.
m	ID length.
J	1 … n.
n	Number of identified tags.
<ID j i h> <ID j i l>	i-th byte of the ID of the j-th identified tag. ASCII encoded byte.
<RSSI Q h> <RSSI Q l>	The RSSI Q-channel value in dB. ASCII encoded byte.
<RSSI I h> <RSSI I l>	The RSSI I-channel value in dB. ASCII encoded byte.
<ant j>	Reading antenna for the j-th identified tag **(optional parameter present only if the reading antenna information flag in the general parameters is active, see the reader user manual for more info)**. 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

c) if no tag is present

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

2.81 Program EPC of an ISO 18000-63 Transponder

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

The 'master' sends the following command:

SOH <add h<add l> STX '1' 'E' <ID 1 h> <ID 1 l>… <ID i h> <ID i l>… <ID 16 h> <ID 16 l> <pwd 1 h> <pwd 1 l> … <pwd j h> <pwd j l> …

<pwd 4 h> <pwd 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	ID length.
<ID i h> <ID i l>	i-th byte of the ID 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.
w	1 … 2.
k	1 … n.
n	Number of blocks to be written, range 0 … 31.
<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' 'E' '0' '0' ETX <bcc> CR

b) if the addressed tag is present but errors occurred

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

c) if the addressed tag is not present

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

2.82 Read Data of an ISO 18000-63 Transponder

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

The 'master' sends the following command:

SOH <add h<add l> STX '1' '9' <ID 1 h> <ID 1 l>… <ID i h> <ID i l>… <ID 16 h> <ID 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	ID length.
<ID i h> <ID i l>	i-th byte of the ID 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 read. ASCII character: • '0': Reserved. • '1': EPC. • '2': TID. • '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 blocks to be read. ASCII encoded byte (1 … 64).

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

2.83 Write Data of an ISO 18000-63 Transponder

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

The 'master' sends the following command:

SOH <add h<add l> STX '1' 'A' <ID 1 h> <ID 1 l>… <ID i h> <ID i l>… <ID 16 h> <ID 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> <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	ID length.

<ID i h> <ID i l>	i-th byte of the ID 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': EPC. • '2': TID. • '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.
<nblk h> <nblk l>	Number of block to be write. ASCII encoded byte (1 … 64).
w	1 … 2.
k	1 … n.
n	Number of blocks to be written, range 1 … 64.
<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

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

The 'master' sends the following command:

SOH <add h<add l> STX '1' 'D' <ID 1 h> <ID 1 l>… <ID i h> <ID i l>… <ID 16 h> <ID 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> <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	ID length.
<ID i h> <ID i l>	i-th byte of the ID 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': EPC. • '2': TID. • '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.
<nblk h> <nblk l>	Number of block to be write. ASCII encoded byte (1 … 64).
w	1 … 2.
k	1 … n.
n	Number of blocks to be written, range 1 … 64.
<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' 'D' '0' '0' ETX <bcc> CR

b) if the addressed tag is present but errors occurred

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

c) if the addressed tag is not present

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

2.84 Lock Data of an ISO 18000-63 Transponder

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

The 'master' sends the following command:

SOH <add h<add l> STX '1' 'B' <ID 1 h> <ID 1 l>… <ID i h> <ID i l>… <ID 16 h> <ID 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	ID length.
<ID i h> <ID i l>	i-th byte of the ID 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 accessibile from all states and may never be locked; • '2': Accessible only from the secured state; • '3': Not accessibile from any state; • '4': No change.
<acc>	Access password lock property. ASCII character: • '0': Accessible from all states; • '1': Permanently accessibile from all states and may never be locked; • '2': Accessible only from the secured state; • '3': Not accessibile 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

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

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

The 'master' sends the following command:

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

Where:

i	1 … m.
m	ID length.
<ID i h> <ID i l>	i-th byte of the ID 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

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

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

The 'master' sends the following command:

SOH <add h<add l> STX '2' '0' <ID 1 h> <ID 1 l>… <ID i h> <ID i l>… <ID 16 h> <ID 16 l> <pwd 1 h> <pwd 1 l> … <pwd j h> <pwd j l> … <pwd 4 h> <pwd 4 l> ETX <bcc> CR

Where:

i	1 … m.
m	ID length.
<ID i h> <ID i l>	i-th byte of the ID 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.

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 command has been successfully executed

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

Where:

<QT ctrl 1 h> <QT ctrl 1 l>	1st byte (MSB) of the QT control field read from tag. ASCII encoded byte.
<QT ctrl 2 h> <QT ctrl 2 l>	2nd byte (LSB) of the QT control field read from tag. ASCII encoded byte.

b) if some error is occurred during the transaction

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

c) if the addressed tag is not present

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

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

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

The 'master' sends the following command:

SOH <add h<add l> STX '2' '1' <ID 1 h> <ID 1 l>… <ID i h> <ID i l>… <ID 16 h> <ID 16 l> <pwd 1 h> <pwd 1 l> … <pwd j h> <pwd j l> … <pwd 4 h> <pwd 4 l> <persistence h> <persistence l> <QT ctrl 1 h> <QT ctrl 1 l> <QT ctrl 2 h> <QT ctrl 2 l> ETX <bcc> CR

Where:

i	1 … m.
m	ID length.
<ID i h> <ID i l>	i-th byte of the ID 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.
<persistence h> <persistence l>	Indicates wether the QT control is written to non volatile or volatile memory. ASCII encoded byte: • 0x00: Write to volatile memory. • 0x01: Write to non volatile memory.
<QT ctrl 1 h> <QT ctrl 1	1st byte (MSB) of the QT control field to write to the tag.

l>	ASCII encoded byte.
<QT ctrl 2 h> <QT ctrl 2 l>	2nd byte (LSB) of the QT control field to write to the tag. ASCII encoded byte.

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

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

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

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

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

b) if some error is occurred during the transaction

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

c) if the addressed tag is not present

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

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

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

The 'master' sends the following command:

SOH <add h<add l> STX '2' '2' <chip h> <chip l> <ID 1 h> <ID 1 l>… <ID i h> <ID i l>… <ID 16 h> <ID 16 l> <pwd 1 h> <pwd 1 l> … <pwd j h> <pwd j l> … <pwd 4 h> <pwd 4 l> ETX <bcc> CR

Where:

<chip h> <chip l>	The chip code. ASCII encoded byte: • 0x02: Magnus S2. • 0x03: Magnus S3.
i	1 … m.
m	ID length.
<ID i h> <ID i l>	i-th byte of the ID 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.

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 command has been successfully executed

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

Where:

<sens hh> <sens hl> <sens lh> <sens ll>	The sensor code read from tag. ASCII encoded word.

b) if some error is occurred during the transaction

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

c) if the addressed tag is not present

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

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

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

The 'master' sends the following command:

SOH <add h<add l> STX '2' '3' <chip h> <chip l> <ID 1 h> <ID 1 l>… <ID i h> <ID i l>… <ID 16 h> <ID 16 l> <pwd 1 h> <pwd 1 l> … <pwd j h> <pwd j l> … <pwd 4 h> <pwd 4 l> <match h> <match l> <thhold h> <thhold l> ETX <bcc> CR

Where:

<chip h> <chip l>	The chip code. ASCII encoded byte: • 0x02: Magnus S2. • 0x03: Magnus S3.
i	1 … m.
m	ID length.
<ID i h> <ID i l>	i-th byte of the ID 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.
<match h> <match l>	The RSSI threshold match criteria. ASCII encoded byte: • 0x00: Match if code is <= threshold. • 0x01: Match if code is > threshold.
<thhold h> <thhold l>	The RSSI threshold in the range 0 … 31. 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 and the command has been successfully executed

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

Where:

<rssi h> <rssi l>	The on-chip RSSI read from tag. ASCII encoded byte.

b) if some error is occurred during the transaction

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

c) if the addressed tag is not present

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

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

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

The 'master' sends the following command:

SOH <add h<add l> STX '2' '4' <chip h> <chip l> <ID 1 h> <ID 1 l>… <ID i h> <ID i l>… <ID 16 h> <ID 16 l> <pwd 1 h> <pwd 1 l> … <pwd j h> <pwd j l> … <pwd 4 h> <pwd 4 l> ETX <bcc> CR

Where:

<chip h> <chip l>	The chip code. ASCII encoded byte: • 0x03: Magnus S3.
i	1 … m.
m	ID length.
<ID i h> <ID i l>	i-th byte of the ID 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.

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 command has been successfully executed

SOH <add h> <add l> STX '2' '4' '0' '0' <temp hh> <temp hl> <temp lh> <temp ll> ETX <bcc> CR

Where:

<temp hh> <temp hl> <temp lh> <temp ll>	The temperature code read from tag. ASCII encoded word.

b) if some error is occurred during the transaction

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

c) if the addressed tag is not present

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

2.91 'Spontaneous' Message

In 'continuous' mode, if the 'spontaneous' feature is set on (see parameters), the **BLUEBOX** will send the following message on the serial line and on the Ethernet channel (if available) every time that it will find a 'new' tag,

a) for LF devices with only 1 antenna:

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

Where:

i	1 … n.
n	Number of bytes of the tag code: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE
<code i h> <code i l>	i-th byte of the code of the identified tag. ASCII encoded byte.
<bcc>	Block check character or checksum calculated as 'xor' of the previous characters starting from STX applying the following rule: if <bcc> = STX or <bcc> = CR, then <bcc> := <bcc>+1 (increment of 1)

b) for LF devices with 2 antennas:

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

Where:

i	1 … n.

n	Number of bytes of the tag code: • 5: UNIQUE, BLUEBOX SHORT • 10: BLUEBOX MEDIUM • 20: BLUEBOX LARGE
<code i h> <code i l>	i-th byte of the code of the identified tag. ASCII encoded byte.
<ant h> <ant l>	The antenna number which have identified the tag: ASCII encoded byte: • 0x01: Antenna nr. 1 • 0x02: Antenna nr. 2
<bcc>	Block check character or checksum calculated as 'xor' of the previous characters starting from STX applying the following rule: if <bcc> = STX or <bcc> = CR, then <bcc> := <bcc>+1 (increment of 1)

c) for HF devices with only 1 antenna:

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

Where:

<type h> <type l>	Transponder type.
i	1 … n (the UID length).
<UID i h> <UID i l>	i-th byte of the UID of the identified tag. ASCII encoded byte.
<bcc>	Block check character or checksum calculated as 'xor' of the previous characters starting from STX applying the following rule: if <bcc> = STX or <bcc> = CR, then <bcc> := <bcc>+1 (increment of 1)

d) for HF devices with 2 antennas:

STX <type h> <type l> <UID 1 h> <UID 1 l>… <UID i h> <UID i l>… <UID n h> <UID n l> <ant h> <ant l> ETX <bcc> CR

Where:

<type h> <type l>	Transponder type.
i	1 … n (the UID length).

<UID i h> <UID i l>	i-th byte of the UID of the identified tag. ASCII encoded byte.
<ant h> <ant l>	The antenna number which have identified the tag: ASCII encoded byte: • 0x01: Antenna nr. 1 • 0x02: Antenna nr. 2
<bcc>	Block check character or checksum calculated as 'xor' of the previous characters starting from STX applying the following rule: if <bcc> = STX or <bcc> = CR, then <bcc> := <bcc>+1 (increment of 1)

e) for UHF devices:

STX <tag h> <tag l> <ID 1 h> <ID 1 l> … <ID i h> <ID i l> … <ID m h> <ID m l> <RSSI Q h> <RSSI Q l> <RSSI I h> <RSSI I l> <ant h> <ant l> dir h> <dir l> <tm 1 h> <tm 1 l> … <tm 7 h> <tm 7 l> ETX <bcc> CR

Where:

i	1 … m.
m	ID length.
<tag h> <tag l>	Transponder type **(optional parameter present only if the tag type information flag in the general parameters is active, see the reader user manual for more info)**. ASCII encoded byte: • 0x02: ISO 18000-63 (EPC C1G2).
<ID i h> <ID i l>	i-th byte of the ID of the identified tag. ASCII encoded byte.
<RSSI Q h> <RSSI Q l> <RSSI I h> <RSSI I l>	RSSI Q and I channel info in dB of the identified tag. **(optional parameter present only if the RSSI information flag in the RF configuration parameters is active, see the reader user manual for more info)**. ASCII encoded bytes.
<ant h> <ant l>	Reading antenna **(optional parameter present only if the reading antenna information flag in the general parameters is active, see the reader user manual for more info)**. ASCII character: • 0x01 -> Antenna 1. • 0x02 -> Antenna 2. • 0x03 -> Antenna 3.

	• 0x04 -> Antenna 4.
<dir h> <dir l>	Gate crossing direction for the identified tag **(optional parameter present only if 'gate' mode is active, see the reader user manual for more info)**. ASCII character: • 0x01 -> Crossing from input 1 to input 2. • 0x02 -> Crossing from input 2 to input 1.
<tm 1 h> <tm 1 l> … <tm 7 h> <tm 7 l>	Timestamp for the identified tag **(optional parameter present only if the reading timestamp information flag in the general parameters is active, see the reader user manual for more info)**. ASCII encoded byte array of the BCD encoded timestamp with the format yyyyMMddhhmmss.
<bcc>	Block check character or checksum calculated as 'xor' of the previous characters starting from STX applying the following rule: if <bcc> = STX or <bcc> = CR, then <bcc> := <bcc>+1 (increment of 1).

3 Examples

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

3.1 Read Firmware Version

This command is used to get the firmware version of the **BLUEBOX**.

The 'master' sends the following command:

A: <SOH>FF<STX>34<ETX><BELL><CR>

H: 01 46 46 02 33 34 03 07 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…5	2	33 34	34	Command code = 0x34
6	1	03	ETX	End of Text
7	1	07	BELL	Packet checksum (BCC)
8	1	0D	CR	Carriage Return

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

A: <SOH>FF<NAK><DC4><CR>

H: 01 46 46 15 14 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
3	1	15	NAK	Not Acknoledge
4	1	14	DC4	Packet checksum (BCC)
5	1	0D	CR	Carriage Return

Otherwise it answers:

A: <SOH>FF<STX>34424C5545424F585F4C4620312E303020<ETX>}<CR>

H: 01 46 46 02 33 34 34 32 34 43 35 35 34 35 34 32 34 46 35 38 35 46 34 43 34 36 32 30 33 31 32 45 33 30 33 30 32 30 03 7D 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…5	2	33 34	34	Command code = 0x34
6…38	32	34 32 34 43 35 35 34 35 34 32 34 46 35 38 35 46 34 43 34 36 32 30 33 31 32 45 33 30 33 30 32 30	424C5545424F585F4C4620312E303020	FW = BLUEBOX_LF 1.00
39	1	03	ETX	End of Text
40	1	7D	}	Packet checksum (BCC)
41	1	0D	CR	Carriage Return

3.2 Buffer Data Request

This command sends back the code of the eventual transponder that is present in the buffer. When 'continuous' mode is enabled, the reply is immediate because the **BLUEBOX** sends back the data 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>FF<ENQ><ENQ><CR>

Hexadecimal: 01 46 46 05 05 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	05	ENQ	Enquiry
4	1	05	ENQ	Packet checksum (BCC)
5	1	0D	CR	Carriage Return

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

<SOH>FF<NAK><DC4><CR>

Hexadecimal: 01 46 46 15 14 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	15	NAK	Not Acknoledge
4	1	14	DC4	Packet checksum (BCC)

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
5	1	0D	CR	Carriage Return

Otherwise,

a) for LF devices with only 1 antenna it answers,

a1) if the antenna have identified a transponder:

<SOH>FF<STX>0011223344<ETX><NULL><CR>

Hexadecimal: 01 46 46 02 30 30 31 31 32 32 33 33 34 34 03 00 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…14	10	30 30 31 31 32 32 33 33 34 34	0011223344	Tag code = 00 11 22 33 44
15	1	03	ETX	End of Text
16	1	00	NULL	Packet checksum (BCC)
17	1	0D	CR	Carriage Return

a2) if no transponder is identified by the antenna:

<SOH>FF<STX>0000000000<ETX><NULL><CR>

Hexadecimal: 01 46 46 02 30 30 30 30 30 30 30 30 30 30 03 00 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…14	10	30 30 30 30 30 30 30 30 30 30	0000000000	Null tag
15	1	03	ETX	End of Text
16	1	00	NULL	Packet checksum (BCC)
17	1	0D	CR	Carriage Return

b) for LF devices with 2 antennas, it answers,

b1) if both antennas have identified a transponder:

<SOH>FF<STX>0011223344-0102030405<ETX><STX><CR>

Hexadecimal: 01 46 46 02 30 30 31 31 32 32 33 33 34 34 2D 30 31 30 32 30 33 30 34 30 35 03 02 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…14	10	30 30 31 31 32 32 33 33 34 34	0011223344	Tag code = 00 11 22 33 44
15	1	2D	-	Separator antenna 1 /2
16…26	10	30 31 30 32 30 33 30 34 30	0102030405	Tag code = 01 02 03 04 05

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
		35		
27	1	03	ETX	End of Text
28	1	02	STX	Packet checksum (BCC)
29	1	0D	CR	Carriage Return

b2) if only antenna 1 have identified a transponder:

<SOH>FF<STX>0011223344-0000000000<ETX>-<CR>

Hexadecimal: 01 46 46 02 30 30 31 31 32 32 33 33 34 34 2D 30 30 30 30 30 30 30 30 30 30 03 2D 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…14	10	30 30 31 31 32 32 33 33 34 34	0011223344	Tag code = 00 11 22 33 44
15	1	2D	-	Separator antenna 1 /2
16…26	10	30 30 30 30 30 30 30 30 30 30	0000000000	Null tag
27	1	03	ETX	End of Text
28	1	05	ENQ	Packet checksum (BCC)
29	1	0D	CR	Carriage Return

b3) if only antenna 2 have identified a transponder:

<SOH>FF<STX>0000000000-0102030405<ETX><STX><CR>

Hexadecimal: 01 46 46 02 30 30 30 30 30 30 30 30 30 30 2D 30 31 30 32 30 33 30 34 30 35 03 02 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…14	10	30 30 30 30 30 30 30 30 30 30	0000000000	Null tag
15	1	2D	-	Separator antenna 1 /2
16…26	8	30 31 30 32 30 33 30 34 30 35	0102030405	Tag code = 01 02 03 04 05
27	1	03	ETX	End of Text
28	1	07	BELL	Packet checksum (BCC)
29	1	0D	CR	Carriage Return

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

<SOH>FF<STX>0000000000-0000000000<ETX>-<CR>

Hexadecimal: 01 46 46 02 30 30 30 30 30 30 30 30 30 30 2D 30 30 30 30 30 30 30 30 30 30 03 2D 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
4…14	10	30 30 30 30 30 30 30 30 30 30	0000000000	Null tag
15	1	2D	-	Separator antenna 1 /2
16…26	10	30 30 30 30 30 30 30 30 30 30	0000000000	Null tag
27	1	03	ETX	End of Text
28	1	2D	-	Packet checksum (BCC)
29	1	0D	CR	Carriage Return

c) for HF devices with only 1 antenna it answers,

c1) if the antenna have identified a transponder:

<SOH>FF<STX>1111223344<ETX><NULL><CR>

Hexadecimal: 01 46 46 02 31 31 31 31 32 32 33 33 34 34 03 00 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…5	2	31 31	11	Tag type = MIFARE 1k (UID=4)
6…14	8	31 31 32 32 33 33 34 34	11223344	Tag code = 11 22 33 44
15	1	03	ETX	End of Text
16	1	00	NULL	Packet checksum (BCC)

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
17	1	0D	CR	Carriage Return

c2) if no transponder is identified by the antenna:

<SOH>FF<STX>0000000000<ETX><NULL><CR>

Hexadecimal: 01 46 46 02 30 30 30 30 30 30 30 30 30 30 03 00 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…14	10	30 30 30 30 30 30 30 30 30 30	0000000000	Null tag
15	1	03	ETX	End of Text
16	1	00	NULL	Packet checksum (BCC)
17	1	0D	CR	Carriage Return

d) for HF devices with 2 antennas, it answers,

d1) if both antennas have identified a transponder:

<SOH>FF<STX>1111223344-1201020304<ETX>)<CR>

Hexadecimal: 01 46 46 02 31 31 31 31 32 32 33 33 34 34 2D 31 32 30 31 30 32 30 33 30 34 03 29 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…5	2	31 31	11	Tag type = MIFARE 1k (UID=4)
6…14	8	31 31 32 32 33 33 34 34	11223344	Tag code = 11 22 33 44
15	1	2D	-	Separator antenna 1 /2
16…17	2	31 32	12	Tag type = MIFARE 4k (UID=4)
18…26	8	30 31 30 32 30 33 30 34	01020304	Tag code = 01 02 03 04
27	1	03	ETX	End of Text
28	1	29	)	Packet checksum (BCC)
29	1	0D	CR	Carriage Return

d2) if only antenna 1 have identified a transponder:

<SOH>FF<STX>1111223344-0000000000<ETX>-<CR>

Hexadecimal: 01 46 46 02 31 31 31 31 32 32 33 33 34 34 2D 30 30 30 30 30 30 30 30 30 30 03 2D 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…5	2	31 31	11	Tag type = MIFARE 1k (UID=4)
6…14	8	31 31 32 32 33 33 34 34	11223344	Tag code = 11 22 33 44

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
15	1	2D	-	Separator antenna 1 /2
16…26	10	30 30 30 30 30 30 30 30 30 30	0000000000	Null tag
27	1	03	ETX	End of Text
28	1	05	ENQ	Packet checksum (BCC)
29	1	0D	CR	Carriage Return

d3) if only antenna 2 have identified a transponder:

<SOH>FF<STX>0000000000-1201020304<ETX><BELL><CR>

Hexadecimal: 01 46 46 02 30 30 30 30 30 30 30 30 30 30 2D 31 32 30 31 30 32 30 33 30 34 03 07 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…14	10	30 30 30 30 30 30 30 30 30 30	0000000000	Null tag
15	1	2D	-	Separator antenna 1 /2
16…17	2	31 32	12	Tag type = MIFARE 4k (UID=4)
18…26	8	30 31 30 32 30 33 30 34	01020304	Tag code = 01 02 03 04
27	1	03	ETX	End of Text
28	1	07	BELL	Packet checksum (BCC)

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
29	1	0D	CR	Carriage Return

d4) if none of the antennas have identified a transponder:

<SOH>FF<STX>0000000000-0000000000<ETX>-<CR>

Hexadecimal: 01 46 46 02 30 30 30 30 30 30 30 30 30 30 2D 30 30 30 30 30 30 30 30 30 30 03 2D 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…14	10	30 30 30 30 30 30 30 30 30 30	0000000000	Null tag
15	1	2D	-	Separator antenna 1 /2
16…26	10	30 30 30 30 30 30 30 30 30 30	0000000000	Null tag
27	1	03	ETX	End of Text
28	1	2D	-	Packet checksum (BCC)
29	1	0D	CR	Carriage Return

e) for UHF devices with 1 antenna, it answers,

e1) if only one transponder was found, transponder info, antenna info, RSSI info and tag read count are not active, and the device is not in 'gate' mode:

<SOH>FF<STX>3000000102030405060708090A0BFF27<ETX><ENQ><CR>

Hexadecimal: 01 46 46 02 33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37 03 05 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…36	32	33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37	30000001 02030405 06070809 0A0BFF27	Tag code = 33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37
37	1	03	ETX	End of Text
38	1	05	ENQ	Packet checksum (BCC)
39	1	0D	CR	Carriage Return

3.3 Queue Data Request

In 'continuous' mode, when the **BLUEBOX** finds a 'new' transponder, it inserts its code in the FIFO queue. This command sends back the first present code in the queue.

The 'master' sends the following command:

<SOH>FF<SYN><ETB><CR>

Hexadecimal: 01 46 46 16 17 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	16	SYN	Synchronouse idle
4	1	17	ETB	Packet checksum (BCC)
5	1	0D	CR	Carriage Return

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

<SOH>FF<NAK><DC4><CR>

Hexadecimal: 01 46 46 15 14 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	15	NAK	Not Acknoledge
4	1	14	DC4	Packet checksum (BCC)
5	1	0D	CR	Carriage Return

Otherwise,

a) for LF devices with only 1 antenna it answers:

<SOH>FF<STX>0011223344<ETX><NULL><CR>

Hexadecimal: 01 46 46 02 30 30 31 31 32 32 33 33 34 34 03 00 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…14	10	30 30 31 31 32 32 33 33 34 34	0011223344	Tag code = 00 11 22 33 44
15	1	03	ETX	End of Text
16	1	00	NULL	Packet checksum (BCC)
17	1	0D	CR	Carriage Return

b) for LF devices with 2 antennas, it answers:

<SOH>FF<STX>001122334401<ETX><STX><CR>

Hexadecimal: 01 46 46 02 30 30 31 31 32 32 33 33 34 34 30 31 03 02 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…14	10	30 30 31 31 32 32 33 33 34 34	0011223344	Tag code = 00 11 22 33 44
15…16	2	30 31	01	Antenna = 1
17	1	03	ETX	End of Text
18	1	02	STX	Packet checksum (BCC)
19	1	0D	CR	Carriage Return

c) for HF devices with only 1 antenna it answers:

<SOH>FF<STX>1111223344<ETX><NULL><CR>

Hexadecimal: 01 46 46 02 31 31 31 31 32 32 33 33 34 34 03 00 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…5	2	31 31	11	Tag type = MIFARE 1k (UID=4)
6…14	8	31 31 32 32 33 33 34 34	11223344	Tag code = 11 22 33 44
15	1	03	ETX	End of Text
16	1	00	NULL	Packet checksum (BCC)
17	1	0D	CR	Carriage Return

d) for HF devices with 2 antennas, it answers:

<SOH>FF<STX>111122334401<ETX><STX><CR>

Hexadecimal: 01 46 46 02 31 31 31 31 32 32 33 33 34 34 30 31 03 02 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…5	2	31 31	11	Tag type = MIFARE 1k (UID=4)
6…14	8	31 31 32 32 33 33 34 34	11223344	Tag code = 11 22 33 44

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
15…16	2	30 31	01	Antenna = 1
17	1	03	ETX	End of Text
18	1	02	STX	Packet checksum (BCC)
19	1	0D	CR	Carriage Return

e) for UHF devices with 1 antenna, it answers,

e1) if transponder info, antenna info and RSSI info are not active, and the device is not in 'gate' mode:

<SOH>FF<STX>3000000102030405060708090A0BFF27<ETX><ENQ><CR>

Hexadecimal: 01 46 46 02 33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37 03 05 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…36	32	33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37	30000001 02030405 06070809 0A0BFF27	Tag code = 33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37
37	1	03	ETX	End of Text
38	1	05	ENQ	Packet checksum (BCC)

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
39	1	0D	CR	Carriage Return

If the queue is empty, the **BLUEBOX** will answer with:

<SOH>FF<STX>0000000000<ETX><NULL><CR>

Hexadecimal: 01 46 46 02 30 30 30 30 30 30 30 30 30 30 03 00 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255
3	1	02	STX	Start of Text
4…14	10	30 30 30 30 30 30 30 30 30 30	0000000000	Null tag
15	1	03	ETX	End of Text
16	1	00	NULL	Packet checksum (BCC)
17	1	0D	CR	Carriage Return

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

<SOH>FF<ACK><BELL><CR>

Hexadecimal: 01 46 46 06 07 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	01	SOH	Start of Header
1…2	2	46 46	FF	Node address = 255

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
3	1	06	ACK	Acknoledge
4	1	07	BELL	Packet checksum (BCC)
5	1	0D	CR	Carriage Return

3.4 'Spontaneous' Message

In 'continuous' mode, if the 'spontaneous' feature is set on (see parameters), the **BLUEBOX** will send the following message on the serial line and on the Ethernet channel (if available) every time that it will find a 'new' tag,

a) for LF devices with only 1 antenna:

<STX>0011223344<ETX><SOH><CR>

Hexadecimal: 02 30 30 31 31 32 32 33 33 34 34 03 01 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	02	STX	Start of Text
1…11	10	30 30 31 31 32 32 33 33 34 34	0011223344	Tag code = 00 11 22 33 44
12	1	03	ETX	End of Text
13	1	01	SOH	Packet checksum (BCC)
14	1	0D	CR	Carriage Return

b) for LF devices with 2 antennas:

<STX>001122334401<ETX><NULL><CR>

Hexadecimal: 02 30 30 31 31 32 32 33 33 34 34 30 31 03 00 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	02	STX	Start of Text
1…11	10	30 30 31 31 32 32 33 33 34 34	0011223344	Tag code = 00 11 22 33 44
11…12	2	30 31	01	Antenna = 1
13	1	03	ETX	End of Text
14	1	00	NULL	Packet checksum (BCC)
15	1	0D	CR	Carriage Return

c) for HF devices with only 1 antenna:

<STX>1111223344<ETX><SOH><CR>

Hexadecimal: 02 31 31 31 31 32 32 33 33 34 34 03 01 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	02	STX	Start of Text
1…2	2	31 31	11	Tag type = MIFARE 1k (UID=4)
3…11	8	31 31 32 32 33 33 34 34	11223344	Tag code = 11 22 33 44
12	1	03	ETX	End of Text
13	1	01	SOH	Packet checksum (BCC)
14	1	0D	CR	Carriage Return

d) for HF devices with 2 antennas:

<STX>111122334401<ETX><NULL><CR>

Hexadecimal: 02 31 31 31 31 32 32 33 33 34 34 30 31 03 00 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	02	STX	Start of Text
1…2	2	31 31	11	Tag type = MIFARE 1k (UID=4)
3…11	8	31 31 32 32 33 33 34 34	11223344	Tag code = 11 22 33 44
11…12	2	30 31	01	Antenna = 1
13	1	03	ETX	End of Text
14	1	00	NULL	Packet checksum (BCC)
15	1	0D	CR	Carriage Return

e) for UHF devices with 1 antenna, it answers,

e1) if transponder info, antenna info and RSSI info are not active, and the device is not in 'gate' mode:

<STX>3000000102030405060708090A0BFF27<ETX><ENQ><CR>

Hexadecimal: 02 33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37 03 05 0D

And in detail:

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
0	1	02	STX	Start of Text
3…35	32	33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37	300000010203040506 0708090A0 BFF27	Tag code = 33 30 30 30 30 30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30 41 30 42 46 46 32 37
36	1	03	ETX	End of Text

Byte#	Tx Bytes	Tx Data (HEX)	Value (ASCII)	Description
37	1	05	ENQ	Packet checksum (BCC)
38	1	0D	CR	Carriage Return

4 Getting Started with C

4.1 Command / Reply Checksum

The code below shows how to calculate the command and reply checksum.

```
#define SOH 0x01
#define EOT 0x04
#define CR 0x0D

unsigned char checksum_bb_cmd(unsigned char * ptBuffer,
                              unsigned long nLength)
{
    unsigned char bChecksum = 0;
    unsigned long nCounter;

    for (nCounter = 0; nCounter < nLength; nCounter++)
        bChecksum ^= ptBuffer[nCounter];

    if ((bChecksum == SOH) || (bChecksum == EOT) || (bChecksum == CR))
        bChecksum++;

    return bChecksum;
}
```

4.2 'Spontaneous' Message Checksum

The code below shows how to calculate the 'spontaneous' message checksum.

```
#define STX 0x02
#define CR 0x0D

unsigned char checksum_bb_spt(unsigned char * ptBuffer,
                              unsigned long nLength)
{
    unsigned char bChecksum = 0;
    unsigned long nCounter;

    for (nCounter = 0; nCounter < nLength; nCounter++)
        bChecksum ^= ptBuffer[nCounter];

    if ((bChecksum == STX) || (bChecksum == CR))
        bChecksum++;

    return bChecksum;
}
```

4.3 Command / Reply Management

The code below shows how to build a packet to send to a **BLUEBOX** and decode the reply received.

Hereinafter some conversion functions.

```
unsigned char bin_to_hex(unsigned char bValue)
{
    bValue &= 0x0F;

    if (bValue < 10)
        return ('0' + bValue);

    return (bValue - 10 + 'A');
}

void byte_to_dchar(unsigned char *pdc,
                   unsigned char bValue)
{
    pdc[0] = bin_to_hex((bValue & 0xF0) >> 4);
    pdc[1] = bin_to_hex(bValue & 0x0F);
}

void multi_b2dc(unsigned char *pdc,
                unsigned char *ptb,
                unsigned long nLength)
{
    unsigned long nCount;

    for (nCount = 0; nCount < nLength; nCount++, pdc += 2)
        byte_to_dchar(pdc, ptb[nCount]);
}

unsigned char hex_to_bin(unsigned char bValue)
{
    if ((bValue >= '0') && (bValue <= '9'))
        return (bValue - '0');

    if ((bValue >= 'A') && (bValue <= 'F'))
        return (bValue - 'A' + 10);

    if ((bValue >= 'a') && (bValue <= 'f'))
        return (bValue - 'a' + 10);

    return (0);
}

unsigned char dchar_to_byte(unsigned char *pdc)
{
    return ((hex_to_bin(pdc[0]) << 4) | hex_to_bin(pdc[1]));

}

void multi_dc2b(unsigned char *ptb,
```

```
                unsigned char *pdc,
                unsigned long nLength)
{
    unsigned long nCount;

    for (nCount = 0; nCount < nLength; nCount++, pdc += 2)
        ptb[w] = dchar_to_byte(pdc);

}
```

Hereinafter the code to build the command packet

```
#define SOH 0x01
#define STX 0x02
#define EOT 0x04
#define CR 0x0D

typedef enum
{
    BLUEBOX_CMD_GET_FIRMWARE = 0x00,
    BLUEBOX_CMD_EMPTY_COMMAND

} BLUEBOX_COMMANDS_E;

typedef struct
{
    char * szCommand;
    char * szDescription;

} BLUEBOX_CMD_INFO_T, *BLUEBOX_PCMD_INFO_T;

const BLUEBOX_CMD_INFO_T tCommands[]=
{
    {"34", "Get firmware"                       },
    {"00", "Empy command (ENQ, SYN, ACK}"       }
};

unsigned char g_bMessage[1024];

void pack_command(BLUEBOX_COMMANDS_E eCommand,
                  unsigned char * pSendData,
                  unsigned long nSendData,
                  unsigned char bRawCommand)
{
    unsigned long nSend = 0;
    unsigned char *pSend = &g_bMessage[0];

    if (pReplyData != NULL)
    {
        // Pack the command to send.
        if (bRawCommand != 0)
        {
            // Only 1 char to send in raw mode!
            nSend = (unsigned long)sprintf((char*)p_send,
                                           "%c%02X%c",
                                           SOH,
                                           0xFF,
```

```
                                        bRawCommand);
        }
        else
        {
            nSend = (unsigned long)sprintf((char*)pSend,
                                        "%c%02X%c%s",
                                        SOH, 0xFF,
                                        STX,
                                        tCommands[eCommand].szCommand);

            multi_b2dc(pSend + nSend, pSendData, nSendData);
            nSend += nSendData * 2;

            nSend += (unsigned long)sprintf((char*)(pSend + nSend),
                                         "%c",
                                         ETX);
        }

        // Checksum and carriage return.
        pSend[nSend++] = checksum_bb_cmd(pSend, nSend);
        pSend[nSend++] = CR;

        // ...add code to send data here.

    }
}
```

For example to send the 'FW Version Reading' command

```
pack_command(BLUEBOX_CMD_GET_FIRMWARE, NULL, 0, 0x00);
```

and to send a 'Data Request' command

```
pack_command(BLUEBOX_CMD_EMPTY_COMMAND, NULL, 0, 0x05);
```

Hereinafter the code to decode the reply received.

```
#define SOH 0x01
#define STX 0x02
#define EOT 0x04
#define ACK 0x06
#define CR 0x0D
#define NAK 0x15

#define __TRUE 1
#define __FALSE 0

bool check_command(unsigned char bAddress,
                   unsigned char * pData,
                   unsigned long nData)
{
    bool fRetval = __FALSE;

    if ((pData != NULL) && (nData >= 0x06))
    {
        unsigned long nLength;
        unsigned char bTemp[3];
```

```
        nLength = (unsigned long)sprintf((char*)bTemp,
                                         "%c%02X",
                                         SOH,
                                         bAddress);

        if (memncmp(pData, bTemp, nLength) == 0)
        {
            if (pData[nData - 0x02] == checksum_bb_cmd(pData, nData - 0x02))
            {
                fRetval = __TRUE;
            }
        }
    }

    return fRetval;
}

bool process_reply(unsigned char * pData,
                   unsigned long nData,
                   unsigned char * pDataReply,
                   unsigned long * nDataReply,
                   bool fIsRawCommand)
{
    bool fRetval = __FALSE;

    // Check if command it's correct.
    if (check_command(0xFF, pData, nData))
    {
        switch (pData[0x03])
        {
            case STX:
            {
                fRetval = __TRUE;
                // Convert to hex.
                multi_dc2b(pDataReply,
                           pData + (fIsRawCommand ? 4 : 6),
                           nData - (fIsRawCommand ? 7 : 9));
                *nDataReply = (nData - (fIsRawCommand ? 7 : 9)) >> 1;
            }
            break;

            case ACK:
            {
                fRetval = __TRUE;
            }
            break;

            case NAK:
            {
                fRetval = __FALSE;
            }
            break;
        }
    }

    return fRetval;
}
```

For example to decode a 'FW Version Reading' reply

```
// ...add code to get the reply (from 0x01 to 0x0D chars) with buffer overflow
check algorithm here. */

char szVersion[65];

unsigned char pDataReply[1024];
unsigned long nDataReply = 0;

if (process_reply(pData, nData, pDataReply, &nDataReply, __FALSE))
{
    memset((unsigned char*)szVersion, 0, sizeof(szVersion));
    memncpy((unsigned char*)szVersion, pDataReply, nDataReply);
    // In szVersion the firmware version...
}
```

5 Document Revision History

Date	Revision	Description
30/08/16	1.00	Initial release.
01/09/16	1.01	Added 5221L, 5222L, 5231L, 5232L, 5241L and 5242L readers support to this manual. Added 5221U-S, 5222U-S, 5237U-S, 5238U-S, 5231U and 5232U readers support to this manual. Added the 'Reading Test Activation/Deactivation' command description (section 2.1). Added LF readers support to 'Data Request' and 'Queue Data Request' commands (sections 2.6 and 2.7). Added LF transponders management commands (sections 2.8 to 2.25). Added LF readers support to 'Spontaneous' message description (section 2.50). Added LF readers support to commands examples (sections 3.2, 3.3 and 3.4). Added UHF readers support to 'Data Request' and 'Queue Data Request' commands (sections 2.6 and 2.7). Added UHF transponders management commands (sections 2.44 to 2.49). Added UHF readers support to 'Spontaneous' message description (section 2.50). Added UHF readers support to commands examples (sections 3.2, 3.3 and 3.4). Added LF supported transponders as appendix A. Added UHF supported transponders as appendix C. Added LF transponders nibble coding descriptions as appendix D. Updated the supported commands table in appendix E.
19/09/16	1.02	Added 5325U, 5325U-RTC, 5335U, 5335U-RTC, 5345U, 5345U-RTC, 5326U, 5326U-RTC, 5336U, 5336U-RTC, 5346U and 5346U-RTC readers support to this manual. Added 5426U, 5426U-RTC, 5426U-G, 5426U-RTC-G, 5526U, 5526U-RTC, 5526U-G and 5526U-RTC-G readers support to this manual. Added 'Temperature Reading' reply format for readers different from 52xxL, 52xxH and 52xxU. Updated the 'Reading Test Activation/Deactivation' command. Added the 'Power Test' command.

Date	Revision	Description
		Updated the supported commands table in appendix E.
26/09/16	1.03	Added the 'Antennas Auto-Tuning' command. Added 1021U, 1031U, 1021U-S, 1031U-S, 1041U, 1051U, 1061U, 1071U, 1061U-S, 1071U-S, 5224U and 5225U readers support to this manual.
30/09/16	1.04	Updated the 5221U-S, 5222U-S, 5237U-S, 5238U-S, 5231U and 5232U firmware versions object of this manual. Added 'RF Power Test' command to 5221U-S, 5222U-S, 5237U-S, 5238U-S, 5231U and 5232U readers. Added 'Antennas Auto-Tuning' command to 5031U and 5032U readers.
04/10/16	1.05	Added 3122L, 3122L-I, 3122H, 3122H-I, 3122U, 3122U-I, 5121L, 5131L, 5121H, 5131H, 5121U and 5131U readers support to this manual.
25/10/16	1.06	Added 7027U reader support to this manual. Added 'Number of Records Reading', 'Record Database Reset', 'Current Record Request' and 'Current Record Unqueue' commands.
23/12/16	1.07	Added 1021N and 1031N readers support to this manual. Added 'ISO 14443A-4 Transponder RATS Command' and 'ISO 14443A-4 Trasnponder Generic Command' commands. Deleted supported tags appendixes.
12/07/17	1.08	Corrected errors in table with supported readers object of this manual. Corrected errors in table with firmware versions object of this manual. Corrected all the read and write commands titles. Added the variable size (max 240 bytes) null terminated string parameters management in 'Read Configuration Parameters' and 'Write Configuration Parameters' commands. Fixed the 'Buffer Data Request', 'Queue Data Request' and 'Spontaneous Message' reply data in case of UHF device.
10/10/17	1.09	Added 1021L, 1031L and 1041L readers support to this manual. Added 1021H, 1031H, 1041H and 1051H readers support to this manual.

Date	Revision	Description
		Added 3122N and 3122N-I readers support to this manual. Added 3221L, 3222L, 3221N and 3222N readers support to this manual. Updated reader's firmware versions object of this manual. Added 'Write Data to an HITAG 2 Transponder', 'Read ID of an HITAG 2 Transponder', 'Read Page of an HITAG 2 Transponder' and 'Write Page of an HITAG 2 Transponder' commands.
07/11/17	1.10	Updated reader's firmware versions object of this manual. Added the MIFARE DESFire Transponder 'Generic Command' with subcommands.
15/01/18	1.11	Updated reader's firmware versions object of this manual. Added the 5325U-FCC, 5325U-RTC-FCC, 5326U-FCC and 5326U-RTC-FCC readers support to this manual. Added the 'RF Sensitivity' Test, Read Reflected Power and Read RSSI Power commands. Added the reading timestamp information to 'Buffer Data Request', 'Queue Data Request' and 'Spontaneous Message'. Added the 'QT Read' and 'QT Write' commands of an Impinj Monza 4QT transponder. Added the 'Read Sensor Code' and 'Read On-Chip RSSI' commands of an RFMicron Magnus S2 and S3 transponders. Added the 'Read Temperature Code' of an RFMicron Magnus S3. Splitted the supported commands table per device family.
30/01/18	1.12	Updated reader's firmware versions object of this manual. Added the 3122H-E, 3122H-K, 3122N-M, 3122U-K and 7062L readers support to this manual.
08/02/18	1.13	Updated reader's firmware versions object of this manual. Corrected the 'Write Data to an HITAG S Transponder' command reply. Canceled the 'Write Data to an HITAG 2 Transponder' command. Corrected the antenna nr 2 support in 'Read ID Code of a HITAG 2 Transponder', 'Read a Page of a HITAG 2 Transponder' and 'Write a Page of a HITAG 2 Transponder' commands.

Date	Revision	Description
		Added the PicoPass transponder 'Inventory' command. Updated the supported commands tables.
29/05/18	1.14	Updated reader's firmware versions object of this manual. Added 'Re-Read an Unqueued Record', 'Start Continuous Read Records' and 'Stop Continuous Read Records' commands.
08/06/18	1.15	Updated reader's firmware versions object of this manual. Fixed the 'Read Temperature' and 'Read Date/Time' commands. Updated the supported commands tables.
05/07/18	1.16	Added the 'Firmware Upgrade' command and procedure. Updated the supported commands tables. Added the XMODEM protocol overview.
01/08/18	1.17	Updated reader's firmware versions object of this manual. Added the RSSI Q and I channel info in 'Buffer Data Request', 'Queue Data Request' and 'Spontaneous Message'.
22/10/18	1.18	Updated reader's firmware versions object of this manual. Added the 5721U reader support to this manual. Removed custom readers support from this manual. Added the tag read count info in 'Buffer Data Request'. Updated the 'Buffer Data Request', 'Queue Data Request' and 'Spontaneous Message' examples. Minor corrections.
26/10/18	1.19	Added the 3223L and 3223N readers support to this manual.
14/12/18	1.20	Updated reader's firmware versions object of this manual. Added the 'Sleep Mode' command. Updated the supported commands tables.
11/01/19	1.21	Updated the company name/logo and BLUEBOX logo. Updated reader's firmware versions object of this manual. Added the max RSSI Q and I channel info in 'Buffer Data Request'.
27/05/19	1.22	Added the 5327U[-FCC], 5337U, 5347U, 5328U[-RTC][-FCC], 5338U[-RTC], 5348U[-RTC], 5427U[-G], 5428U[-RTC][-G], 5527U[-G], 5528U[-RTC][-G] and 5346U-BL

Date	Revision	Description
		readers support to this manual. Replaced the 'Read MAC Address' command with 'Read Ethernet MAC Address' command. Added the 'Read Bluetooth MAC Address' command. Updated the supported commands tables.
02/09/19	1.23	Updated reader's firmware versions object of this manual. Replaced 'Write General Parameters' command with 'Write ROM General Parameters' command. Replaced 'Write Configuration Parameters' command with 'Write ROM Configuration Parameters' command. Added the 'Write RAM Configuration Parameters' command. Replaced the 'Set Default Parameters' command with 'Write ROM Default Parameters' command. Replaced the 'Read General Parameters' command with 'Read RAM General Parameters' command. Replaced the 'Read Configuration Parameters' command with 'Read RAM Configuration Parameters' command. Added the 'Read ROM Configuration Parameters' command. Updated the supported commands tables.
14/11/19	1.24	Updated reader's firmware versions object of this manual. Removed custom readers from this manual. Replaced ISO 18000-6C with ISO 18000-63. They are the same standard, 18000-6C became 18000-63 in 2012 due to ISO naming rules that do not allow letters in standards names. Updated the supported commands tables.
04/05/20	1.25	Added the 1021U-FCC, 1021U-BRA, 1021U-S-FCC, 1021U-S-BRA, 1031U-FCC, 1031U-BRA, 1031U-S-FCC, 1031U-S-BRA, 1041U-FCC, 1041U-BRA, 1051U-FCC, 1051U-BRA, 1061U-FCC, 1061U-BRA, 1061U-S-FCC, 1061U-S-BRA, 1071U-FCC, 1071U-BRA, 1071U-S-FCC, 1071U-S-BRA readers support to this manual. Updated the reader's description object of this manual. Updated the supported commands tables. Format changes and document fixes in all sections.
07/06/21	1.26	Added the 1021L v4 and 1031L v4 readers support to this manual. Added the 1021H v2 and 1021H v2 readers support to

Date	Revision	Description
		this manual. Added 3122L v2+4 and 3122H v2+2 readers support to this manual. Added 3221L v4, 3222L v4 and 3223L v1+4 readers support to this manual. Updated the reader's description object of this manual. Updated the supported commands tables.
20/12/21	1.27	Updated the reader's description object of this manual.
24/01/22	1.28	Updated the reader's description object of this manual. Updated the supported commands tables.

A Supported Commands Table

A.1 OEM Devices

	1021L v3, 1031L v3, 1041L v3	1021L v4, 1031L v4	1021H v1, 1031H v1, 1041H v1	1051H v1	1021H v2, 1031H v2	1021N v1, 1031N v1	1021U, 1031U, 1061U, 1071U	1041U, 1051U
Device Reset	✓	✓	✓	✓	✓	✓	✓	✓
Read Device Serial Number		✓			✓	✓	✓	✓
Read Ethernt MAC Address								
Read Bluetooth MAC Address								
Read Firmware Version	✓	✓	✓	✓	✓	✓	✓	✓
Firmware Upgrade	✓	✓	✓	✓	✓	✓	✓	✓
Read Temperature								
Read Date/Time								
Write Date/Time								
Write ROM General Parameters	✓	✓	✓	✓	✓	✓	✓	✓
Write RAM Configuration Parameters							✓	✓
Write ROM Configuration Parameters							✓	✓
Write ROM Default Parameters	✓	✓	✓	✓	✓	✓	✓	✓
Read RAM General Parameters	✓	✓	✓	✓	✓	✓	✓	✓
Read RAM Configuration Parameters							✓	✓
Read ROM Configuration Parameters							✓	✓
Sleep Mode							✓	✓
'RF Reading' Test							✓	✓
'RF Power' Test							✓	✓
'RF Sensitivity' Test							✓	✓
Read Reflected Power							✓	✓
Read RSSI Power							✓	✓
Digital Output Activation	✓	✓	✓	✓	✓	✓	✓	✓
Read Device Status	✓	✓	✓	✓	✓	✓	✓	✓
RF Deactivation	✓	✓	✓	✓	✓	✓	✓	✓
RF Activation	✓	✓	✓	✓	✓	✓	✓	✓
Antennas Auto-Tuning								✓
Buffer Data Request	✓	✓	✓	✓	✓	✓	✓	✓
Queue Data Request	✓	✓	✓	✓	✓	✓	✓	✓
Read Number of Records								
Reset Record Database								

	1021L v3, 1031L v3, 1041L v3	1021L v4, 1031L v4	1021H v1, 1031H v1, 1041H v1	1051H v1	1021H v2, 1031H v2	1021N v1, 1031N v1	1021U, 1031U, 1061U, 1071U	1041U, 1051U
Request Current Record								
Unqueue Current Record								
Re-Read an Unqueued Record								
Start Continuous Read Records								
Stop Continuous Read Records								
Write Data to an EM4305 Transponder	✓	✓						
Read ID Code of an EM4305 Transponder	✓	✓						
Write Data to a T5557 Transponder	✓	✓						
Read ID Code of a T5557 Transponder	✓	✓						
Write Data to a Q5 Transponder	✓	✓						
Read ID Code of a Q5 Transponder	✓	✓						
Write Data to an HITAG S Transponder	✓	✓						
Read ID Code of an HITAG 1 / HITAG S Transponder	✓	✓						
Read a Page of an HITAG 1 / HITAG S Transponder	✓	✓						
Write a Page of an HITAG 1 / HITAG S Transponder	✓	✓						
Read ID Code of an HITAG 2 Transponder	✓	✓						
Read a Page of an HITAG 2 Transponder	✓	✓						
Write a Page of an HITAG 2 Transponder	✓	✓						
'Reset' Commad for a TITAN Transponder	✓	✓						
'Login' Command for a TITAN Transponder	✓	✓						
'Write Password' Command for a TITAN Transponder	✓	✓						
'Standard Read' Command for a TITAN Transponder	✓	✓						
'Selective Read' Command for a TITAN Transponder	✓	✓						
'Write Word' Command for a TITAN Transponder	✓	✓						
'Write Several Words' Command for a TITAN Transponder	✓	✓						
'Read After Write Word' Command for a TITAN Transponder	✓	✓						
ISO 15693 Transponders 'Inventory' Command			✓		✓	✓		
Read a Data Block of an ISO 15693 Transponder			✓		✓	✓		
Write a Data Block of an ISO 15693 Transponder			✓		✓	✓		
Lock a Data Block of an ISO 15693 Transponder			✓		✓	✓		
ISO 15693 Transponder 'Get System Info' Command			✓		✓	✓		
ISO 15693 Transponder 'Generic' Command			✓		✓	✓		
ISO 14443A Transponders 'Inventory' Command			✓	✓	✓	✓		
Read a Data Block of a MIFARE Mini/1k/4k (UID 4) Transponder			✓	✓	✓	✓		
Write a Data Block of a MIFARE Mini/1k/4k (UID 4) Transponder			✓	✓	✓	✓		
Read a Data Block of a MIFARE 1k/4k (UID 7) Transponder			✓	✓	✓	✓		

	1021L v3, 1031L v3, 1041L v3	1021L v4, 1031L v4	1021H v1, 1031H v1, 1041H v1	1051H v1	1021H v2, 1031H v2	1021N v1, 1031N v1	1021U, 1031U, 1061U, 1071U	1041U, 1051U
Write a Data Block of a MIFARE 1k/4k (UID 7) Transponder			✓	✓	✓	✓		
Read a Data Page of a MIFARE Ultralight Transponder			✓	✓	✓	✓		
Write a Data Page of a MIFARE Ultralight Transponder			✓	✓	✓	✓		
Read a Data Page of a NTAG213/215/216 Transponder			✓	✓	✓	✓		
Write a Data Page of a NTAG213/215/216 Transponder			✓	✓	✓	✓		
ISO 14443A-4 Transponder 'RATS' Command			✓	✓	✓	✓		
ISO 14443°-4 Transponder 'Generic' Command			✓	✓	✓	✓		
MIFARE DESFire Transponder 'Generic' Command					✓	✓		
ISO 14443B Transponders 'Inventory' Command			✓					
Read a Data Block of a SR 176 Transponder			✓					
Write a Data Block of a SR176 Transponder			✓					
PicoPass Transponders 'Inventory' Command			✓		✓			
ISO 18000-63 Transponder 'Inventory' Command							✓	✓
Program EPC of an ISO 18000-63 Transponder							✓	✓
Read Data of an ISO 18000-63 Transponder							✓	✓
Write Data of an ISO 18000-63 Transponder							✓	✓
Lock Data of an ISO 18000-63 Transponder							✓	✓
'Kill' Command of an ISO 18000-63 Transponder							✓	✓
'QT Read' Command of an Impinj Monza 4QT Transponder							✓	✓
'QT Write' Command of an Impinj Monza 4QT Transponder							✓	✓
'Read Sensor Code' Command of an RFMicron Magnus S2 / S3 Transponder							✓	✓
'Read On-Chip RSSI' Command of an RFMicron Magnus S2 / S3 Transponder							✓	✓
'Read Temperature Code' Command of an RFMicron Magnus S2 / S3 Transponder							✓	✓

A.2 Desktop Devices

	3122L v2+3	3122L v2+4	3122H v2+1	3122H v2+2	3122N v2+1	3122U	3221L v3, 3222L v3	3221L v4, 3122L v4	3221N v1, 3222N v1	3221N v2, 3222N v2	3223L v1+3	3223L v1+4	3223N v1+1	3223N v1+2
Device Reset	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read Device Serial Number	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
Read Ethernt MAC Address														
Read Bluetooth MAC Address														
Read Firmware Version	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Firmware Upgrade	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read Temperature														
Read Date/Time														
Write Date/Time														
Write ROM General Parameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Write RAM Configuration Parameters	✓	✓	✓	✓	✓	✓					✓	✓	✓	✓
Write ROM Configuration Parameters	✓	✓	✓	✓	✓	✓					✓	✓	✓	✓
Write ROM Default Parameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read RAM General Parameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read RAM Configuration Parameters	✓	✓	✓	✓	✓	✓					✓	✓	✓	✓
Read ROM Configuration Parameters	✓	✓	✓	✓	✓	✓					✓	✓	✓	✓
Sleep Mode														
'RF Reading' Test														
'RF Power' Test						✓								
'RF Sensitivity' Test						✓								
Read Reflected Power						✓								
Read RSSI Power						✓								
Digital Output Activation														
Read Device Status	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RF Deactivation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RF Activation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Antennas Auto-Tuning														
Buffer Data Request	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Queue Data Request	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read Number of Records														
Reset Record Database														
Request Current Record														

	3122L v2+3	3122L v2+4	3122H v2+1	3122H v2+2	3122N v2+1	3122U	3221L v3, 3222L v3	3221L v4, 3122L v4	3221N v1, 3222N v1	3221N v2, 3222N v2	3223L v1+3	3223L v1+4	3223N v1+1	3223N v1+2
Unqueue Current Record														
Re-Read an Unqueued Record														
Start Continuous Read Records														
Stop Continuous Read Records														
Write Data to an EM4305 Transponder	✓	✓					✓	✓			✓	✓		
Read ID Code of an EM4305 Transponder	✓	✓					✓	✓			✓	✓		
Write Data to a T5557 Transponder	✓	✓					✓	✓			✓	✓		
Read ID Code of a T5557 Transponder	✓	✓					✓	✓			✓	✓		
Write Data to a Q5 Transponder	✓	✓					✓	✓			✓	✓		
Read ID Code of a Q5 Transponder	✓	✓					✓	✓			✓	✓		
Write Data to an HITAG S Transponder	✓	✓					✓	✓			✓	✓		
Read ID Code of an HITAG 1 / HITAG S Transponder	✓	✓					✓	✓			✓	✓		
Read a Page of an HITAG 1 / HITAG S Transponder	✓	✓					✓	✓			✓	✓		
Write a Page of an HITAG 1 / HITAG S Transponder	✓	✓					✓	✓			✓	✓		
Read ID Code of an HITAG 2 Transponder	✓	✓					✓	✓			✓	✓		
Read a Page of an HITAG 2 Transponder	✓	✓					✓	✓			✓	✓		
Write a Page of an HITAG 2 Transponder	✓	✓					✓	✓			✓	✓		
'Reset' Commad for a TITAN Transponder	✓	✓					✓	✓			✓	✓		
'Login' Command for a TITAN Transponder	✓	✓					✓	✓			✓	✓		
'Write Password' Command for a TITAN Transponder	✓	✓					✓	✓			✓	✓		
'Standard Read' Command for a TITAN Transponder	✓	✓					✓	✓			✓	✓		
'Selective Read' Command for a TITAN Transponder	✓	✓					✓	✓			✓	✓		
'Write Word' Command for a TITAN Transponder	✓	✓					✓	✓			✓	✓		
'Write Several Words' Command for a TITAN Transponder	✓	✓					✓	✓			✓	✓		
'Read After Write Word' Command for a TITAN Transponder	✓	✓					✓	✓			✓	✓		
ISO 15693 Transponders 'Inventory' Command			✓	✓	✓				✓	✓			✓	✓
Read a Data Block of an ISO 15693 Transponder			✓	✓	✓				✓	✓			✓	✓
Write a Data Block of an ISO 15693 Transponder			✓	✓	✓				✓	✓			✓	✓
Lock a Data Block of an ISO 15693 Transponder			✓	✓	✓				✓	✓			✓	✓
ISO 15693 Transponder 'Get System Info' Command			✓	✓	✓				✓	✓			✓	✓
ISO 15693 Transponder 'Generic' Command			✓	✓	✓				✓	✓			✓	✓
ISO 14443A Transponders 'Inventory' Command			✓	✓	✓				✓	✓			✓	✓
Read a Data Block of a MIFARE Mini/1k/4k (UID 4) Transponder			✓	✓	✓				✓	✓			✓	✓
Write a Data Block of a MIFARE Mini/1k/4k (UID 4) Transponder			✓	✓	✓				✓	✓			✓	✓
Read a Data Block of a MIFARE 1k/4k (UID 7) Transponder			✓	✓	✓				✓	✓			✓	✓
Write a Data Block of a MIFARE 1k/4k (UID 7) Transponder			✓	✓	✓				✓	✓			✓	✓

	3122L v2+3	3122L v2+4	3122H v2+1	3122H v2+2	3122N v2+1	3122U	3221L v3, 3222L v3	3221L v4, 3122L v4	3221N v1, 3222N v1	3221N v2, 3222N v2	3223L v1+3	3223L v1+4	3223N v1+1	3223N v1+2
Read a Data Page of a MIFARE Ultralight Transponder			✓	✓	✓				✓	✓			✓	✓
Write a Data Page of a MIFARE Ultralight Transponder			✓	✓	✓				✓	✓			✓	✓
Read a Data Page of a NTAG213/215/216 Transponder			✓	✓	✓				✓	✓			✓	✓
Write a Data Page of a NTAG213/215/216 Transponder			✓	✓	✓				✓	✓			✓	✓
ISO 14443A-4 Transponder 'RATS' Command			✓	✓	✓				✓	✓			✓	✓
ISO 14443A-4 Transponder 'Generic' Command			✓	✓	✓				✓	✓			✓	✓
MIFARE DESFire Transponder 'Generic' Command				✓	✓				✓	✓			✓	✓
ISO 14443B Transponders 'Inventory' Command			✓											
Read a Data Block of a SR 176 Transponder			✓											
Write a Data Block of a SR176 Transponder			✓											
PicoPass Transponders 'Inventory' Command			✓	✓						✓				✓
ISO 18000-63 Transponder 'Inventory' Command						✓								
Program EPC of an ISO 18000-63 Transponder						✓								
Read Data of an ISO 18000-63 Transponder						✓								
Write Data of an ISO 18000-63 Transponder						✓								
Lock Data of an ISO 18000-63 Transponder						✓								
'Kill' Command of an ISO 18000-63 Transponder						✓								
'QT Read' Command of an Impinj Monza 4QT Transponder						✓								
'QT Write' Command of an Impinj Monza 4QT Transponder						✓								
'Read Sensor Code' Command of an RFMicron Magnus S2 / S3 Transponder						✓								
'Read On-Chip RSSI' Command of an RFMicron Magnus S2 / S3 Transponder						✓								
'Read Temperature Code' Command of an RFMicron Magnus S2 / S3 Transponder						✓								

A.3 Industrial Devices

	5121L, 5131L	5121H, 5131H	5121U, 5131U	5221L, 5222L, 5231L, 5232L, 5241L, 5242L	5221H, 5222H, 5231H, 5232H, 5241H, 5242H	5221U-S, 5222U-S, 5237U-S, 5238U-S	5231U, 5232U	5224U, 5225U	532xU, 533xU, 534xU	532xU-RTC, 533xU-RTC, 534xU-RTC	5346U-BL	542xU, 552xU	542xU-RTC, 552xU-RTC	5721U
Device Reset	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read Device Serial Number	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read Ethernt MAC Address				✓	✓	✓	✓		✓	✓	✓	✓	✓	
Read Bluetooth MAC Address											✓			
Read Firmware Version	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Firmware Upgrade	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read Temperature				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Read Date/Time				✓	✓	✓	✓			✓	✓		✓	
Write Date/Time				✓	✓	✓	✓			✓	✓		✓	
Write ROM General Parameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Write RAM Configuration Parameters								✓	✓	✓	✓	✓	✓	✓
Write ROM Configuration Parameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Write ROM Default Parameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read RAM General Parameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read RAM Configuration Parameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read ROM Configuration Parameters								✓	✓	✓	✓	✓	✓	✓
Sleep Mode														
'RF Reading' Test			✓			✓	✓		✓	✓	✓	✓	✓	✓
'RF Power' Test			✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
'RF Sensitivity' Test			✓					✓	✓	✓	✓	✓	✓	✓
Read Reflected Power			✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
Read RSSI Power			✓					✓	✓	✓	✓	✓	✓	✓
Digital Output Activation	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	
Read Device Status	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RF Deactivation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RF Activation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Antennas Auto-Tuning							✓		✓	✓	✓	✓	✓	
Buffer Data Request	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Queue Data Request	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read Number of Records									✓	✓	✓	✓	✓	
Reset Record Database									✓	✓	✓	✓	✓	
Request Current Record									✓	✓	✓	✓	✓	

	5121L, 5131L	5121H, 5131H	5121U, 5131U	5221L, 5222L, 5231L, 5232L, 5241L, 5242L	5221H, 5222H, 5231H, 5232H, 5241H, 5242H	5221U-S, 5222U-S, 5237U-S, 5238U-S	5231U, 5232U	5224U, 5225U	532xU, 533xU, 534xU	532xU-RTC, 533xU-RTC, 534xU-RTC	5346U-BL	542xU, 552xU	542xU-RTC, 552xU-RTC	5721U
Unqueue Current Record									✓	✓	✓	✓	✓	
Re-Read an Unqueued Record									✓	✓	✓	✓	✓	
Start Continuous Read Records									✓	✓	✓	✓	✓	
Stop Continuous Read Records									✓	✓	✓	✓	✓	
Write Data to an EM4305 Transponder	✓			✓										
Read ID Code of an EM4305 Transponder	✓			✓										
Write Data to a T5557 Transponder	✓			✓										
Read ID Code of a T5557 Transponder	✓			✓										
Write Data to a Q5 Transponder	✓			✓										
Read ID Code of a Q5 Transponder	✓			✓										
Write Data to an HITAG S Transponder	✓			✓										
Read ID Code of an HITAG 1 / HITAG S Transponder	✓			✓										
Read a Page of an HITAG 1 / HITAG S Transponder	✓			✓										
Write a Page of an HITAG 1 / HITAG S Transponder	✓			✓										
Read ID Code of an HITAG 2 Transponder	✓			✓										
Read a Page of an HITAG 2 Transponder	✓			✓										
Write a Page of an HITAG 2 Transponder	✓			✓										
'Reset' Commad for a TITAN Transponder	✓			✓										
'Login' Command for a TITAN Transponder	✓			✓										
'Write Password' Command for a TITAN Transponder	✓			✓										
'Standard Read' Command for a TITAN Transponder	✓			✓										
'Selective Read' Command for a TITAN Transponder	✓			✓										
'Write Word' Command for a TITAN Transponder	✓			✓										
'Write Several Words' Command for a TITAN Transponder	✓			✓										
'Read After Write Word' Command for a TITAN Transponder	✓			✓										
ISO 15693 Transponders 'Inventory' Command		✓			✓									
Read a Data Block of an ISO 15693 Transponder		✓			✓									
Write a Data Block of an ISO 15693 Transponder		✓			✓									
Lock a Data Block of an ISO 15693 Transponder		✓			✓									
ISO 15693 Transponder 'Get System Info' Command		✓			✓									
ISO 15693 Transponder 'Generic' Command		✓			✓									
ISO 14443A Transponders 'Inventory' Command		✓			✓									
Read a Data Block of a MIFARE Mini/1k/4k (UID 4) Transponder		✓			✓									
Write a Data Block of a MIFARE Mini/1k/4k (UID 4) Transponder		✓			✓									
Read a Data Block of a MIFARE 1k/4k (UID 7) Transponder		✓			✓									
Write a Data Block of a MIFARE 1k/4k (UID 7) Transponder		✓			✓									

	5121L, 5131L	5121H, 5131H	5121U, 5131U	5221L, 5222L, 5231L, 5232L, 5241L, 5242L	5221H, 5222H, 5231H, 5232H, 5241H, 5242H	5221U-S, 5222U-S, 5237U-S, 5238U-S	5231U, 5232U	5224U, 5225U	532xU, 533xU, 534xU	532xU-RTC, 533xU-RTC, 534xU-RTC	5346U-BL	542xU, 552xU	542xU-RTC, 552xU-RTC	5721U
Read a Data Page of a MIFARE Ultralight Transponder		✓			✓									
Write a Data Page of a MIFARE Ultralight Transponder		✓			✓									
Read a Data Page of a NTAG213/215/216 Transponder		✓			✓									
Write a Data Page of a NTAG213/215/216 Transponder		✓			✓									
ISO 14443A-4 Transponder 'RATS' Command		✓			✓									
ISO 14443A-4 Transponder 'Generic' Command		✓			✓									
MIFARE DESFire Transponder 'Generic' Command														
ISO 14443B Transponders 'Inventory' Command		✓			✓									
Read a Data Block of a SR 176 Transponder		✓			✓									
Write a Data Block of a SR176 Transponder		✓			✓									
PicoPass Transponders 'Inventory' Command					✓									
ISO 18000-63 Transponder 'Inventory' Command			✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
Program EPC of an ISO 18000-63 Transponder			✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
Read Data of an ISO 18000-63 Transponder			✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
Write Data of an ISO 18000-63 Transponder			✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
Lock Data of an ISO 18000-63 Transponder			✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
'Kill' Command of an ISO 18000-63 Transponder			✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
'QT Read' Command of an Impinj Monza 4QT Transponder			✓					✓	✓	✓	✓	✓	✓	✓
'QT Write' Command of an Impinj Monza 4QT Transponder			✓					✓	✓	✓	✓	✓	✓	✓
'Read Sensor Code' Command of an RFMicron Magnus S2 / S3 Transponder			✓					✓	✓	✓	✓	✓	✓	✓
'Read On-Chip RSSI' Command of an RFMicron Magnus S2 / S3 Transponder			✓					✓	✓	✓	✓	✓	✓	✓
'Read Temperature Code' Command of an RFMicron Magnus S2 / S3 Transponder			✓					✓	✓	✓	✓	✓	✓	✓

B XMODEM Protocol Overview

1/1/82 by Ward Christensen.

I will maintain a master copy of this. Please pass on changes or suggestions via CBBS/Chicago at (312) 545-8086, or by voice at (312) 849-6279.

NOTE: This does not include things which I am not familiar with, such as the CRC option implemented by John Mahr.

Last Rev: (none)

At the request of Rick Malinak on behalf of the guys at Standard Oil with IBM P.C.s, as well as several previous requests, I finally decided to put my modem protocol into writing. It had been previously formally published only in the AMRAD newsletter.

-------- Table of Contents

-------- 1. DEFINITIONS.

<soh> 01H
<eot> 04H
<ack> 06H
<nak> 15H
<can> 18H

-------- 2. TRANSMISSION MEDIUM LEVEL PROTOCOL

```
     Asynchronous, 8 data bits, no parity, one stop bit.

    The protocol imposes no restrictions on the contents of the
data being transmitted.  No control characters are looked for
in the 128-byte data messages.  Absolutely any kind of data may
be sent - binary, ASCII, etc.  The protocol has not formally
been adopted to a 7-bit environment for the transmission of
ASCII-only (or unpacked-hex) data , although it could be simply
by having both ends agree to AND the protocol-dependent data
with 7F hex before validating it.  I specifically am referring
to the checksum, and the block numbers and their ones-
complement.
    Those wishing to maintain compatibility of the CP/M file
structure, i.e. to allow modemming ASCII files to or from CP/M
systems should follow this data format:
  * ASCII tabs used (09H); tabs set every 8.
  * Lines terminated by CR/LF (0DH 0AH)
  * End-of-file indicated by ctl/z, 1AH.  (one or more)
  * Data is variable length, i.e. should be considered a
    continuous stream of data bytes, broken into 128-byte
    chunks purely for the purpose of transmission.

  * A CP/M "peculiarity": If the data ends exactly on a
    128-byte boundary, i.e. CR in 127, and LF in 128, a
    subsequent sector containing the ctl/z EOF character(s)
    is optional, but is preferred.  Some utilities or
    user programs still do not handle EOF without ctl/z's.
  * The last block sent is no different from others, i.e.
    there is no "short block".


 -------- 3. MESSAGE BLOCK LEVEL PROTOCOL

 Each block of the transfer looks like:
 <SOH><blk #><255-blk #><--128 data bytes--><cksum>
    in which:

 <SOH>       = 01 hex
 <blk #>     = binary number, starts at 01 increments by 1, and
               wraps 0FFH to 00H (not to 01)
 <255-blk #> = blk # after going thru 8080 "CMA" instr.
               Formally, this is the "ones complement".
 <cksum>     = the sum of the data bytes only.  Toss any carry.



 -------- 4. FILE LEVEL PROTOCOL
```

```
 ---- 4A. COMMON TO BOTH SENDER AND RECEIVER:

    All errors are retried 10 times.  For versions running with
an operator (i.e. NOT with XMODEM), a message is typed after 10
errors asking the operator whether to "retry or quit".
    Some versions of the protocol use <can>, ASCII ctl/x, to
cancel transmission.  This was never adopted as a standard, as
having a single "abort" character makes the transmission
susceptible to false termination due to an <ack> <nak> or <soh>
being corrupted into a <can> and canceling transmission.
    The protocol may be considered "receiver driven", that is,
the sender need not automatically re-transmit, although it does
in the current implementations.

 ---- 4B. RECEIVE PROGRAM CONSIDERATIONS:

    The receiver has a 10-second timeout.  It sends a <nak>
every time it times out.  The receiver's first timeout, which
sends a <nak>, signals the transmitter to start.  Optionally,
the receiver could send a <nak> immediately, in case the sender
was ready.  This would save the initial 10 second timeout.
However, the receiver MUST continue to timeout every 10 seconds
in case the sender wasn't ready.
    Once into a receiving a block, the receiver goes into a
one-second timeout for each character and the checksum.  If the
receiver wishes to <nak> a block for any reason (invalid
header, timeout receiving data), it must wait for the line to
clear.  See "programming tips" for ideas
    Synchronizing:  If a valid block number is received, it
will be: 1) the expected one, in which case everything is fine;
or 2) a repeat of the previously received block.  This should
be considered OK, and only indicates that the receivers <ack>
got glitched, and the sender re-transmitted; 3) any other block
number indicates a fatal loss of synchronization, such as the
rare case of the sender getting a line-glitch that looked like
an <ack>.  Abort the transmission, sending a <can>



 ---- 4C. SENDING PROGRAM CONSIDERATIONS.

    While waiting for transmission to begin, the sender has
only a single very long timeout, say one minute.  In the
current protocol, the sender has a 10 second timeout before
retrying.  I suggest NOT doing this, and letting the protocol
be completely receiver-driven.  This will be compatible with
```

existing programs.
    When the sender has no more data, it sends an <eot>, and awaits an <ack>, resending the <eot> if it doesn't get one. Again, the protocol could be receiver-driven, with the sender only having the high-level 1-minute timeout to abort.

-------- 5. DATA FLOW EXAMPLE INCLUDING ERROR RECOVERY

Here is a sample of the data flow, sending a 3-block message. It includes the two most common line hits - a garbaged block, and an <ack> reply getting garbaged. <xx> represents the checksum byte.

```
  SENDER                            RECIEVER
                                    Times out after 10 seconds,
                           <---     <nak>
  <soh> 01 FE -data- <xx>   --->
                           <---     <ack>
  <soh> 02 FD -data- <xx>   --->    (data gets line hit)
                           <---     <nak>
  <soh> 02 FD -data- <xx>   --->
                           <---     <ack>
  <soh> 03 FC -data- <xx>   --->
    (ack gets garbaged)    <---     <ack>
  <soh> 03 FC -data- <xx>   --->
                           <---     <ack>
  <eot>                     --->
                           <---     <ack>
```

-------- 6. PROGRAMMING TIPS.

* The character-receive subroutine should be called with a parameter specifying the number of seconds to wait. The receiver should first call it with a time of 10, then <nak> and try again, 10 times.
  After receiving the <soh>, the receiver should call the character receive subroutine with a 1-second timeout, for the remainder of the message and the <cksum>. Since they are sent as a continuous stream, timing out of this implies a serious like glitch that caused, say, 127 characters to be seen instead of 128.

* When the receiver wishes to <nak>, it should call a "PURGE" subroutine, to wait for the line to clear. Recall the sender

```
tosses any characters in its UART buffer immediately upon
completing sending a block, to ensure no glitches were mis-
interpreted.
  The most common technique is for "PURGE" to call the
character receive subroutine, specifying a 1-second timeout,
and looping back to PURGE until a timeout occurs.  The <nak> is
then sent, ensuring the other end will see it.

* You may wish to add code recommended by Jonh Mahr to your
character receive routine - to set an error flag if the UART
shows framing error, or overrun.  This will help catch a few
more glitches - the most common of which is a hit in the high
bits of the byte in two consecutive bytes.  The <cksum> comes
out OK since counting in 1-byte produces the same result of
adding 80H + 80H as with adding 00H + 00H.

============================================================

  *THE FOLLOWING IS NOT PART OF THE ORIGINAL SPECIFICATION*


A summary of two of my own modifications to the specification
to make the protocol less susceptible to failure due to delays
encountered in large networks like CIS.

I do not use a 1-second per character timeout once into
receiving a block.  I consider this overkill and not favorable
to the protocol.

On entry to Xmodem Receive initial setup is done, an initial
"nak" is sent, and a 22 second timeout clock is started.

After the first timeout this clock is then kept in 10 second
intervals until 9 more consecutive timeouts occur.  After 10
consecutive timeouts the protocol is abandoned as it is assumed
that the transmitter is not ready or the communications link
has failed.

Each character received resets the timeout clock to zero.  This
provides for the extreme possibility that the transmitter may
stop sending several times in mid-block for considerable
periods of time.  I would agree that this may be overkill also
but I believe that it is in favor of the protocol rather than
against it.

If a received block does not have the expected attributes a
"nak" is sent and a "retry count" is incremented.  After 10
```

```
consecutive retries of the same block the protocol is abandoned
as it is assumed that syncronization has been lost.

The reception of a good block resets the "retry counter" to
zero.

Naturally, if all goes well, the data will eventually be
completely received intact.

If a bad block is received a "purge" routine is invoked.  The
"purge" routine monitors the RS-232 interface until there has
been no activity for 3 seconds.  When this requirement is
satisfied the bad block just received is discarded and a "nak"
is sent.

Mike Ward
```