

# **BLUEBOX RFID System**

## **COMMUNICATION PROTOCOL**



**SAE J1939**

## Preface

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

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

Issue 1.13a  
– 25. July 2023 –

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

Subject to alteration without prior notice.  
© Copyright iDTRONIC GmbH 2023  
Printed in Germany

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

- The device may only be used for the intended purpose designed by the manufacturer. The operation manual should be 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 RFID UHF device with integrated antenna. CAN bus communication interface. EU1 (865 MHz ... 868 MHz) version.

5226U



Mid Range read / write UHF RFID (EU) device with integrated antenna. CAN bus and Ethernet 10-100M communication interface on 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). CAN bus and Ethernet 10-100M communication interface on 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. CAN bus and Ethernet 10-100M communication interface on M12 4 poles D-coded female connector. FCC (902 MHz ... 928 MHz) version.

5328U-  
FCC

**Description:**
**Order  
Number:**

Mid Range read / write UHF RFID (US) device with integrated antenna and RTC (Real Time Clock). CAN bus and Ethernet 10-100M communication interface on 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. CAN bus and Ethernet 10-100M communication interface on M12 4 poles D-coded female connector. EU1 (865 MHz ... 868 MHz) version.

5338U

Long Range read / write UHF RFID device with 1 external antenna with RTC (Real Time Clock). CAN bus and Ethernet 10-100M communication interface on 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. CAN bus and Ethernet 10-100M communication interface on 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). CAN bus and Ethernet 10-100M communication interface on 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 and RTC (Real Time Clock). 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). CAN bus and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz ... 868 MHz) version.

5428U-  
RTC

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

**NO PRODUCT  
IMAGE**

**Description:**

Mid Range read / write UHF RFID device with integrated antenna and RTC (Real Time Clock). CAN bus and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz ... 868 MHz) version.

**Order Number:**

5428-  
RTC-G

**NO PRODUCT  
IMAGE**

Long range read / write UHF RFID device with integrated antenna. 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). 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 integrated antenna. CAN bus and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz ... 868 MHz) version.

5528U-G

**NO PRODUCT  
IMAGE**

Long range read / write UHF RFID device with integrated antenna and RTC (Real Time Clock). CAN bus and Ethernet 10-100M communication interface. Grey (RAL 7045) case color. EU1 (865 MHz ... 868 MHz) version.

5528U-  
RTC-G

**This manual is valid as of firmware version:**

Order Number	Hardware Version	Firmware Version
5226U	1	1.54
5226U	2	2.54
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
5348U	2	2.67
5348U-RTC	2	2.67R
5428U	2	2.67E
5428U-RTC	2	2.67U
5428U-G	2	2.67E
5428U-RTC-G	2	2.67U
5528U	2	2.67E
5528U-RTC	2	2.67U
5528U-G	2	2.67E
5528U-RTC-G	2	2.67U

## Table of Contents

1	Introduction.....	9
2	Communication Features .....	10
2.1	SAE J1939 Message Format.....	10
2.1.1	PDU Format 1 .....	11
2.1.2	PDU Format 2 .....	11
2.1.3	Proprietary Protocol in PDU Format 1 .....	11
2.1.4	Proprietary Protocol in PDU Format 2 .....	12
2.2	Device Reset .....	12
2.3	Read Device Serial Number .....	13
2.4	Read Firmware Version .....	14
2.5	Firmware Upgrade.....	15
2.6	Read Temperature .....	16
2.7	Write Date/Time .....	16
2.8	Read Date/Time.....	17
2.9	Write ROM General Parameters.....	18
2.10	Write RAM Configuration Parameters .....	19
2.11	Write ROM Configuration Parameters .....	20
2.12	Write ROM Default Parameters.....	21
2.13	Read RAM General Parameters.....	22
2.14	Read RAM Configuration Parameters.....	23
2.15	Read ROM Configuration Parameters .....	24
2.16	'RF Reading' Test .....	24
2.17	'RF Power' Test .....	25
2.18	'RF Sensitivity' Test .....	26
2.19	Read Reflected Power .....	27
2.20	Read RSSI Power .....	28
2.21	Digital Output Activation .....	29
2.22	Read Device Status .....	30
2.23	RF Deactivation.....	31
2.24	RF Activation .....	31
2.25	Antennas Auto-Tuning .....	32
2.26	Buffer Data Request .....	32
2.27	Queue Data Request.....	35
2.28	Read Number of Records.....	38
2.29	Reset Record Database .....	38
2.30	Read Current Record .....	39
2.31	Unqueue Current Record .....	40
2.32	ISO 18000-63 Transponder 'Inventory' Command .....	41
2.33	Program EPC of an ISO 18000-63 Transponder .....	43
2.34	Read Data of an ISO 18000-63 Transponder.....	45
2.35	Write Data of an ISO 18000-63 Transponder .....	46
2.36	Lock Data of an ISO 18000-63 Transponder .....	50
2.37	'Kill' Command of an ISO 18000-63 Transponder .....	52
2.38	'QT Read' Command of a Monza 4QT Transponder.....	53

2.39	'QT Write' Command of a Monza 4QT Transponder .....	54
2.40	'Read Sensor Code' Command of a Magnus Sx Transponder .....	56
2.41	'Read On-Chip RSSI' Command of a Magnus Sx Transponder.....	57
2.42	'Read Temperature Code' Command of a Magnus S3 Transponder..	59
2.43	'Spontaneous' Message .....	60
3	Settings .....	62
3.1	Device Address and Baud Rate .....	62
3.2	Address Claiming .....	62
4	Examples .....	64
4.1	Read Temperature .....	64
4.2	Read Firmware Version .....	65
4.3	Read RAM Configuration Parameters.....	70
4.4	Write ROM Configuration Parameters.....	74
4.5	Buffer Data Request .....	79
4.6	Queue Data Request.....	91
4.7	'Spontaneous' Message.....	95
5	Document Revision History .....	99
A	Supported Commands Table .....	102



## 1 Introduction

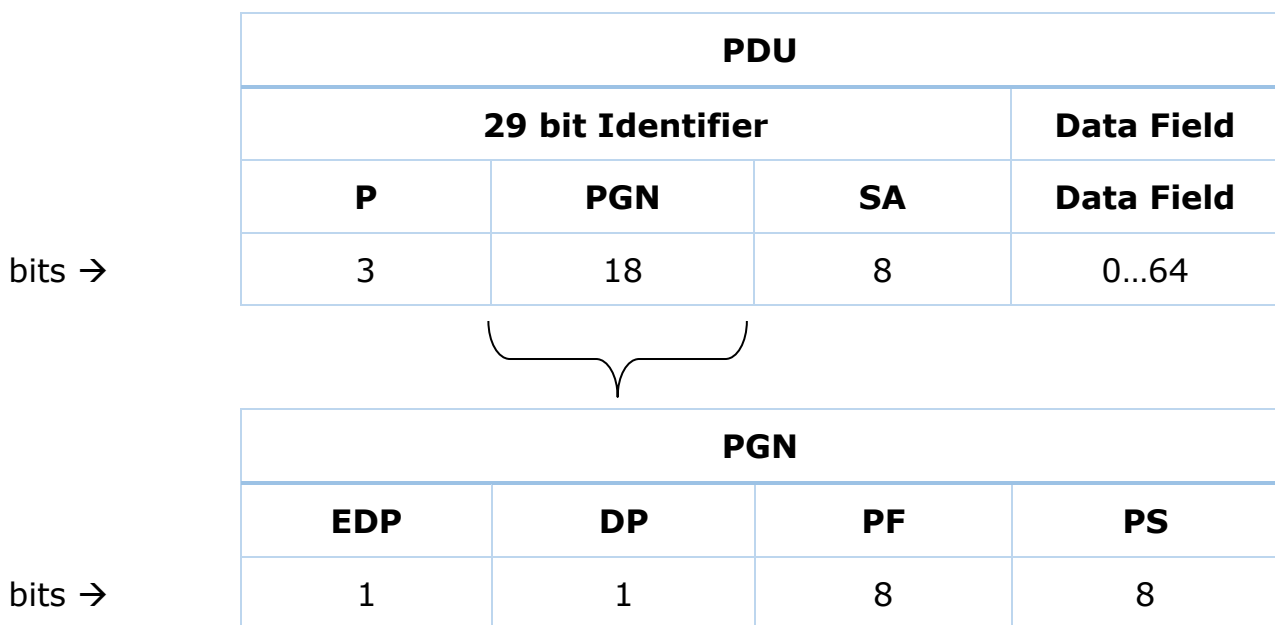
This document describes the message format of the SAE J1939 communication protocol used by the host and the **BLUEBOX** in order to issuing commands and reply with responses.

## 2 Communication Features

**BLUEBOX** has a standardised SAE J1939 interface. All measured values and parameters can be accessed via the SAE J1939 protocol. The individual configuration can be saved in the internal permanent memory.

### 2.1 SAE J1939 Message Format

A SAE J1939 message consists of a single PDU (Protocol Data Unit) and has the following data structure.



Where:

PGN	Parameter Group Number
P	Priority
EDP	Extended Data Page
DP	Data Page
PF	PDU Format
PS	PDU Specific
SA	Source Address

### 2.1.1 PDU Format 1

This format defines a message which is sent to a defined device. The PS field contains the destination address (DA) of the device.

PDU Format 1 messages are determined by the PF field. When the PF field value is 0 to 239, the message is a PDU Format 1 message.

A PF equal to 239, with EPF = 0 and DP = 0 or 1, is assigned for proprietary use.

### 2.1.2 PDU Format 2

This format defines a message which is sent globally. The PS field contains the group extension (GE).

PDU Format 2 messages are determined by the PF field. When the PF field value is 240 to 255, the message is a PDU Format 2 message.

A PF equal to 255, with EPF = 0 and DP = 0 or 1, is assigned for proprietary use.

### 2.1.3 Proprietary Protocol in PDU Format 1

To access the device the proprietary PDU Format 1 message is used. PF field is 239, the PS field contains the destination address (DA) of the device. If less than 8 bytes are required all of the 8 data bytes of the CAN frame are used and the exceeding bytes are set to 'not available' (=255). If more than 8 bytes are transferred, the SAE J1939 transport protocol must be used.

This is the general structure of a message

#### **Master -> ECU**

byte →	Len		Data		
	0	1	2	...	Len-1

#### **Master <- ECU**

byte →	Len		Data		
	0	1	2	...	Len-1

Where:

Len	Length of the data of the message without pad bytes, LSB first.
Data	Data of the message.

#### 2.1.4 Proprietary Protocol in PDU Format 2

Spontaneous messages are sent through proprietary PDU Format 2 messages. PF field is 255, the PS field contains the group extension (GE). If less than 8 bytes are required all of the 8 data bytes of the CAN frame are used and the exceeding bytes are set to 'not available' (=255). If more than 8 bytes are transferred, the SAE J1939 transport protocol must be used.

This is the general structure of a message

##### **Master -> ECU**

byte →	Len		Data		
	0	1	2	...	Len-1

##### **Master <- ECU**

byte →	Len		Data		
	0	1	2	...	Len-1

Where:

Len	Length of the data of the message without pad bytes, LSB first.
Data	Data of the message.

## 2.2 Device Reset

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

Byte	Value	Description	Notes
1	0x30	Command code	

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

Byte	Value	Description	Notes
1	0x30	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x30	Command code	
2	0x00	Status success	

### 2.3 Read Device Serial Number

This command is used to get the serial number of the **BLUEBOX** (unique for each device and assigned during the production process), the serial number is 6 bytes length.

Byte	Value	Description	Notes
1	0x2A	Command code	
2	0x01		

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

Byte	Value	Description	Notes
1	0x2A	Command code	
2	0x15	NAK status code	
3	0x01		

Otherwise it answers:

Byte	Value	Description	Notes
1	0x2A	Command code	
2	0x00	Status success	
3	0x01		
4	0x..	Serial number, 1st byte	

Byte	Value	Description	Notes
5	0x..	Serial number, 2nd byte	
6	0x..	Serial number, 3rd byte	
7	0x..	Serial number, 4th byte	
8	0x..	Serial number, 5th byte	
9	0x..	Serial number, 6th byte	



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

## 2.4 Read Firmware Version

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

Byte	Value	Description	Notes
1	0x34	Command code	

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

Byte	Value	Description	Notes
1	0x34	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x34	Command code	
2	0x00	Status success	
2+1	0x..	Firmware version, 1st ASCII char	The firmware version is 16 chars long
2+i	0x..	Firmware version, i-th ASCII char	1 < i < 16
2+16	0x..	Firmware version, 16th ASCII char	

The 16 bytes of the firmware version are a string of 16 ASCII chars that defines the firmware version.

## 2.5 Firmware Upgrade

This command is used to upgrade the firmware of the **BLUEBOX**.

Byte	Value	Description	Notes
1	0x33	Command code	
2	0x..	Index of the image file data block, MSB	0x0000: 1st data block 0x0001: 2nd data block ... 0xFFFF: start upgrade
3	0x..	Index of the image file data block, LSB	
4	0x..	Image file data block, 1st byte	Image file data block is not present if index = 0xFFFF
...	...		
132	0x..	Image file data block, 128th byte	

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

Byte	Value	Description	Notes
1	0x33	Command code	
2	0x15	NAK status code	

Otherwise it answers,

a) in case of image file data block successfully stored in external FLASH EEPROM memory:

Byte	Value	Description	Notes
1	0x33	Command code	
2	0x00	Status success	

b) in case of a not valid image file:

Byte	Value	Description	Notes
1	0x33	Command code	

2	0x02	Status image file not valid	
---	------	-----------------------------	--



Do not power off the reader during the firmware upgrade successfully started with index = 0xFFFF, the **BLUEBOX** shall resets itself at the end of the procedure.

## 2.6 Read Temperature

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

Byte	Value	Description	Notes
1	0x3A	Command code	

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

Byte	Value	Description	Notes
1	0x3A	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x3A	Command code	
2	0x00	Status success	
3	0x..	Integer value of the temperature in °C	

## 2.7 Write Date/Time

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

Byte	Value	Description	Notes
1	0x29	Command code	



Byte	Value	Description	Notes
2	0x..	Year value thousands and hundreds. BCD encoded byte	
3	0x..	Year value tens and units. BCD encoded byte	
4	0x..	Month value tens and units. BCD encoded byte	
5	0x..	Day value tens and units. BCD encoded byte	
6	0x..	Hour value tens and units. BCD encoded byte	
7	0x..	Minute value tens and units. BCD encoded byte	
8	0x..	Second value tens and units. BCD encoded byte	

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

Byte	Value	Description	Notes
1	0x29	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x29	Command code	
2	0x00	Status success	

## 2.8 Read Date/Time

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

Byte	Value	Description	Notes
1	0x28	Command code	

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

Byte	Value	Description	Notes
1	0x28	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x28	Command code	
2	0x00	Status success	
3	0x..	Year value thousands and hundreds. BCD encoded byte	
4	0x..	Year value tens and units. BCD encoded byte	
5	0x..	Month value tens and units. BCD encoded byte	
6	0x..	Day value tens and units. BCD encoded byte	
7	0x..	Hour value tens and units. BCD encoded byte	
8	0x..	Minute value tens and units. BCD encoded byte	
9	0x..	Second value tens and units. BCD encoded byte	

## 2.9 Write ROM General Parameters

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

Byte	Value	Description	Notes
1	0x2F	Command code	
2	0x..	General parameter byte 1	See the reader user manual for details
3	0x..	General parameter byte 2	See the reader user manual for details
4	0x..	General parameter byte 3	See the reader user manual for details
5	0x..	General parameter byte 4	See the reader user manual for details
6	0x..	General parameter byte 5	See the reader user manual for details
7	0x..	General parameter byte 6	See the reader user manual for details
8	0x..	General parameter byte 7	See the reader user manual for details

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

Byte	Value	Description	Notes
1	0x2F	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x2F	Command code	
2	0x00	Status success	



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

## 2.10 Write RAM Configuration Parameters

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

Byte	Value	Description	Notes
1	0x3F	Command code	
2	0x..	The configuration page (0x00 ... 0x0F, 0x80 ... 0x87, 0xC0 ... 0xCF)	
2+1	0x..	Configuration parameter byte 1	See the reader user manual for details
2+i	0x..	Configuration parameter byte i	See the reader user manual for details
2+n	0x..	Configuration parameter byte n. <ul style="list-style-type: none"> <li>n=7 if configuration page is 0x00 ... 0x0F</li> <li>n=14 if configuration page is 0x80 ... 0x87</li> <li>Variable size (max 240 bytes) null terminated string if configuration page is 0xC0 ... 0xCF</li> </ul>	See the reader user manual for details

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

Byte	Value	Description	Notes
1	0x3F	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x3F	Command code	
2	0x00	Status success	



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.

## 2.11 Write ROM Configuration Parameters

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

Byte	Value	Description	Notes
1	0x3D	Command code	
2	0x..	The configuration page (0x00 ... 0x0F, 0x80 ... 0x87, 0xC0 ... 0xCF)	
2+1	0x..	Configuration parameter byte 1	See the reader user manual for details
2+i	0x..	Configuration parameter byte i	See the reader user manual for details
2+n	0x..	Configuration parameter byte n. <ul style="list-style-type: none"> <li>n=7 if configuration page is 0x00 ... 0x0F</li> <li>n=14 if configuration page is 0x80 ... 0x87</li> <li>Variable size (max 240 bytes) null terminated string if configuration page is 0xC0 ... 0xCF</li> </ul>	See the reader user manual for details

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

Byte	Value	Description	Notes
1	0x3D	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x3D	Command code	
2	0x00	Status success	



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.

## 2.12 Write ROM Default Parameters

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

Byte	Value	Description	Notes
1	0x31	Command code	

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

Byte	Value	Description	Notes
1	0x31	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x31	Command code	
2	0x00	Status success	



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

### 2.13 Read RAM General Parameters

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

Byte	Value	Description	Notes
1	0x2A	Command code	

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

Byte	Value	Description	Notes
1	0x2A	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x2A	Command code	
2	0x00	Status success	
3	0x..	General parameter byte 1	See the reader user manual for details
4	0x..	General parameter byte 2	See the reader user manual for details
5	0x..	General parameter byte 3	See the reader user manual for details
6	0x..	General parameter byte 4	See the reader user manual for details
7	0x..	General parameter byte 5	See the reader user manual for details
8	0x..	General parameter byte 6	See the reader user manual for details

Byte	Value	Description	Notes
9	0x..	General parameter byte 7	See the reader user manual for details

## 2.14 Read RAM Configuration Parameters

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

Byte	Value	Description	Notes
1	0x3C	Command code	
2	0x..	The configuration page (0x00 ... 0x0F, 0x80 ... 0x87, 0xC0 ... 0xCF)	

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

Byte	Value	Description	Notes
1	0x3C	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x3C	Command code	
2	0x00	Status success	
2+1	0x..	Configuration parameter byte 1	See the reader user manual for details
2+i	0x..	Configuration parameter byte i	See the reader user manual for details
2+n	0x..	Configuration parameter byte n. <ul style="list-style-type: none"> <li>n=7 if configuration page is 0x00 ... 0x0F</li> <li>n=14 if configuration page is 0x80 ... 0x87</li> <li>Variable size (max 240 bytes) null terminated string if configuration page is 0xC0 ... 0xCF</li> </ul>	See the reader user manual for details

## 2.15 Read ROM Configuration Parameters

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

Byte	Value	Description	Notes
1	0x3E	Command code	
2	0x..	The configuration page (0x00 ... 0x0F, 0x80 ... 0x87, 0xC0 ... 0xCF)	

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

Byte	Value	Description	Notes
1	0x3E	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x3E	Command code	
2	0x00	Status success	
2+1	0x..	Configuration parameter byte 1	See the reader user manual for details
2+i	0x..	Configuration parameter byte i	See the reader user manual for details
2+n	0x..	Configuration parameter byte n. <ul style="list-style-type: none"> <li>n=7 if configuration page is 0x00 ... 0x0F</li> <li>n=14 if configuration page is 0x80 ... 0x87</li> <li>Variable size (max 240 bytes) null terminated string if configuration page is 0xC0 ... 0xCF</li> </ul>	See the reader user manual for details

## 2.16 'RF Reading' Test

In 'continuous' mode, this command is used to activate/deactivate the 'RF reading' test mode.

Byte	Value	Description	Notes
1	0xD7	Command code	



Byte	Value	Description	Notes
2	0x..	To activate/deactivate the 'RF reading' test mode: <ul style="list-style-type: none"> <li>0x00: To deactivate 'RF reading' test mode;</li> <li>0x01: To activate 'RF reading' test mode.</li> </ul>	

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

Byte	Value	Description	Notes
1	0xD7	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0xD7	Command code	
2	0x00	Status success	



The 'RF reading' test mode setting is stored in EEPROM and resumed at every restart of the **BLUEBOX**.

## 2.17 '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.

Byte	Value	Description	Notes
1	0xDA	Command code	
2	0x..	Antenna to use for test: <ul style="list-style-type: none"> <li>0x01 -&gt; Antenna 1.</li> <li>0x02 -&gt; Antenna 2.</li> </ul>	
3	0x..	RF channel to use for test:	

Byte	Value	Description	Notes
		<ul style="list-style-type: none"> <li>0x01 ... 0x0A if ETSI region is selected.</li> <li>0x01 ... 0x32 if FCC region is selected.</li> </ul>	

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

Byte	Value	Description	Notes
1	0xDA	Command code	
2	0x15	NAK status code	

Otherwise it answers:

a) if a tag has been identified

Byte	Value	Description	Notes
1	0xDA	Command code	
2	0x00	Status success	
3	0x..	Minimum RF output power needed to read the tag	

b) if no tag has been found

Byte	Value	Description	Notes
1	0xDA	Command code	
2	0x01	Status transponder not present	

## 2.18 '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.

Byte	Value	Description	Notes
1	0xDB	Command code	
2	0x..	Antenna to use for test: <ul style="list-style-type: none"> <li>0x01 -&gt; Antenna 1.</li> </ul>	

Byte	Value	Description	Notes
		<ul style="list-style-type: none"> <li>0x02 -&gt; Antenna 2.</li> </ul>	
3	0x..	RF channel to use for test: <ul style="list-style-type: none"> <li>0x01 ... 0x0A if ETSI region is selected.</li> <li>0x01 ... 0x32 if FCC region is selected.</li> </ul>	

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

Byte	Value	Description	Notes
1	0xDB	Command code	
2	0x15	NAK status code	

Otherwise it answers:

a) if a tag has been identified

Byte	Value	Description	Notes
1	0xDB	Command code	
2	0x00	Status success	
3	0x..	Minimum RF input sensitivity needed to read the tag	

b) if no tag has been found

Byte	Value	Description	Notes
1	0xDB	Command code	
2	0x01	Status transponder not present	

## 2.19 Read Reflected Power

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

Byte	Value	Description	Notes
1	0xFE	Command code	
2	0x..	Antenna to use for test: <ul style="list-style-type: none"> <li>0x01 -&gt; Antenna 1.</li> <li>0x02 -&gt; Antenna 2.</li> </ul>	

Byte	Value	Description	Notes
3	0x..	The frequency to test in MHz in the range 840 ... 960 MHz. MSB.	The frequency is 3 bytes length
4	0x..		
5	0x..	The frequency to test in MHz in the range 840 ... 960 MHz. LSB.	

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

Byte	Value	Description	Notes
1	0xFE	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0xFE	Command code	
2	0x00	Status success	
3	0x..	The I-channel RSSI value	
4	0x..	The Q-channel RSSI value	
5	0x..	The G value used to calculate the reflected power as defined below	

And the reflected power is calculated as follows

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

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

## 2.20 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.

Byte	Value	Description	Notes
1	0xFD	Command code	
2	0x..	Antenna to use for test: <ul style="list-style-type: none"> <li>0x01 -&gt; Antenna 1.</li> <li>0x02 -&gt; Antenna 2.</li> </ul>	
3	0x..	The frequency to test in MHz in the range 840 ... 960 MHz. MSB.	The frequency is 3 bytes length
4	0x..		
5	0x..	The frequency to test in MHz in the range 840 ... 960 MHz. LSB.	

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

Byte	Value	Description	Notes
1	0xFD	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0xFD	Command code	
2	0x00	Status success	
3	0x..	The I-channel RSSI value	
4	0x..	The Q-channel RSSI value	
5	0x..	The G value used to calculate the RSSI power as defined below	

And the RSSI power is calculated as follows

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

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

## 2.21 Digital Output Activation

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

Byte	Value	Description	Notes
1	0x37	Command code	
2	0x..	Output to activate: <ul style="list-style-type: none"> <li>0x01 -&gt; Output 1</li> <li>0x02 -&gt; Output 2</li> </ul>	
3	0x..	Activation time: <ul style="list-style-type: none"> <li>0x01 ... 0x63 (1 ... 99 seconds) -&gt; 'Impulsive' output activation</li> <li>0x81 -&gt; 'Continuous' activation</li> <li>0x80 -&gt; Deactivation</li> </ul>	

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

Byte	Value	Description	Notes
1	0x37	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x37	Command code	
2	0x00	Status success	

## 2.22 Read Device Status

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

Byte	Value	Description	Notes
1	0x36	Command code	

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

Byte	Value	Description	Notes
1	0x36	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x36	Command code	
2	0x00	Status success	
3	0x..	BLUEBOX status byte 1.	See the device user manual for details.
4	0x..	BLUEBOX status byte 2.	See the device user manual for details.

## 2.23 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.

Byte	Value	Description	Notes
1	0x38	Command code	

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

Byte	Value	Description	Notes
1	0x38	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x38	Command code	
2	0x00	Status success	

## 2.24 RF Activation

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

Byte	Value	Description	Notes
1	0x39	Command code	

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

Byte	Value	Description	Notes
1	0x39	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x39	Command code	
2	0x00	Status success	

## 2.25 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**.

Byte	Value	Description	Notes
1	0xD4	Command code	

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

Byte	Value	Description	Notes
1	0xD4	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0xD4	Command code	
2	0x00	Status success	

## 2.26 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.

Byte	Value	Description	Notes
1	0x05	Command code	

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

Byte	Value	Description	Notes
1	0x05	Command code	
2	0x15	NAK status code	

Otherwise it answers,

a) in case of at least one ISO 18000-63 (EPC Class-1 Gen-2) transponder present the **BLUEBOX** sends one message per transponder and a no transponder message at the end:

Byte	Value	Description	Notes
1	0x05	Command code	
2	0x00	Status success	
...	0x..	Transponder type: <ul style="list-style-type: none"> <li>0x02: ISO 18000-63 (EPC Class-1 Gen-2).</li> </ul>	Opt. parameter present only if the tag type information flag in the general parameters is active. See the reader user manual for more info
...	0x..	Transponder code, 1st byte.	
...	...		
...	0x..	Transponder code, n-th byte	
...	0x..	1st seen RSSI Q value in dB of the m-th tag.	Opt. parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	1st seen RSSI I value in dB of the m-th tag.	Opt. parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info

Byte	Value	Description	Notes
...	0x..	Last seen RSSI Q value in dB of the m-th tag.	Opt. parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Last seen RSSI I value in dB of the m-th tag.	Opt. parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Max seen RSSI Q value in dB of the m-th tag.	Opt. parameter present only if the max RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Max seen RSSI I value in dB of the m-th tag.	Opt. parameter present only if the max RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Reading antenna of the tag: <ul style="list-style-type: none"> <li>0x01 -&gt; Antenna 1.</li> <li>0x02 -&gt; Antenna 2.</li> </ul>	Opt. parameter present only if the reading antenna information flag in the general parameters is active. See the reader user manual for more info
...	0x..	Gate crossing direction for the identified tag: <ul style="list-style-type: none"> <li>0x01 -&gt; Crossing from input 1 to input 2.</li> <li>0x02 -&gt; Crossing from input 2 to input 1.</li> </ul>	Opt. parameter present only if 'gate' mode is active. See the reader user manual for more info
...	0x..	First seen timestamp for the identified tag. BCD encoded with the format yyyyMMddhhmmss, 1st byte.	Opt. parameter present only if the reading timestamp information flag in the general parameters is active. See the reader user manual for more info
...	...		
...	0x..	First seen timestamp for the identified tag. BCD encoded with the format yyyyMMddhhmmss, 7th byte.	
...	0x..	Last seen timestamp for the identified tag. BCD encoded with the format yyyyMMddhhmmss, 1st byte.	Opt. parameter present only if the reading timestamp information flag in the general parameters is active. See

Byte	Value	Description	Notes
			the reader user manual for more info
...	...		
...	0x..	Last seen timestamp for the identified tag. BCD encoded with the format yyyyMMddhhmmss, 7th byte.	
...	0x..	The tag read count for the identified tag. MSB byte.	Opt. parameter present only if the tag read count information flag in the RF parameters is active. See the reader user manual for more info
...	0x..	The tag read count for the identified tag. LSB byte.	Opt. parameter present only if the tag read count information flag in the RF parameters is active. See the reader user manual for more info

b) case of no transponders:

Byte	Value	Description	Notes
1	0x05	Command code	
2	0x00	Status success	
3	0x00		
4	0x00		
5	0x00		
6	0x00		
7	0x00		

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

Byte	Value	Description	Notes
1	0x06	Command code	

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

Byte	Value	Description	Notes
1	0x06	Command code	
2	0x15	NAK status code	

Otherwise it answers,

a) in case of at least one ISO 18000-63 (EPC Class-1 Gen-2) transponder present with n bytes ID length:

Byte	Value	Description	Notes
1	0x06	Command code	
2	0x00	Status success	
...	0x..	Transponder type: <ul style="list-style-type: none"> <li>0x02: ISO 18000-63 (EPC Class-1 Gen-2).</li> </ul>	Opt. parameter present only if the tag type information flag in the general parameters is active. See the reader user manual for more info
...	0x..	Transponder code, 1st byte	
...	...		
...	0x..	Transponder code, n-th byte	
...	0x..	RSSI Q value in dB of the tag.	Opt. parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	RSSI I value in dB of the tag.	Opt. parameter present only if the RSSI information flag in the RF configuration parameters is active. See the device user manual for more info
...	0x..	Reading antenna of the identified tag: <ul style="list-style-type: none"> <li>0x01 -&gt; Antenna 1.</li> </ul>	Opt. parameter present only if the reading antenna information flag in the

Byte	Value	Description	Notes
		<ul style="list-style-type: none"> <li>0x02 -&gt; Antenna 2.</li> </ul>	general parameters is active. See the reader user manual for more info
...	0x..	Gate crossing direction for the identified tag: <ul style="list-style-type: none"> <li>0x01 -&gt; Crossing from input 1 to input 2.</li> <li>0x02 -&gt; Crossing from input 2 to input 1.</li> </ul>	Opt. parameter present only if 'gate' mode is active. See the reader user manual for more info
...	0x..	Timestamp for the identified tag. BCD-coded with the format yyyyMMddhhmmss, 1st byte.	Opt. parameter present only if the reading timestamp information flag in the general parameters is active. See the reader user manual for more info
...	...		
...	0x..	Timestamp for the identified tag. BCD-coded with the format yyyyMMddhhmmss, 7th byte.	

b) case of no transponders:

Byte	Value	Description	Notes
1	0x06	Command code	
2	0x00	Status success	
3	0x00		
4	0x00		
5	0x00		
6	0x00		
7	0x00		

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

Byte	Value	Description	Notes
1	0x07	Command code	

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

Byte	Value	Description	Notes
1	0x07	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x07	Command code	
2	0x00	Status success	

## 2.28 Read Number of Records

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

Byte	Value	Description	Notes
1	0x00	Command code	

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

Byte	Value	Description	Notes
1	0x00	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x00	Command code	
2	0x00	Status success	
3	0x..	The number of unread records in the database, MSB.	
4	0x..	The number of unread records in the database, LSB.	

## 2.29 Reset Record Database

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

Byte	Value	Description	Notes
1	0x35	Command code	

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

Byte	Value	Description	Notes
1	0x35	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x35	Command code	
2	0x00	Status success	

### 2.30 Read Current Record

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

Byte	Value	Description	Notes
1	0x01	Command code	

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

Byte	Value	Description	Notes
1	0x01	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x01	Command code	
2	0x00	Status success	
3	0x..	The index of the record in the database, MSB.	
4	0x..	The index of the record in the database, LSB.	

Byte	Value	Description	Notes
5	0x..	1st byte of the record.	
...	...		
...	0x..	Last byte of the record.	



The index of the record in the answer must be used to unqueue the record from the database.

### 2.31 Unqueue Current Record

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

Byte	Value	Description	Notes
1	0x02	Command code	
2	0x..	The index of the record in the database, MSB.	
3	0x..	The index of the record in the database, LSB.	

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

Byte	Value	Description	Notes
1	0x02	Command code	
2	0x15	NAK status code	

Otherwise it answers:

Byte	Value	Description	Notes
1	0x02	Command code	
2	0x00	Status success	





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

## 2.32 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.

Byte	Value	Description	Notes
1	0x18	Command code.	

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

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x15	NAK status code	

Otherwise it answers,

a1) if at least one tag is present and flag for reading antenna disabled the **BLUEBOX** sends one message per transponder and a no transponder message at the end:

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x00	Status success	
2+1	0x..	1st transponder code, 1st byte	
...	...		
2+n	0x..	1st transponder code, n-th byte	

a2) if at least one tag is present and flag for reading antenna enabled the **BLUEBOX** sends one message per transponder and a no transponder message at the end:

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x00	Status success	
2+1	0x..	1st transponder code, 1st byte	
...	...		
2+n	0x..	1st transponder code, n-th byte	
2+n+1	0x..	Reading antenna of the 1st tag: <ul style="list-style-type: none"> <li>0x01 -&gt; Antenna 1.</li> <li>0x02 -&gt; Antenna 2.</li> </ul>	

b) if no tag is present

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x01	Status transponder not present	

This command could also be used to get the RSSI of the read transponders:

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x01		

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

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x15	NAK status code	

Otherwise it answers,

a1) if at least one tag is present and flag for reading antenna disabled the **BLUEBOX** sends one message per transponder and a no transponder message at the end:

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x00	Status success	

Byte	Value	Description	Notes
2+1	0x..	1st transponder code, 1st byte	
...	...		
2+n	0x..	1st transponder code, n-th byte	
2+n+1	0x..	The RSSI Q-Channel in dB of the 1st transponder	
2+n+2	0x..	The RSSI I-Channel in dB of the 1st transponder	

a2) if at least one tag is present and flag for reading antenna enabled the **BLUEBOX** sends one message per transponder and a no transponder message at the end:

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x00	Status success	
2+1	0x..	1st transponder code, 1st byte	
...	...		
2+n	0x..	1st transponder code, n-th byte	
2+n+1	0x..	The RSSI Q-Channel in dB of the 1st transponder	
2+n+2	0x..	The RSSI I-Channel in dB of the 1st transponder	
2+n+3	0x..	Reading antenna of the 1st tag: <ul style="list-style-type: none"> <li>• 0x01 -&gt; Antenna 1.</li> <li>• 0x02 -&gt; Antenna 2.</li> </ul>	

b) if no tag is present

Byte	Value	Description	Notes
1	0x18	Command code	
2	0x01	Status transponder not present	

### 2.33 Program EPC of an ISO 18000-63 Transponder

This command is used to program the EPC on a known (ID) ISO 18000-63 tag.

Byte	Value	Description	Notes
1	0x1E	Command code	
1+1	0x..	Transponder code, 1st byte	
...	...		
1+n	0x..	Transponder code, n-th byte	
1+n+1	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length.
		...	
1+n+4		Tag access password, 4th byte	
1+n+4+1	0x..	1st data block, 1st byte	Every data block is 2 bytes length.
1+n+4+2	0x..	1st data block, 2nd byte	
...	...		
...	0x..	m-th data block, 1st byte	
...	0x..	m-th data block, 2nd byte	

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

Byte	Value	Description	Notes
1	0x1E	Command code	
2	0x15	NAK status code	

Otherwise it answers,

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

Byte	Value	Description	Notes
1	0x1E	Command code	
2	0x00	Status success	

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

Byte	Value	Description	Notes
1	0x1E	Command code	

Byte	Value	Description	Notes
2	0x02	Status error	

c) if no tag is present

Byte	Value	Description	Notes
1	0x1E	Command code	
2	0x01	Status transponder not present	

### 2.34 Read Data of an ISO 18000-63 Transponder

This command is used to get data blocks (data block → 2 consecutive bytes) of a known IDISO 18000-63 tag.

Byte	Value	Description	Notes
1	0x19	Command code	
1+1	0x..	Transponder code, 1st byte	
...	...		
1+n	0x..	Transponder code, n-th byte	
1+n+1	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
1+n+4	0x..	Tag access password, 4th byte	
1+n+4+1	0x..	Memory bank: <ul style="list-style-type: none"> <li>0x00: Reserved.</li> <li>0x01: EPC.</li> <li>0x02: TID.</li> <li>0x03: User.</li> </ul>	
1+n+4+1+1	0x..	Memory address of the 1st byte of the 1st memory block to read, 1st byte	The memory address is 4 bytes length
...	...		
1+n+4+1+4	0x..	Memory address of the 1st byte of the 1st memory block to read, 4th byte	
1+n+4+1+4+1	0x..	Number of blocks to read (1 ... 64)	

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

Byte	Value	Description	Notes
1	0x19	Command code	
2	0x15	NAK status code	

Otherwise it answers,

a) if the addressed tag is present

Byte	Value	Description	Notes
1	0x19	Command code	
2	0x00	Status success	
2+1	0x..	1st data block, 1st byte	Every data block is 2 bytes length
2+2	0x..	1st data block, 2nd byte	
...	...		
...	0x..	n-th data block, 1st byte	
...	0x..	n-th data block, 2nd byte	

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

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

c) if no tag is present

Byte	Value	Description	Notes
1	0x19	Command code	
2	0x01	Status transponder not present	

### 2.35 Write Data of an ISO 18000-63 Transponder

This command is used to write data on a known (ID) ISO 18000-63 tag.

Byte	Value	Description	Notes
1	0x1A	Command code	
1+1	0x..	Transponder code, 1st byte	
...	...		
1+n	0x..	Transponder code, n-th byte	
1+n+1	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
1+n+4	0x..	Tag access password, 4th byte	
1+n+4+1	0x..	Memory bank: <ul style="list-style-type: none"> <li>• 0x00: Reserved.</li> <li>• 0x01: EPC.</li> <li>• 0x02: TID.</li> <li>• 0x03: User.</li> </ul>	
1+n+4+1+1	0x..	Memory address of the 1st byte of the 1st memory block to write, 1st byte	The memory address is 4 bytes length
...	...		
1+n+4+1+4	0x..	Memory address of the 1st byte of the 1st memory block to write, 4th byte	
1+n+4+1+4+1	0x..	Number of blocks to write (1 ... 64)	
1+n+4+1+4+1+1	0x..	1st data block, 1st byte	Every data block is 2 bytes length
1+n+4+1+4+1+2	0x..	1st data block, 2nd byte	
...	...		
...	0x..	m-th data block, 1st byte	
...	0x..	m-th data block, 2nd byte	

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

Byte	Value	Description	Notes
1	0x1A	Command code	
2	0x15	NAK status code	

Otherwise it answers,

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

Byte	Value	Description	Notes
1	0x1A	Command code	
2	0x00	Status success	

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

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

c) if no tag is present

Byte	Value	Description	Notes
1	0x1A	Command code	
2	0x01	Status transponder not present	

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 Gen-2 standard and not only as a loop of Write commands:

Byte	Value	Description	Notes
1	0x1D	Command code	
1+1	0x..	Transponder code, 1st byte	
...	...		
1+n	0x..	Transponder code, n-th byte	
1+n+1	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
1+n+4	0x..	Tag access password, 4th byte	
1+n+4+1	0x..	Memory bank: <ul style="list-style-type: none"> <li>• 0x00: Reserved.</li> <li>• 0x01: EPC.</li> <li>• 0x02: TID.</li> <li>• 0x03: User.</li> </ul>	



Byte	Value	Description	Notes
1+n+4+1+1	0x..	Memory address of the 1st byte of the 1st memory block to write, 1st byte	The memory address is 4 bytes length
...	...		
1+n+4+1+4	0x..	Memory address of the 1st byte of the 1st memory block to write, 4th byte	
1+n+4+1+4+1	0x..	Number of blocks to write (1 ... 64)	
1+n+4+1+4+1+1	0x..	1st data block, 1st byte	Every data block is 2 bytes length
1+n+4+1+4+1+2	0x..	1st data block, 2nd byte	
...	...		
...	0x..	m-th data block, 1st byte	
...	0x..	m-th data block, 2nd byte	

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

Byte	Value	Description	Notes
1	0x1D	Command code	
2	0x15	NAK status code	

Otherwise it answers,

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

Byte	Value	Description	Notes
1	0x1D	Command code	
2	0x00	Status success	

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

Byte	Value	Description	Notes
1	0x1D	Command code	
2	0x02	Status error	

c) if no tag is present

Byte	Value	Description	Notes
1	0x1D	Command code	
2	0x01	Status transponder not present	

### 2.36 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.

Byte	Value	Description	Notes
1	0x1B	Command code	
1+1	0x..	Transponder code, 1st byte	
...	...		
1+n	0x..	Transponder code, n-th byte	
1+n+1	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
1+n+4	0x..	Tag access password, 4th byte	
1+n+4+1	0x..	Kill password lock property. ASCII character: <ul style="list-style-type: none"> <li>0x00: Accessible from all states;</li> <li>0x01: Permanently accessible from all states and may never be locked;</li> <li>0x02: Accessible only from the secured state;</li> <li>0x03: Not accessible from any state;</li> <li>0x04: No change.</li> </ul>	
1+n+4+2	0x..	Access password lock property. ASCII character: <ul style="list-style-type: none"> <li>0x00: Accessible from all states;</li> <li>0x01: Permanently accessible from all states and may never be locked;</li> <li>0x02: Accessible only from the secured state;</li> <li>0x03: Not accessible from any state;</li> <li>0x04: No change.</li> </ul>	
1+n+4+3	0x..	EPC memory bank lock property. ASCII character: <ul style="list-style-type: none"> <li>0x00: Writable from all states;</li> </ul>	

Byte	Value	Description	Notes
		<ul style="list-style-type: none"> <li>0x01: Permanently writable from all states and may never be locked;</li> <li>0x02: Writable only from the secured state;</li> <li>0x03: Not writable from any state;</li> <li>0x04: No change.</li> </ul>	
1+n+4+4	0x..	TID memory bank lock property. ASCII character: <ul style="list-style-type: none"> <li>0x00: Writable from all states;</li> <li>0x01: Permanently writable from all states and may never be locked;</li> <li>0x02: Writable only from the secured state;</li> <li>0x03: Not writable from any state;</li> <li>0x04: No change.</li> </ul>	
1+n+4+5	0x..	User memory bank lock property. ASCII character: <ul style="list-style-type: none"> <li>0x00: Writable from all states;</li> <li>0x01: Permanently writable from all states and may never be locked;</li> <li>0x02: Writable only from the secured state;</li> <li>0x03: Not writable from any state;</li> <li>0x04: No change.</li> </ul>	

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

Byte	Value	Description	Notes
1	0x1B	Command code	
2	0x15	NAK status code	

Otherwise it answers,

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

Byte	Value	Description	Notes
1	0x1B	Command code	
2	0x00	Status success	

b) if some error is occurred during the transaction

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

c) if no tag is present

Byte	Value	Description	Notes
1	0x1B	Command code	
2	0x01	Status transponder not present	

### 2.37 'Kill' Command of an ISO 18000-63 Transponder

This command is used to kill a known (ID) ISO 18000-63 tag.

Byte	Value	Description	Notes
1	0x1C	Command code	
1+1	0x..	Transponder code, 1st byte	
...	...		
1+n	0x..	Transponder code, n-th byte	
1+n+1	0x..	Tag kill password, 1st byte	The tag kill password is 4 bytes length
...	...		
1+n+4	0x..	Tag kill password, 4th byte	

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

Byte	Value	Description	Notes
1	0x1C	Command code	
2	0x15	NAK status code	

Otherwise it answers,

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

Byte	Value	Description	Notes
1	0x1C	Command code.	
2	0x00	Status success	

b) if some error is occurred during the transaction

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

c) if no tag is present

Byte	Value	Description	Notes
1	0x1C	Command code.	
2	0x01	Status transponder not present	

### 2.38 '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.

Byte	Value	Description	Notes
1	0x20	Command code	
1+1	0x..	Transponder code, 1st byte	
...	...		
1+n	0x..	Transponder code, n-th byte	
1+n+1	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
1+n+4	0x..	Tag access password, 4th byte	

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

Byte	Value	Description	Notes
1	0x20	Command code	

Byte	Value	Description	Notes
2	0x15	NAK status code	

Otherwise it answers,

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

Byte	Value	Description	Notes
1	0x20	Command code.	
2	0x00	Status success	
3	0x..	QT control field to be written. MSB.	The QT control field is 2 bytes length
4	0x..	QT control field to be written. LSB.	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x20	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x20	Command code.	
2	0x01	Status transponder not present	

### 2.39 '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.

Byte	Value	Description	Notes
1	0x21	Command code	
1+1	0x..	Transponder code, 1st byte	
...	...		
1+n	0x..	Transponder code, n-th byte	
1+n+1	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
1+n+4	0x..	Tag access password, 4th byte	
1+n+5	0x..	The persistence. Indicates whether the QT control is written to non volatile (NVM) or volatile memory: <ul style="list-style-type: none"> <li>0x00: Write to volatile memory.</li> <li>0x01: Write to NVM.</li> </ul>	
1+n+6	0x..	QT control field to be written. MSB.	The QT control field is 2 bytes length
1+n+7	0x..	QT control field to be written. LSB.	

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

Byte	Value	Description	Notes
1	0x21	Command code	
2	0x15	NAK status code	

Otherwise it answers,

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

Byte	Value	Description	Notes
1	0x21	Command code.	
2	0x00	Status success	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x21	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x21	Command code.	
2	0x01	Status transponder not present	

#### 2.40 '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.

Byte	Value	Description	Notes
1	0x22	Command code	
1+1	0x..	The chip code: <ul style="list-style-type: none"> <li>0x02: Magnus S2.</li> <li>0x03: Magnus S3.</li> </ul>	
1+1+n	0x..	Transponder code, 1st byte	
...	...		
1+n+n	0x..	Transponder code, n-th byte	
1+1+n+1	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
1+1+n+4	0x..	Tag access password, 4th byte	

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

Byte	Value	Description	Notes
1	0x22	Command code	
2	0x15	NAK status code	

Otherwise it answers,



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

Byte	Value	Description	Notes
1	0x22	Command code.	
2	0x00	Status success	
3	0x..	The sensor code read from tag. MSB.	The sensor code is 2 bytes length
4	0x..	The sensor code read from tag. LSB.	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x22	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x22	Command code.	
2	0x01	Status transponder not present	

## 2.41 '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.

Byte	Value	Description	Notes
1	0x23	Command code	
1+1	0x..	The chip code: <ul style="list-style-type: none"> <li>0x02: Magnus S2.</li> <li>0x03: Magnus S3.</li> </ul>	
1+1+n	0x..	Transponder code, 1 <sup>st</sup> byte	
...	...		
1+n+n	0x..	Transponder code, n-th byte	
1+1+n+1	0x..	Tag access password, 1 <sup>st</sup> byte	The tag access password is 4 bytes length

Byte	Value	Description	Notes
...	...		
1+1+n+4	0x..	Tag access password, 4 <sup>th</sup> byte	
1+1+n+4+1	0x..	The RSSI threshold match criteria: <ul style="list-style-type: none"> <li>0x00: Match if code is &lt;= threshold.</li> <li>0x01: Match if code is &gt; threshold</li> </ul>	
1+1+n+4+2	0x..	The RSSI threshold in the range 0 ... 31.	

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

Byte	Value	Description	Notes
1	0x23	Command code	
2	0x15	NAK status code	

Otherwise it answers,

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

Byte	Value	Description	Notes
1	0x23	Command code.	
2	0x00	Status success	
3	0x..	The on-chip RSSI read from tag.	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x23	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x23	Command code.	
2	0x01	Status transponder not present	

## 2.42 '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.

Byte	Value	Description	Notes
1	0x24	Command code	
1+1	0x..	The chip code: • 0x03: Magnus S3.	
1+1+n	0x..	Transponder code, 1st byte	
...	...		
1+n+n	0x..	Transponder code, n-th byte	
1+1+n+1	0x..	Tag access password, 1st byte	The tag access password is 4 bytes length
...	...		
1+1+n+4	0x..	Tag access password, 4th byte	

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

Byte	Value	Description	Notes
1	0x24	Command code	
2	0x15	NAK status code	

Otherwise it answers,

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

Byte	Value	Description	Notes
1	0x24	Command code.	
2	0x00	Status success	
3	0x..	The temperature code read from tag. MSB.	The temperature code is 2 bytes length

Byte	Value	Description	Notes
4	0x..	The temperature code read from tag. LSB.	

b) if some error is occurred during the transaction

Byte	Value	Description	Notes
1	0x24	Command code.	
2	0x02	Status error.	

c) if no tag is present

Byte	Value	Description	Notes
1	0x24	Command code.	
2	0x01	Status transponder not present	

## 2.43 'Spontaneous' Message

In 'continuous' mode, if the 'spontaneous' feature is set on (see parameters), the **BLUEBOX** will send the following message every time that it will find a 'new' tag,

Byte	Value	Description	Notes
1	0x..	Transponder type: <ul style="list-style-type: none"> <li>0x02: ISO 18000-63 (EPC Class-1 Gen-2).</li> </ul>	Opt. parameter present only if the tag type information flag in the general parameters is active. See the reader user manual for more info
...	0x..	Transponder code, 1st byte.	
...	...		
...	0x..	Transponder code, n-th byte	
...	0x..	1st seen RSSI Q value in dB of the tag.	Opt. parameter present only if the RSSI information flag in the EPC C1G2 configuration parameters is active. See the device user manual for more info
...	0x..	1st seen RSSI I value in dB of the tag.	Opt. parameter present only if the RSSI information flag in the

Byte	Value	Description	Notes
			EPC C1G2 configuration parameters is active. See the device user manual for more info
...	0x..	Reading antenna of the tag: <ul style="list-style-type: none"> <li>• 0x01 -&gt; Antenna 1.</li> <li>• 0x02 -&gt; Antenna 2.</li> </ul>	Opt. parameter present only if the reading antenna information flag in the general parameters is active. See the reader user manual for more info
...	0x..	Gate crossing direction for the identified tag: <ul style="list-style-type: none"> <li>• 0x01 -&gt; Crossing from input 1 to input 2.</li> <li>• 0x02 -&gt; Crossing from input 2 to input 1.</li> </ul>	Opt. parameter present only if 'gate' mode is active. See the reader user manual for more info
...	0x..	Timestamp for the identified tag. BCD encoded with the format yyyyMMddhhmmss, 1st byte.	Opt. parameter present only if the reading timestamp information flag in the general parameters is active. See the reader user manual for more info
...	...		
...	0x..	Timestamp for the identified tag. BCD encoded with the format yyyyMMddhhmmss, 7th byte.	

### 3 Settings

#### 3.1 Device Address and Baud Rate

Valid values for the device address are 0 to 253. The preset device address should be preferably used, but via 'address claiming' the device address can be changed.

Valid values for the baud rate are: 250 kbit/s and 500 kbit/s.



The **BLUEBOX** is supplied with the following default settings:

Device address 235

Baud rate 250 kbit/s



Make sure to set the **BLUEBOX** address only once in the CAN network. Use 'address claiming' to avoid conflicts.

#### 3.2 Address Claiming

The **BLUEBOX** supports 'dynamic address claiming'. The default address is 235.

With this address, the **BLUEBOX** logs in to the network during start-up. Unless there is an address conflict with other network participants, the **BLUEBOX** starts communication automatically.

Arbitrary address capable (CA): if the set address of the **BLUEBOX** is already used in the network, the participant with a higher priority will be accepted by the network. The rejected network participant with a lower priority will be assigned another valid device address.



The **BLUEBOX** tries to claim the preferred device address first. If a participant with higher priority claims this device address, the **BLUEBOX** will look for another valid device address.

## 4 Examples

Hereinafter the temperature reading, firmware version reading, the data request and the queue data request commands usage example.

### 4.1 Read Temperature

The master device requests the firmware version of the **BLUEBOX**; for the example, we suppose:

- Address target unit (ECU): 0xEB
- Address control unit (master): 0x14
- Temperature: 37 °C = 0x25

#### Master -> ECU

29 bit Identifier	Data Frame							
0x18EFEB14	0x01	0x00	0x3A	0xFF	0xFF	0xFF	0xFF	0xFF

And in detail:

- Priority: 6
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 239
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 61184 (Proprietary A)
- Data Length: 8

Byte#	Value	Description
0...1	0x0100	Length of the data of the message = 1
2	0x3A	Command code = 0x3A
3...7	0xFFFFFFFF FFF	Pad with 'not available' byte



## **Master <- ECU**

29 bit Identifier	Data Frame							
0x18EF14EB	0x03	0x00	0x3A	0x00	0x25	0xFF	0xFF	0xFF

And in detail:

- Priority: 6
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 239
- Destination Address: 235
- Source Address: 235
- Parameter Group Number: 61184 (Proprietary A)
- Data Length: 8

Byte#	Value	Description
0...1	0x0300	Length of the data of the message = 3
2	0x3A	Command code = 0x3A
3	0x00	Status = OK
4	0x25	Temperature in hexadecimal value 0 37 °C
5...7	0xFFFFFFFF	Pad with 'not available' byte

## 4.2 Read Firmware Version

The master device requests the firmware version of the **BLUEBOX**; for the example, we suppose:

- Address target unit (ECU): 0xEB
- Address control unit (master): 0x14
- Firmware version: BLUEBOXUHF 2.40

## **Master -> ECU**

29 bit Identifier	Data Frame							
0x18EFEB14	0x01	0x00	0x34	0xFF	0xFF	0xFF	0xFF	0xFF

And in detail:

- Priority: 6
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 239
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 61184 (Proprietary A)
- Data Length: 8

Byte#	Value	Description
0...1	0x0100	Length of the data of the message = 1
2	0x34	Command code = 0x34
3...7	0xFFFFFFFF FFF	Pad with 'not available' byte

### **Master <- ECU**

29 bit Identifier	Data Frame							
0x1CEC14EB	0x10	0x14	0x00	0x03	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x10	Connection mode request to send = 16
1...2	0x1400	Total message size = 20 bytes
3	0x03	Total number of packets = 3

Byte#	Value	Description
4	0xFF	Maximum number of packets that can be sent in response to one CTS = 0xFF (no limits exists for the originator)
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master -> ECU**

29 bit Identifier	Data Frame							
0x1CECEB14	0x11	0x03	0x01	0xFF	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x11	Connection mode clear to send = 17
1	0x03	Number of packets that can be sent = 3
2	0x01	Next packet number to be sent = 1
3...4	0xFFFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master <- ECU**

29 bit Identifier	Data Frame							
0x1CEB14EB	0x01	0x12	0x00	0x34	0x00	0x42	0x4C	0x55

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x01	Sequence number = 1
1...7	0x120034 00424C55	Packetized data: <ul style="list-style-type: none"> <li>• Length of the data of the message = 0x1200 = 18</li> <li>• Command code = 0x34</li> <li>• Status success = 0x00</li> <li>• Firmware version (1...3) = 0x424C55</li> </ul>

29 bit Identifier	Data Frame							
0x1CEB14EB	0x02	0x45	0x42	0x4F	0x58	0x55	0x48	0x46

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x02	Sequence number = 2
1...7	0x45424F 58554846	Packetized data: <ul style="list-style-type: none"> <li>• Firmware version (4...10) = 0x45424F58554846</li> </ul>

29 bit Identifier	Data Frame							
0x1CEB14EB	0x03	0x20	0x32	0x2E	0x34	0x30	0x20	0xFF

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x03	Sequence number = 3
1...6	0x20322E 343020	Packetized data: <ul style="list-style-type: none"> <li>• Firmware version (11...16) = 0x20322E343020</li> </ul>
7	0xFF	Pad data = 0xFF

### **Master -> ECU**

29 bit Identifier	Data Frame							
0x1CECEB14	0x13	0x14	0x00	0x03	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x13	Connection mode end of message ACK = 19
1...2	0x1400	Total message size = 20 bytes
3	0x03	Total number of packets = 3
4	0xFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### 4.3 Read RAM Configuration Parameters

The master device requests the RF configuration parameters (page 0x02) of the **BLUEBOX**; for the example, we suppose:

- Address target unit (ECU): 0xEB
- Address control unit (master): 0x14
- RF configuration parameters: 0x02140001300000
  - RF region: 0x02 (Europe)
  - RF output power = 0x14 (20dBm)
  - RF channel = 0x00 (default)
  - RF antenna= 0x01 (antenna #1)
  - EPC C1G2 = 0x30
    - Q=3
    - dynamic Q algorithm
    - session S0
    - target A
  - RF max allocation time = 0x00 (default)
  - RF min allocation time = 0x00 (default)

#### **Master -> ECU**

29 bit Identifier	Data Frame							
0x18EFEB14	0x02	0x00	0x3C	0x02	0xFF	0xFF	0xFF	0xFF

And in detail:

- Priority: 6
- Extended Data Page: 0
- Date Page: 0

- PDU Format: 239
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 61184 (Proprietary A)
- Data Length: 8

Byte#	Value	Description
0...1	0x0200	Length of the data of the message = 2
2	0x3C	Command code = 0x3C
3	0x02	Page number = 0x02
4...7	0xFFFFFFFF F	Pad with 'not available' byte

### **Master <- ECU**

29 bit Identifier	Data Frame							
0x1CEC14EB	0x10	0x0B	0x00	0x02	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x10	Connection mode request to send = 16
1...2	0x0B00	Total message size = 11 bytes
3	0x02	Total number of packets = 2
4	0xFF	Maximum number of packets that can be sent in response to one CTS = 0xFF (no limits exists for the originator)
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master -> ECU**

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CECEB14	0x11	0x02	0x01	0xFF	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

<b>Byte#</b>	<b>Value</b>	<b>Description</b>
0	0x11	Connection mode clear to send = 17
1	0x02	Number of packets that can be sent = 2
2	0x01	Next packet number to be sent = 1
3...4	0xFFFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master <- ECU**

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CEB14EB	0x01	0x09	0x00	0x3C	0x00	0x02	0x14	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235



- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x01	Sequence number = 1
1...7	0x09003C 00021400	Packetized data: <ul style="list-style-type: none"> <li>• Length of the data of the message = 0x0900 = 9</li> <li>• Command code = 0x3C</li> <li>• Status success = 0x00</li> <li>• Configuration parameters (1...3) = 0x021400</li> </ul>

29 bit Identifier	Data Frame							
0x1CEB14EB	0x02	0x01	0x30	0x00	0x00	0xFF	0xFF	0xFF

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x02	Sequence number = 2
1...4	0x013000 00	Packetized data: <ul style="list-style-type: none"> <li>• Configuration parameters (4...7) = 0x01300000</li> </ul>
5...7	0xFFFFFFFF	Pad data = 0xFFFFFFFF

### Master -> ECU

29 bit Identifier	Data Frame							
0x1CECEB14	0x13	0x0B	0x00	0x02	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x13	Connection mode end of message ACK = 19
1...2	0x0B00	Total message size = 11 bytes
3	0x02	Total number of packets = 2
4	0xFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

#### 4.4 Write ROM Configuration Parameters

The master device requests to write the RF configuration parameters (page 0x02) of the **BLUEBOX**; for the example, we suppose:

- Address target unit (ECU): 0xEB
- Address control unit (master): 0x14
- RF configuration parameters: 0x011E0001300000
  - RF region: 0x01 (North America)
  - RF output power = 0x1E (30dBm)
  - RF channel = 0x00 (default)
  - RF antenna= 0x01 (antenna #1)
  - EPC C1G2 = 0x30
    - Q=3
    - dynamic Q algorithm
    - session S0
    - target A
  - RF max allocation time = 0x00 (default)
  - RF min allocation time = 0x00 (default)

### **Master -> ECU**

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CECEB14	0x10	0x0B	0x00	0x02	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

<b>Byte#</b>	<b>Value</b>	<b>Description</b>
0	0x10	Connection mode request to send = 16
1...2	0x0B00	Total message size = 11 bytes
3	0x02	Total number of packets = 2
4	0xFF	Maximum number of packets that can be sent in response to one CTS = 0xFF (no limits exists for the originator)
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master <- ECU**

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CEC14EB	0x11	0x01	0x01	0xFF	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 20
- Source Address: 235

- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x11	Connection mode clear to send = 17
1	0x01	Number of packets that can be sent = 1
2	0x01	Next packet number to be sent = 1
3...4	0xFFFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master -> ECU**

29 bit Identifier	Data Frame							
0x1CEBEB14	0x01	0x09	0x00	0x3D	0x02	0x01	0x1E	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x01	Sequence number = 1
1...7	0x09003D 02011E00	Packetized data: <ul style="list-style-type: none"> <li>• Length of the data of the message = 0x0900 = 9</li> <li>• Command code = 0x3D</li> <li>• Page number = 0x02</li> <li>• Configuration parameters (1...3) = 0x011E00</li> </ul>

### **Master <- ECU**

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CEC14EB	0x11	0x01	0x02	0xFF	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

<b>Byte#</b>	<b>Value</b>	<b>Description</b>
0	0x11	Connection mode clear to send = 17
1	0x01	Number of packets that can be sent = 1
2	0x02	Next packet number to be sent = 2
3...4	0xFFFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master -> ECU**

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CEBEB14	0x02	0x01	0x30	0x00	0x00	0xFF	0xFF	0xFF

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60160 (TP.DT)

- Data Length: 8

Byte#	Value	Description
0	0x02	Sequence number = 2
1...4	0x013000 00	Packetized data: <ul style="list-style-type: none"> <li>• Configuration parameters (4...7) = 0x01300000</li> </ul>
5...7	0xFFFFFFFF	Pad data = 0xFFFFFFFF

### **Master <- ECU**

29 bit Identifier	Data Frame							
0x1CEC14EB	0x13	0x0B	0x00	0x02	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x13	Connection mode end of message ACK = 19
1...2	0x0B00	Total message size = 11 bytes
3	0x02	Total number of packets = 2
4	0xFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

29 bit Identifier	Data Frame							
0x18EF14EB	0x02	0x00	0x3D	0x00	0xFF	0xFF	0xFF	0xFF

And in detail:

- Priority: 6
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 239
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 61184 (Proprietary A)
- Data Length: 8

Byte#	Value	Description
0...1	0x0200	Length of the data of the message = 2
2	0x3D	Command code = 0x3D
3	0x00	Status success = 0x00
4...7	0xFFFFFFFF F	Pad with 'not available' byte

#### 4.5 Buffer Data Request

This command sends back the code of the eventual transponder that is present in the buffer; for the example, we suppose that two transponders were found, transponder info, antenna info, RSSI info and tag read count info are not active, and the device is not in 'gate' mode:

- Address target unit (ECU): 0xEB
- Address control unit (master): 0x14
- 1st tag code: 0x3000E200408067100151253014E75466
- 2nd tag code: 0x3000E200408067100137253014C855B2

#### **Master -> ECU**

29 bit Identifier	Data Frame							
0x18EFEB14	0x01	0x00	0x05	0xFF	0xFF	0xFF	0xFF	0xFF

And in detail:

- Priority: 6
- Extended Data Page: 0
- Date Page: 0

- PDU Format: 239
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 61184 (Proprietary A)
- Data Length: 8

Byte#	Value	Description
0...1	0x0100	Length of the data of the message = 1
2	0x05	Command code = 0x05
3...7	0xFFFFFFFF FFF	Pad with 'not available' byte

### **Master <- ECU**

29 bit Identifier	Data Frame							
0x1CEC14EB	0x10	0x14	0x00	0x03	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x10	Connection mode request to send = 16
1...2	0x1400	Total message size = 20 bytes
3	0x03	Total number of packets = 3
4	0xFF	Maximum number of packets that can be sent in response to one CTS = 0xFF (no limits exists for the originator)
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)



### **Master -> ECU**

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CECEB14	0x11	0x03	0x01	0xFF	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

<b>Byte#</b>	<b>Value</b>	<b>Description</b>
0	0x11	Connection mode clear to send = 17
1	0x03	Number of packets that can be sent = 5
2	0x01	Next packet number to be sent = 1
3...4	0xFFFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master <- ECU**

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CEB14EB	0x01	0x12	0x00	0x05	0x00	0x30	0x00	0xE2

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)

- Data Length: 8

Byte#	Value	Description
0	0x01	Sequence number = 1
1...7	0x120005 003000E2	Packetized data: <ul style="list-style-type: none"> <li>• Length of the data of the message = 0x1200 = 18</li> <li>• Command code = 0x05</li> <li>• Status success = 0x00</li> <li>• Tag code (1...3) = 0x3000E2</li> </ul>

29 bit Identifier	Data Frame							
0x1CEB14EB	0x02	0x00	0x40	0x80	0x67	0x10	0x01	0x51

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x02	Sequence number = 2
1...7	0x004080 67100151	Packetized data: <ul style="list-style-type: none"> <li>• Tag code (4...10) = 0x00408067100151</li> </ul>

29 bit Identifier	Data Frame							
0x1CEB14EB	0x03	0x25	0x30	0x14	0xE7	0x54	0x66	0xFF

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x03	Sequence number = 3
1...6	0x253014 E75466	Packetized data: <ul style="list-style-type: none"> <li>• Tag code (11...16) = 0x253014E75466</li> </ul>
7	0xFF	Pad data = 0xFF

### **Master -> ECU**

29 bit Identifier	Data Frame							
0x1CECEB14	0x13	0x14	0x00	0x03	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x13	Connection mode end of message ACK = 19
1...2	0x1400	Total message size = 20 bytes
3	0x03	Total number of packets = 3

Byte#	Value	Description
4	0xFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master <- ECU**

29 bit Identifier	Data Frame							
0x1CEC14EB	0x10	0x14	0x00	0x03	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x10	Connection mode request to send = 16
1...2	0x1400	Total message size = 20 bytes
3	0x03	Total number of packets = 3
4	0xFF	Maximum number of packets that can be sent in response to one CTS = 0xFF (no limits exists for the originator)
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master -> ECU**

29 bit Identifier	Data Frame							
0x1CECEB14	0x11	0x03	0x01	0xFF	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x11	Connection mode clear to send = 17
1	0x03	Number of packets that can be sent = 3
2	0x01	Next packet number to be sent = 1
3...4	0xFFFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master <- ECU**

29 bit Identifier	Data Frame							
0x1CEB14EB	0x01	0x12	0x00	0x05	0x00	0x30	0x00	0xE2

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x01	Sequence number = 1

Byte#	Value	Description
1...7	0x120005 003000E2 0040	Packetized data: <ul style="list-style-type: none"> <li>Length of the data of the message = 0x1200 = 18</li> <li>Command code = 0x05</li> <li>Status success = 0x00</li> <li>Tag code (1...3) = 0x3000E2</li> </ul>

29 bit Identifier	Data Frame							
0x1CEB14EB	0x02	0x00	0x40	0x80	0x67	0x10	0x01	0x37

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x02	Sequence number = 2
1...7	0x004080 67100137	Packetized data: <ul style="list-style-type: none"> <li>Tag code (4...10) = 0x00408067100137</li> </ul>

29 bit Identifier	Data Frame							
0x1CEB14EB	0x03	0x25	0x30	0x14	0xC8	0x55	0xB2	0xFF

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20

- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x03	Sequence number = 3
1...6	0x253014 C855B2	Packetized data: <ul style="list-style-type: none"> <li>• Tag code (11...16) = 0x253014C855B2</li> </ul>
7	0xFF	Pad data = 0xFF

### **Master -> ECU**

29 bit Identifier	Data Frame							
0x1CECEB14	0x13	0x14	0x00	0x03	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x13	Connection mode end of message ACK = 19
1...2	0x1400	Total message size = 20 bytes
3	0x03	Total number of packets = 3
4	0xFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master <- ECU**

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CEC14EB	0x10	0x09	0x00	0x02	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

<b>Byte#</b>	<b>Value</b>	<b>Description</b>
0	0x10	Connection mode request to send = 16
1...2	0x0900	Total message size = 9 bytes
3	0x02	Total number of packets = 2
4	0xFF	Maximum number of packets that can be sent in response to one CTS = 0xFF (no limits exists for the originator)
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master -> ECU**

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CECEB14	0x11	0x02	0x01	0xFF	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20



- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x11	Connection mode clear to send = 17
1	0x02	Number of packets that can be sent = 2
2	0x01	Next packet number to be sent = 1
3...4	0xFFFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master <- ECU**

29 bit Identifier	Data Frame							
0x1CEB14EB	0x01	0x07	0x00	0x05	0x00	0x00	0x00	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x01	Sequence number = 1
1...7	0x070005 00000000	Packetized data: <ul style="list-style-type: none"> <li>• Length of the data of the message = 0x0700 = 7</li> <li>• Command code = 0x05</li> <li>• Status success = 0x00</li> <li>• No tag (1...4) = 0x00000000</li> </ul>

29 bit Identifier	Data Frame							
0x1CEB14EB	0x02	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x02	Sequence number = 2
1	0x00	Packetized data: <ul style="list-style-type: none"> <li>• No tag (5) = 0x00</li> </ul>
2...7	0xFFFFFFFF FFFF	Pad data = 0xFFFFFFFFFFFF

### **Master -> ECU**

29 bit Identifier	Data Frame							
0x1CECEB14	0x13	0x09	0x00	0x02	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x13	Connection mode end of message ACK = 19
1...2	0x0900	Total message size = 9 bytes
3	0x02	Total number of packets = 2
4	0xFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

#### 4.6 Queue Data Request

The master device requests the first present code in the queue; for the example, we suppose that transponder info, antenna info and RSSI info are not active, and the device is not in 'gate' mode:

- Address target unit (ECU): 0xEB
- Address control unit (master): 0x14
- Tag code: 0x3000E200408067100151253014E75466

#### **Master -> ECU**

29 bit Identifier	Data Frame							
0x18EFEB14	0x01	0x00	0x06	0xFF	0xFF	0xFF	0xFF	0xFF

And in detail:

- Priority: 6
- Extended Data Page: 0
- Data Page: 0
- PDU Format: 239
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 61184 (Proprietary A)
- Data Length: 8

Byte#	Value	Description
0...1	0x0100	Length of the data of the message = 1

Byte#	Value	Description
2	0x06	Command code = 0x06
3...7	0xFFFFFFFF FFF	Pad with 'not available' byte

### **Master <- ECU**

29 bit Identifier	Data Frame							
0x1CEC14EB	0x10	0x14	0x00	0x03	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x10	Connection mode request to send = 16
1...2	0x1400	Total message size = 20 bytes
3	0x03	Total number of packets = 3
4	0xFF	Maximum number of packets that can be sent in response to one CTS = 0xFF (no limits exists for the originator)
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master -> ECU**

29 bit Identifier	Data Frame							
0x1CECEB14	0x11	0x03	0x01	0xFF	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x11	Connection mode clear to send = 17
1	0x03	Number of packets that can be sent = 3
2	0x01	Next packet number to be sent = 1
3...4	0xFFFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

### **Master <- ECU**

29 bit Identifier	Data Frame							
0x1CEB14EB	0x01	0x12	0x00	0x06	0x00	0x30	0x00	0xE2

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x01	Sequence number = 1
1...7	0x120006 003000E2	Packetized data: <ul style="list-style-type: none"> <li>• Length of the data of the message = 0x1200 = 18</li> <li>• Command code = 0x06</li> </ul>

Byte#	Value	Description
		<ul style="list-style-type: none"> <li>Status success = 0x00</li> <li>Tag code (1...3) = 0x3000E2</li> </ul>

29 bit Identifier	Data Frame							
0x1CEB14EB	0x02	0x00	0x40	0x80	0x67	0x10	0x01	0x51

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x02	Sequence number = 2
1...7	0x004080 67100151	Packetized data: <ul style="list-style-type: none"> <li>• Tag code (4...10) = 0x00408067100151</li> </ul>

29 bit Identifier	Data Frame							
0x1CEB14EB	0x03	0x25	0x30	0x14	0xE7	0x54	0x66	0xFF

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: 20
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x03	Sequence number = 3
1...6	0x253014 E75466	Packetized data: <ul style="list-style-type: none"> <li>Tag code (11...16) = 0x253014E75466</li> </ul>
7	0xFF	Pad data = 0xFF

### **Master -> ECU**

29 bit Identifier	Data Frame							
0x1CECEB14	0x13	0x14	0x00	0x03	0xFF	0x00	0xEF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: 235
- Source Address: 20
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

Byte#	Value	Description
0	0x13	Connection mode end of message ACK = 19
1...2	0x1400	Total message size = 20 bytes
3	0x03	Total number of packets = 3
4	0xFF	Reserved
5...7	0x00EF00	Parameter Group Number = 61184 (Proprietary A)

## 4.7 'Spontaneous' Message

In 'continuous' mode, if the 'spontaneous' feature is set on (see parameters), the **BLUEBOX** will send the following message every time that it will find a 'new' tag; for the example, we suppose that transponder info, antenna info and RSSI info are not active, and the device is not in 'gate' mode:

- Address target unit (ECU): 0xEB

- Address control unit (master): 0x14
- Tag code: 0x3000E200408067100151253014E75466

### **Master <- ECU**

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CECFFEB	0x20	0x12	0x00	0x03	0xFF	0x00	0xFF	0x00

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 236
- Destination Address: Global
- Source Address: 235
- Parameter Group Number: 60416 (TP.CM)
- Data Length: 8

<b>Byte#</b>	<b>Value</b>	<b>Description</b>
0	0x20	Connection mode broadcast announce message = 32
1...2	0x1200	Total message size = 18 bytes
3	0x03	Total number of packets = 3
4	0xFF	Maximum number of packets that can be sent in response to one CTS = 0xFF (no limits exists for the originator)
5...7	0x00FF00	Parameter Group Number = 65280 (Proprietary B)

<b>29 bit Identifier</b>	<b>Data Frame</b>							
0x1CEBFFEB	0x01	0x10	0x00	0x30	0x00	0xE2	0x00	0x40

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: Global



- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x01	Sequence number = 1
1...7	0x100030 00E20040	Packetized data: <ul style="list-style-type: none"> <li>• Length of the data of the message = 0x1000 = 16</li> <li>• Tag code (1...5) = 0x3000E20040</li> </ul>

29 bit Identifier	Data Frame							
0x1CEBFFEB	0x02	0x80	0x67	0x10	0x01	0x51	0x25	0x30

And in detail:

- Priority: 7
- Extended Data Page: 0
- Date Page: 0
- PDU Format: 235
- Destination Address: Global
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x02	Sequence number = 2
1...7	0x806710 01512530	Packetized data: <ul style="list-style-type: none"> <li>• Tag code (6...12) = 0x80671001512530</li> </ul>

29 bit Identifier	Data Frame							
0x1CEBFFEB	0x03	0x14	0xE7	0x54	0x66	0xFF	0xFF	0xFF

And in detail:

- Priority: 7
- Extended Data Page: 0

- Date Page: 0
- PDU Format: 235
- Destination Address: Global
- Source Address: 235
- Parameter Group Number: 60160 (TP.DT)
- Data Length: 8

Byte#	Value	Description
0	0x03	Sequence number = 3
1...4	0x14E754 66	Packetized data: <ul style="list-style-type: none"> <li>• Tag code (13...16) = 0x14E75466</li> </ul>
5...7	0xFFFFFFFF	Pad data = 0xFFFFFFFF

## 5 Document Revision History

Date	Revision	Description
13/04/17	1.00	Initial release.
12/05/17	1.01	<p>Added 5428U[-RTC][-G] and 5528U[-RTC][-G] readers support to this manual.</p> <p>Corrected 'Proprietary Protocol' description in section 2.13.</p> <p>Changed the 'Read Device Serial Number' command reply (section 2.3).</p> <p>Changed the 'Buffer Data Request' command reply (section 2.21).</p> <p>Changed the 'Queue Data Request' command reply (section 2.22).</p> <p>Changed the 'ISO 18000-6C Transponder Inventory Command' reply (section 2.23).</p> <p>Added usage examples (section 4).</p>
27/06/17	1.02	<p>Corrected error in 'Read Firmware Version' example in section 4.1.</p> <p>Added the 'Read Configuration Parameters' and 'Write Configuration Parameters' examples (sections 4.2 and 4.3).</p>
12/07/17	1.03	<p>Updated the device firmware versions object of this manual.</p> <p>Corrected the Proprietary Protocol in PDU Format 1 with messages with data length less than 9.</p> <p>Added the Proprietary Protocol in PDU Format 2 section.</p> <p>Changed the data message format in Proprietary Protocol in PDU Format 1 and 2.</p> <p>Added the variable size (max 240 bytes) null terminated string parameters management in 'Read Configuration Parameters' and 'Write Configuration Parameters' commands.</p> <p>Corrected the examples due to the corrections and changes in Proprietary Protocol in PDU Format 1.</p> <p>Added the 'spontaneous' message example.</p>
15/01/18	1.04	<p>Updated reader's firmware versions object of this manual.</p> <p>Added the 'RF Sensitivity' Test, Read Reflected Power and Read RSSI Power commands.</p>

Date	Revision	Description
		<p>Added the reading timestamp information to 'Buffer Data Request', 'Queue Data Request' and 'Spontaneous Message'.</p> <p>Added the 'QT Read' and 'QT Write' commands of an Impinj Monza 4QT transponder.</p> <p>Added the 'Read Sensor Code' and 'Read On-Chip RSSI' commands of an RFMicron Magnus S2 and S3 transponders.</p> <p>Added the 'Read Temperature Code' of an RFMicron Magnus S3.</p>
08/06/18	1.05	<p>Added 5226U reader support to this manual.</p> <p>Updated the supported commands table.</p>
01/08/18	1.06	<p>Updated reader's firmware versions object of this manual.</p> <p>Added the RSSI Q and I channel info in 'Buffer Data Request' and 'Queue Data Request' commands and 'Spontaneous Message'.</p>
22/10/18	1.07	<p>Updated reader's firmware versions object of this manual.</p> <p>Added the tag read count info in 'Buffer Data Request'.</p> <p>Updated the 'Buffer Data Request', 'Queue Data Request' and 'Spontaneous Message' examples.</p>
11/01/19	1.08	<p>Updated the company name/logo and BLUEBOX logo.</p> <p>Updated reader's firmware versions object of this manual.</p> <p>Added the max RSSI Q and I channel info in 'Buffer Data Request'.</p>
02/09/19	1.09	<p>Updated reader's firmware versions object of this manual.</p> <p>Replaced 'Write General Parameters' command with 'Write ROM General Parameters' command.</p> <p>Replaced 'Write Configuration Parameters' command with 'Write ROM Configuration Parameters' command.</p> <p>Added the 'Write RAM Configuration Parameters' command.</p> <p>Replaced the 'Set Default Parameters' command with 'Write ROM Default Parameters' command.</p> <p>Replaced the 'Read General Parameters' command with 'Read RAM General Parameters' command.</p>

Date	Revision	Description
		Replaced the 'Read Configuration Parameters' command with 'Read RAM Configuration Parameters' command. Added the 'Read ROM Configuration Parameters' command. Updated the supported commands tables.
10/12/19	1.10	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.
04/05/20	1.11	Updated the reader's description object of this manual. Updated the supported commands tables. Format changes and document fixes in all sections.
16/12/20	1.12	Corrected the usage examples.
29/01/21	1.13	Corrected error in 'spontaneous' message usage example.
25/07/23	1.13a	Example with Read Temperature added.

## A Supported Commands Table

	5226U	53x8U	53x8U-RTC	5428U	5428U-RTC	5528U	5528U-RTC
Device Reset	✓	✓	✓	✓	✓	✓	✓
Read Device Serial Number	✓	✓	✓	✓	✓	✓	✓
Read Firmware Version	✓	✓	✓	✓	✓	✓	✓
Firmware Upgrade	✓	✓	✓	✓	✓	✓	✓
Read Temperature	✓	✓	✓	✓	✓	✓	✓
Write Date/Time			✓		✓		✓
Read Date/Time			✓		✓		✓
Write General Parameters	✓	✓	✓	✓	✓	✓	✓
Write ROM General Parameters	✓	✓	✓	✓	✓	✓	✓
Write RAM Configuration Parameters	✓	✓	✓	✓	✓	✓	✓
Write ROM Configuration Parameters	✓	✓	✓	✓	✓	✓	✓
Write ROM Default Parameters	✓	✓	✓	✓	✓	✓	✓
Read RAM General Parameters	✓	✓	✓	✓	✓	✓	✓
Read RAM Configuration Parameters	✓	✓	✓	✓	✓	✓	✓
'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		✓	✓	✓	✓	✓	✓
Date Request	✓	✓	✓	✓	✓	✓	✓
Queue Data Request	✓	✓	✓	✓	✓	✓	✓
Read Number of Records		✓	✓	✓	✓	✓	✓
Reset Record Database		✓	✓	✓	✓	✓	✓
Read Current Record		✓	✓	✓	✓	✓	✓
Unqueue Current Record		✓	✓	✓	✓	✓	✓
ISO 18000-63 Transponder 'Inventory' Command	✓	✓	✓	✓	✓	✓	✓
Program EPC of an ISO 18000-63 Transponder	✓	✓	✓	✓	✓	✓	✓

	5226U	53x8U	53x8U-RTC	5428U	5428U-RTC	5528U	5528U-RTC
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	✓	✓	✓	✓	✓	✓	✓
'Read On-Chip RSSI' Command of an RFMicron Magnus S2 / S3	✓	✓	✓	✓	✓	✓	✓
'Read Temperature Code' Command of an RFMicron Magnus S3	✓	✓	✓	✓	✓	✓	✓
'Spontaneous' Message	✓	✓	✓	✓	✓	✓	✓