TECHNICAL INFORMATION MANUAL

Revision 19.1 – 29 April 2016

CAEN UHF RFID READERS

COMMUNICATION PROTOCOL

THE ART OF IDENTIFICATION

Scope of Manual

The goal of this manual is to provide the basic information to work with the CAEN UHF RFID READERS Communication Protocol.

Change Document Record

Date	Revision	Changes	Pages
18 Apr 2011	14	-	-
05 Oct 2012	15	Added AVP_PC parameter in the *Tab. 2.1: Attribute types*	10
		Modified *Tab. 2.3: Commands with Optional Parameters Table*	23
		Added *Tab. 2.4: Renamed Commands Table*	25
		Added bit 7 and 8 in the flag description of *InventoryTag* command	12
		Added reference to Ion R4300P Reader in the *Tab. 2.2: Command codes* and in the *Tab. 2.3: Commands with Optional Parameters Table*	22, 24
15 Oct 2013	16	Added *PowerSet* unit of measurement	10
		Modified *Set Power Command*	32
		Added R1230CB as supported reader for the *ProgramID_EPC_C1G2* command	18
		Renamed *NewRawReadIDs* in *setProtocol* in the *Set Protocol Command* and in the *Continuous Inventory Command*	27, 34
		Renamed *NewRawReadIDs* in *InventoryTag* in the *InventoryTag Command*	28
		Renamed *G2Write* in *WriteTagData_EPC_C1G2* in the *WriteTagData_EPC_C1G2 Command*	30
		Renamed *G2Read* in *ReadTagData_EPC_C1G2* in the *ReadTagData_EPC_C1G2 Command*	31
		Renamed *G2Lock* in *LockTag_EPC_C1G2* and *Tag Address* in *G2Password* in the *LockTag Command*	33
		Added *RFRegulation* in the *Tab.2.1: Attribute types*	10
		Added footnotes in the *Tab. 2.1: Attribute types*	10
		Added values for the *ResultCode* attribute	7
		Added reference to Muon A528B and qID R1240I Readers in the *Tab. 2.2: Command codes* and in the *Tab. 2.3: Commands with Optional Parameters Table*	22, 24
-	17	-	
-	18	-	
18 Jan 2016	19	Added reference to Quark Up R1270, qIDmini R1170I, ion R4301P, qID R1240IE, qID R1240IU and R1250I Tile Readers in the *Tab. 2.2: Command codes* and in the *Tab. 2.3: Commands with Optional Parameters Table*	22, 24
		Removed *Default Configuration* chapter	-
		Removed reference to obsolete readers	All pages
		Modified *Tab. 2.3: Commands with Optional Parameters Table*	24
29 Apr 2016	19.1	Added reference to qIDmini R1170INF Reader in the *Tab. 2.2: Command codes* and in the *Tab. 2.3: Commands with Optional Parameters Table*	22, 24

Reference Document

[RD1] Reader Protocol 1.0 – Working Draft Version of 25 August 2004 – Document revision 33 - EPCGlobal

[RD2] EPC Radio Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz – 960MHz – Version 1.0.9 – EPCGlobal

[RD3] ISO/IEC FDIS 18000-6:2003(E) – Information technology automatic identification and data capture techniques – Radio frequency identification for item management air interface – Part 6: Parameters for air interface communication at 860-960 MHz

CAEN RFID srl

Via Vetraia, 11 55049 Viareggio (LU) - ITALY
Tel. +39.0584.388.398 Fax +39.0584.388.959
info@caenrfid.com
www.caenrfid.com

Federal Communications Commission (FCC) Notice (Preliminary) [1]

This device was tested and found to comply with the limits set forth in Part 15 of the FCC Rules. Operation is subject to the following conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received including interference that may cause undesired operation. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment.

This device generates, uses, and can radiate radio frequency energy. If not installed and used in accordance with the instruction manual, the product may cause harmful interference to radio communications. Operation of this product in a residential area is likely to cause harmful interference, in which case, the user is required to correct the interference at their own expense. The authority to operate this product is conditioned by the requirements that no modifications be made to the equipment unless the changes or modifications are expressly approved by CAEN RFID.

[1] This declaration only applies to FCC readers R1230CB, R1260I, R1260U, R4300P, A528B, R1240I, R1270, R1170I (Mod. WR1170IUAPLP and WR1170IUHIDP), R1250I (Mod. WR1250IUXAAA, WR1250IUXAFL, WR1250IUXBAA, WR1250IUXBFL), R1240IU, R4301P.

Index

List of Tables

1 INTRODUCTION

General Information

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

The protocol is based on the Attribute Value Pair (AVP) schema and foresees a message header in order to identify the message scope.

The command set and the firmware architecture draw inspiration from the Reader Protocol 1.0 specification draft from EPCGlobal but, at now, this protocol is not fully compatible with the same last specifications.

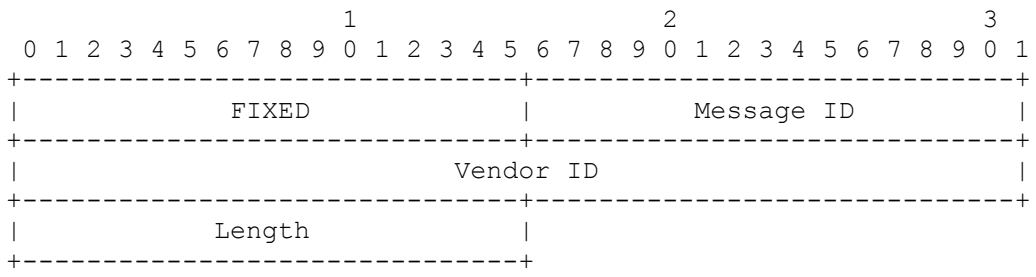
Message fields are described left to right, with the most significant byte on the left and the least on the right.

2 PROTOCOL SPECIFICATION

Introduction

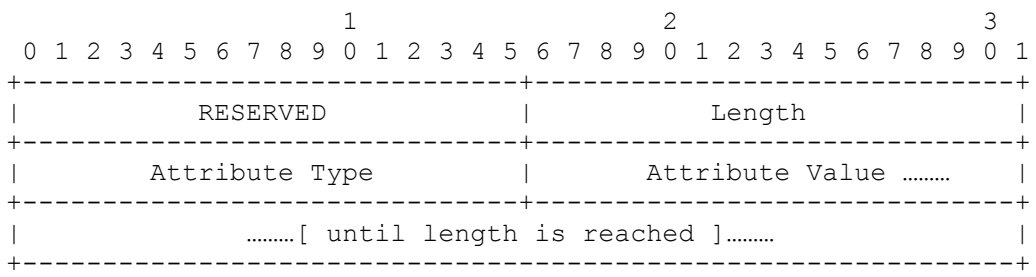
CAEN UHF RFID Reader protocol uses two logical communication channels: one for synchronous commands and one for asynchronous notifications. Command channel is mandatory and, at now, it is implemented on top of a TCP/IP socket (port 1000) and on RS232 while notification channels are implemented only with sockets.

All the messages (commands, responses and notifications) are composed by a header and a body. In all cases the body of the message is a list of attribute-value pairs. Responses always echo the Command AVP sent by the host.
All the packets for the control and notification channel share a common header format:



FIXED: Must be 0x8001 for commands and 0x0001 for responses.
Message ID: Id of the message. It is a sequence number used to map requests to its responses: a request and its corresponding response have the same message ID (the id is local to the channel).
Vendor ID: Must be 21336: the IANA "SMI Network Management Private Enterprise Code" assigned to CAEN SpA.
Length: Encodes the length of the message (in bytes) including the header.

The header is followed by a list of AVPs the number of which depends on the command. Each AVP have the following format:



RESERVED: The first 16 bits are reserved for future extensions. All reserved bits must be set to 0 on outgoing messages and ignored on incoming messages.
Length: Encodes the length of the AVP packet including the length and the reserved fields.
Attribute type: A 2 byte code identifying the attribute type.
Attribute value: The actual attribute value according to the type. It follows immediately after the Attribute Type field and runs for the remaining bytes indicated in the Length (i.e. Length minus 6 bytes of header).

Attribute types

Code	Description
0x01	**CommandName**: the command to be executed. All the commands are specified in the relevant table. Attribute value is 2 bytes long.
0x02	**ResultCode**: a code representing an indication on the result of the command. All the commands are specified in the relevant table. Attribute value is 2 bytes long. The complete list of all possible return values is: ERR_SUCCESS = 0 ERR_UNKNOWN = 102, ERR_INVALIDCMD = 127, ERR_PWROUTRANGE = 183, ERR_INVALIDPAR = 200, ERR_TAGNOTPRESENT = 202, ERR_TAGWRITE = 203, ERR_TAGBADADDRESS = 205, ERR_INVALIDFUNCTION = 206, ERR_LOCKED = 209, ERR_FAILED = 210 Note that the ERR_INVALIDPAR is used as a generic error and may be returned even if all the parameters passed to the reader are valid (for example, you may get an ERR_INVALIDPAR during the execution of a lock function just to signal that the lock operation has failed).
0x0E	**EventType**: the type of the notified event. Attribute value is 4 bytes long and can assume the following values: 0x00 = Unknown Event 0x01 = Tag glimpsed 0x02 = Tag New 0x03 = Tag Observed 0x04 = Tag Lost 0x05 = Tag Purged
0x0F	**TagIDLen**: the length of the tag ID. Attribute value is 2 bytes long.
0x10	**TimeStamp**: an indication of the time. Attribute is 8 bytes long and must be interpreted as follow: - the 4 least significant bytes are the seconds elapsed from the 1 January 1970. - the 4 most significant bytes are the micro-seconds.
0x11	**TagID**: the ID read from the tag. Attribute value has a maximum length of 64 bytes. For ISO18000 tags only the first 8 bytes are significant while for EPC tags all the 12 bytes are significant.
0x12	**TagType**: the tag's type. Attribute value is 2 bytes long and can assume the following values: 0x00 = ISO18KB 0x01 = EPCC1G1 0x02 = ISO18KA 0x03 = EPCC1G2 0x05 = EPC119
0x1E	**ChannelName**: the name of the notification channel. Attribute value has a maximum length of 30 bytes.
0x1F	**ChannelAddress**: the address of the notification channel. Attribute value has a maximum length of 30 bytes.
0x20	**TriggerName**: the name of the trigger. Attribute value has a maximum length of 30 bytes.
0x21	**TriggerType**: the type of the trigger. Attribute value has a maximum length of 30 bytes.
0x22	**ReadPointName**: a string[2] representing the name of the read point. Attribute value has a maximum length of 5 bytes and can assume the following values: "Ant0", "Ant1", "Ant2", "Ant3"
0x4D	**TagValue**: data read from the tag memory (when applicable). Attribute value has a maximum length of 128 bytes.
0x4E	**TagAddress**: the memory location address of the tag where read or write data (when applicable). Attribute value is 2 bytes long.
0x4F	**RESERVED**.
0x50	**Length**: a value representing the length of a parameter. Attribute value is 2 bytes long.

[2] Regarding the string format our convention is to use a NULL terminate string, i.e. all the string end with 0x00.

Code	Description
0x51	**BitRate**: a value representing the RF BitRate. Attribute value is 2 bytes long and can assume the following values: 0x00 – Transmit : DSB ASK 10kbit, Receive : FM0 10kbit 0x01 – Transmit : DSB ASK 10kbit, Receive : FM0 40kbit 0x02 – Transmit : DSB ASK 40kbit, Receive : FM0 40kbit 0x03 – Transmit : DSB ASK 40kbit, Receive : FM0 160kbit 0x04 – Transmit : DSB ASK 160kbit, Receive : FM0 400kbit 0x05 – Transmit : DSB ASK 40kbit, Receive : Miller M=2 160kbit 0x06 – Transmit : PR ASK 40kbit, Receive : Miller M=4 250kbit 0x07 – Transmit : PR ASK 40kbit, Receive : Miller M=4 300kbit 0x08 – Transmit : PR ASK 40kbit, Receive : Miller M=2 250kbit 0x09 – Transmit : PR ASK 40kbit, Receive : FM0 40kbit 0x0A – Transmit : DSB ASK 40kbit, Receive : Miller M=4 256kbit 0x0B – Transmit : PR ASK 40kbit, Receive : Miller M=4 320kbit 0x0C – Transmit : PR ASK 40kbit, Receive : FM0 640kbit 0x0D – Transmit : PR ASK 80kbit, Receive : Miller M=4 320kbit 0x0E – Transmit : PR ASK 40kbit, Receive : Miller M=4 256kbit Note: not all the value are supported by all the readers. For the list of mode supported by each reader please refer to the reader's user manual.
0x52	**PowerGet**: a value representing the RF power. Attribute value is 4 bytes long. (used for read the current setting)
0x53	**RESERVED**.
0x54	**Protocol**: a value representing the air protocol. Attribute value is 4 bytes long and can assume the following values: 0x00 = ISO18000-6B 0x01 = EPCC1G1 0x02 = ISO18000-6A 0x03 = EPCC1G2
0x56	**ReadPointStatus**: a value representing the antenna's status. Attribute value is 4 bytes long and can assume the following values: 0x00 = Good: antenna is well connected. 0x01 = Poor: antenna has a low quality connection. 0x02 = Bad: antenna is not connected or broken.
0x57	**Boolean**: a value representing a boolean data. Attribute value is 2 bytes long and can assume the following values: 0x00 = FALSE. Not 0x00 = TRUE.
0x58	**IPAddress**: a string[3] representing an IP address formatted with the standard IP dotted decimal format. Attribute value has a maximum length of 30 bytes.
0x59	**IPNetMask**: a string[4] representing an IP netmask formatted with the standard IP dotted decimal format. Attribute value has a maximum length of 30 bytes.
0x5A	**IPGateway**: a string[5] representing an IP address formatted with the standard IP dotted decimal format. Attribute value has a maximum length of 30 bytes.
0x5B	**DESBEnable**: used to enable/disable the Data Exchange Status Bit handling for ISO18000-6b and EPC 1.19 anti-collision algorithm. Attribute value is 2 bytes long and can assume the following values: 0x00 = Disable the DESB handling. Not 0x00 = Enable the DESB handling.
0x5C	**FWRelease**: a string[6] representing the device's firmware revision. Attribute value has a maximum length of 200 bytes.
0x5D	**DESBStatus**: used to check the Data Exchange Status Bit handling for ISO18000-6b and EPC 1.19 anti-collision algorithm. Attribute value is 2 bytes long and can assume the following values: 0x00 = DESB handling is not enabled. Not 0x00 = DESB handling is enabled.
0x5E	**EPCPWD**: a value representing an EPC tag password. Attribute value is 2 bytes long.

[3] Regarding the string format our convention is to use a NULL terminate string, i.e. all the string end with 0x00.
[4] Regarding the string format our convention is to use a NULL terminate string, i.e. all the string end with 0x00.
[5] Regarding the string format our convention is to use a NULL terminate string, i.e. all the string end with 0x00.
[6] Regarding the string format our convention is to use a NULL terminate string, i.e. all the string end with 0x00.

Code	Description
0x5F	**RFOnOff**: used to start the generation of a continuous wave for test purposes. Attribute value is 2 bytes long and can assume the following vaules: 0x00 = Stop the wave generation. Not 0x00 = Start the wave generation.
0x60	**BaudRate**: a value representing the baudrate setting of serial port. Attribute value is 4 bytes long.
0x61	**DataBits**: a value representing the databits setting of serial port. Attribute value is 4 bytes long.
0x62	**StopBits**: a value representing the stopbits setting of serial port. Attribute value is 4 bytes long.
0x63	**Parity**: a value representing the parity setting of serial port. Attribute value is 4 bytes long and can assume the following values: 0x00 = No parity 0x01 = Odd parity 0x02 = Even parity
0x64	**FlowCtrl**: a value representing the flow control setting of serial port. Attribute value is 4 bytes long and can assume the following values: 0x00 = No flow control 0x01 = Hardware flow control 0x02 = Software flow control (not yet implemented)
0x65	**DateTime**: a value representing a date and time. Attribute value has a maximum length of 30 bytes. The data format is: YYYY–MM–DD HH:MM:SS
0x66	**SelUnselOp**: a value representing the tag selection operation defined by the ISO18000-6B protocol. Attribute value is 2 bytes long and can assume the following values: 0x00 = select equal 0x01 = select not equal 0x02 = select greater than 0x03 = select lower than 0x04 = unselect equal 0x05 = unselect not equal 0x06 = unselect greater than 0x07 = unselect lower than
0x67	**Bitmask**: a value representing the flag parameter used in the newRawReadID command. Attribute value is 2 bytes long (only 8 least significant bits are used).
0x68	**REESERVED**.
0x69	**IORegister**: a value representing the status of the I/O lines of the reader. Where input lines are separated from output ones, input lines are mapped on the less significant bits while outputs are mapped on the most significant. Attribute value is 4 bytes long (effective used bits depend on the reader model).
0x6A	**ConfigParameter**: a value representing a configuration parameter. Attribute value is 4 bytes long and can assume the following values: 0x00 = ReadCycle configuration 0x01 = Observed Threshold configuation 0x02 = Lost Threshold configuration 0x03 = Starting Q value (Valid values: 0 ÷ 15). EPC C1GEN2 Protocol only. 0x04 = Session (Valid values: 0 ÷ 3). EPC C1GEN2 protocol only. 0x05 = Target (Valid values: 0 ÷ 1). EPC C1GEN2 protocol only. 0x06 = Selected (Valid values: 0, 1, 2, 3). EPC C1GEN2 protocol only. 0x07 = Data Exchange Status B (Valid values: 0 ÷ 1). ISO 18000-6B protocol only. 0x08 = Antenna dwell time during inventory (msec). A528 only. 0x09 = Inventory type (Valid values: 0 ÷ 3). A528 only.
0x6B	**ConfigValue**: a value for the configuration parameter. Attribute value is 4 bytes long.
0x6C	**NoOfTriggers**: a value representing the number of triggers. Attribute value is 2 bytes long.
0x6D	**NoOfChannels**: a value representing the number of channels. Attribute value is 2 bytes long.
0x6E	**EventMode**: a value representing the event handling mode. Attribute value is 2 bytes long and can assume the following values: 0x00 = ReadCycle mode 0x01 = Time Mode 0x02 = No Event Mode
0x6F	**UpgradeType**: a value representing the type of upgrade to perform. Attribute value is 2 bytes long and can assume the following values: 0x01 = TFTP firmware upgrade.

Code	Description
0x70	**UpgradeArgument**: a value representing the argument for the requested upgrade. Attribute value has a maximum length of 255 bytes. For TFTP upgrade (code 0x01) the string[7] has the form: '<tftpserverip> : <filename>'.
0x71	**MemoryBank**: a value representing the memory bank of a EPC Class 1 Generation 2 tag. Attribute value is 2 bytes long and can assume the following values: 0x00 = Reserved Memory Bank 0x01 = EPC Memory Bank 0x02 = TID Memory Bank 0x03 = User Memory Bank
0x72	**Payload**: a value representing the payload parameter for the EPC Class 1 Gen 2 lock command (see the EPC Gen2 specification for details). Attribute value is 4 bytes long.
0x73	**G2Password**: a value representing the Acess / Kill password parameter for the EPC Class 1 Gen 2 commands (see the EPC Gen2 specification for details). Attribute value is 4 bytes long.
0x74	**G2NSI**: a value representing the numbering system identifier for the EPC Class 1 Gen 2 tags' id (see the EPC Gen2 specification for details). Attribute value is 2 bytes long.
0x75	**QParameter**: a value representing the initial value for the Q parameter involved in the EPC Class 1 Gen 2 anticollision algorithm (see the EPC Gen2 specification for details). Attribute value is 2 bytes long.
0x76	**ReaderInfo**: a string[8] indicating the model and the serial number of the reader.
0x77	**RFRegulation**: a value representing the RF regulation to use. Attribute value is 2 bytes long and can assume the following values: 0x00 = ETSI EN 302 208 0x01 = ETSI EN 300 220 0x02 = FCC 0x03 = Malaysia 0x04 = Japan 0x05 = Korea 0x06 = Australia 0x07 = China 0x08 = Taiwan 0x09 = Singapore 0x0A = Brazil 0x0B = Japan_STD_T106 0x0C = Japan_STD_T107 Note: not all the values are supported by all the readers. For the list of RF regulation supported by each reader please refer to the reader's user manual.
0x78	**RFChannel**: a value representing the RF channel to use. Attribute value is 2 bytes long and can assume values in the range 0 … 9. Channels are referred to the ETSI EN 302 208 regulation.
0x7A	**RSSI**: a value representing the backscattered RF field strenght. Attribute value is 2 bytes long.
0x7B	**AVP_OPTION**
0x7C	**AVP_XPC** a value representing the XPC word. Attribute value is 4 bytes long.
0x7D	**AVP_PC** a value representing the PC word. Attribute value is 4 bytes long.
0x96	**PowerSet**: a value (mW) representing the RF power emitted during the communication with tags. Attribute value is 4 bytes long (used to set a new current value).
0xFB	**SourceName**: a string[9] representing the name of the data source. Attribute value has a maximum length of 30 bytes and can assume the following values: "Source_0", "Source_1", "Source_2", "Source_3"

Tab. 2.1: Attribute types

[7] Regarding the string format our convention is to use a NULL terminate string, i.e. all the string end with 0x00.

[8] The reader info string 's format is in the form <reader name> <space> <serial number>.Regarding the string format our convention is to use a NULL terminate string, i.e. all the string end with 0x00.

[9] Regarding the string format our convention is to use a NULL terminate string, i.e. all the string end with 0x00.

Command codes

Note: Some commands have been renamed to align the nomenclature in this manual and in the CAEN RFID API Reference Manual. See § *Tab. 2.4: Renamed Commands Table* pag. 25 to know the equivalence between old and new name of the renamed commands.

Note: Some commands have optional parameters. See § *Tab. 2.3: Commands with Optional Parameters Table pag. 24* to know the CAEN RFID readers that support them.

For the compliance of the command codes with the **obsolete readers**, please refer to the previous revisions of the manual that can be downloaded in the *Manuals and Documents* area of the CAEN RFID web site.

Code	Description	Comp.
0x12	**RawReadIDs**: permits to get all the tag's Ids that are under the RF field of the selected source [obsolete]. Parameters: *SourceNameIn:* [in] the name of the source to use. *SourceNameOut:* [out] the name of the source used. *ReadPointName:* [out] the name of the readpoint. *TimeStamp:* [out] the time at which the tags are detected. *TagIDLen*: [out] the ID length of the tags detected. *ListOfIDs*: [out] the list of Ids detected from the source. *ResultCode*: [out] the result code. Note: out parameters are repeated for each readpoint in the source.	**A941M**

Code	Description	Comp.
0x13	**InventoryTag**: permits to get all the tag's Ids that are under the RF field of the selected source. Parameters: *SourceNameIn*: [in] the name of the source to use (optional) *Bank:* [in] the number of the bank to use (optional). *Length*: [in] Filter Mask Length (optional). *TagID*: [in] the Filter Mask Value (optional). *TagAddress*: [in] Filter Mask Start Address (optional) *Bitmask*: [in] Inventory Flags. When set to 1 for each tag detected the RSSI value is returned. Default value 0. (Optional). Flags: (optional). *Bit0*: RSSI: a 1 value indicates the reader will transmit the RSSI (Return Signal Strength Indicator) in the response. *Bit1*: FRAMED:a 1 value indicates that the tag's data will be transmitted by the reader to the PC as soon as the tag is detected, a 0 value means that all the tags detected are buffered in the reader and trasmitted all together at the end of the inventory cycle *Bit2*: CONTINUOS: a 1 value indicates that the inventory cycle is repeated by the reader depending on the SetReadCycle setting value, a 0 value means that only one inventory cycle will be performed. If the continuous mode is selected a 0 value in the ReadCycle setting will instruct the reader to repeat the inventory cycle until an InventoryAbort method is invoked, a value X different from 0 means that the inventory cycle will be performed X times by the reader. *Bit3*: Compact data: a 1 value indicates that only the EPC of the tag will be returned by the reader, a 0 value indicates that the complete data will be returned. In case that the compact option is enabled all the other data will be populated by this library with fakes values. *Bit4*: TID reading: a 1 value indicates that also the TID of the tag will be returned by the reader together with the other information. *Bit5*: EventTrigger: a 1 value indicated that the continuous acquisition will start once an event as a key pressure occurs (this flag requires also bit2 flag active) *Bit6*: XPC: a 1 value allows the reader to get the XPC word if backscattered by a tag. Tags that do not backscatter an XPC word will return an XPC attribute with all the 4 bytes set to 0 (A528, A528B, R1230CB, R1260I, R1260E and R1260U only). *Bit7*: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*) *Bit8*: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*) For each tag detected the parameters returned by the command are: *SourceNameOut:* [out] the name of the source used *ReadPointName:* [out] the name of the readpoint. *TimeStamp:* [out] the time at which the tags are detected. *TagType*: [out] the tag's type. *TagIDLen*: [out] the ID length of the tags detected. *TagID*: [out] the tag's id. *RSSI*: [out] the tag's backscattered field strength (optional, A528, A528B, R1230CB, R1260I, R1260E, R1260U , R4300P and R1240I only). *ResultCode*: [out] the result code. Note: out parameters are repeated for each readpoint in the source. (See *§ Tab. 2.3: Commands with Optional Parameters Table pag. 24* to know the CAEN RFID readers that support them)	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x3F	**AddReadTrigger**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x40	**AddNotifyTrigger**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x41	**RemoveReadTrigger**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	

Code	Description	Comp.
0x42	**RemoveNotifyTrigger**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x49	**AllocateTrigger**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x4A	**DeallocateTrigger**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x53	**AllocateChannel**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x54	**DeallocateChannel**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x5D	**AddSourceToChannel**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x5E	**RemoveSourceFromChannel**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x5F	**AddReadPointToSource**: permits to add a readpoint to a source. Parameters: *SourceName*: [in] the name of the source. *ReadPointName*: [in] the name of the readpoint. *ResultCode*: [out] the result code.	**A528B** **R1240I** **R1240IE** **R1240IU**
0x60	**RemoveReadPointFromSource**: permits to remove a readpoint from a source. Parameters: *SourceName*: [in] the name of the source. *ReadPointName*: [in] the name of the readpoint. *ResultCode*: [out] the result code.	**A528B** **R1240I** **R1240IE** **R1240IU**
0x64	**SetPower**: permits to set the RF power level. Parameters: *PowerSet*: [in] the power level to set. *ResultCode*: [out] the result code.	**R1230CB** **R1260I** **R1260E** **R1260U** **A941M** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x6E	**ReadTagData**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x6F	**WriteTagData**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x70	**LockTag**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x71	**RESERVED**	
0x72	**SetBitRate**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	**A941M** **A528B** **R1240I** **R1240IE** **R1240IU** **R1250I**

Code	Description	Comp.
0x73	**GetPower**: permits to get the current RF power level. Parameters: *PowerGet*: [out] the current power level. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x74	**SetProtocol**: permits to set the protocol to use. Parameters: *Protocol*: [in] the protocol to use. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x75	**RESERVED**	
0x76	**CheckReadPointStatus**: permits to check the quality of the antenna connection. Parameters: *ReadPointName*: [in] the name of the readpoint. *ReadPointStatus*: [out] the quality of the connection. *ResultCode*: [out] the result code.	**A941M** **A528B** **R1240I** **R1240IE** **R1240IU** **R1250I**
0x77	**CheckSourceInChannel**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x78	**CheckReadPointInSource**: permits to verify if a readpoint belongs to a givens source. Parameters: *ReadPointName*: [in] the name of the readpoint. *SourceName*: [in] the name of the source. *Value*: [out] a Boolean value meaning the belonging to the source. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**

Code	Description	Comp.
0x79	**GetProtocol**: permits to get the protocol in use. Parameters: *Protocol*: [out] the protocol in use. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x7A	**SetNetwork**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x7B	**SetDESB**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x7C	**GetFirmwareRelease**: permits to get the firmware revision. Parameters: *FWRelease*: [in] the firmware release. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x7D	**GetDESB**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x7E	**ProgramID**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x7F	**KillTag**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x80	**RFOnOff**: permits to start/stop the generation of a continuous wave. Used only for test and measurements purposes. Parameters: *RFOnOff*: [in] = 0 ➔ stop; != 0 ➔ start *ResultCode*: [out] the result code	**R1230CB** **R1260I** **R1260U** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x81	**GetBitRate**: permits to get the BitRate in use. Parameters: *BitRate*: [out] the BitRate in. *ResultCode*: [out] the result code.	**A941M** **A528B** **R1240I** **R1240IE** **R1240IU**
0x82	**BlockWriteTag**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	

Code	Description	Comp.
0x83	**SetRS232**: permits to modify the settings of the serial port. Parameters: *Baudrate*: [in] the baud rate value. *Databits*: [in] the data bits setting. *Stopbits*: [in] the stop bits setting. *Parity*: [in] the parity setting. *Flowctrl*: [in] the flow control setting. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **R1170I** **R1170INF** **R1270** **R1250I**
0x84	**SetDateTime**: permits to modify date and time. Parameters: *Datetime*: [in] the date and time to set up. *ResultCode*: [out] the result code.	**A941M**
0x85	**GroupSelectUnselect**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x86	**GetIO**: permits to read the current status of the I/O lines. Parameters: *IORegister*: [out] the status of the I/O lines. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x87	**SetIO**: permits to set the level of the output lines. Parameters: *IORegister*: [in] the value to set to the output lines. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x88	**SetIODirection**: permits to define the direction of the I/O lines. (0 = input; 1 = output) Parameters: *IORegister*: [in] the direction to set to the I/O lines. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**

Code	Description	Comp.
0x89	**GetIODirection**: permits to read the current status of the I/O lines. (0 = input; 1 = output) Parameters: *IORegister*: [out] the direction of the I/O lines. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1270** **R1250I**
0x8A	**SetSourceConfig**: permits to set a configure parameter for a logical source. Parameters: *SourceName*: [in] the name of the source to configure. *ConfigParameter*: [in] the code of the parameter. *ConfigValue*: [in] the value for the parameter. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x8B	**GetSourceConfig**: permits to read a configure parameter for a logical source. Parameters: *SourceName*: [in] the name of the source to configure. *ConfigParameter*: [in] the code of the parameter. *ConfigValue*: [out] the value for the parameter. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x8C	**GetTriggers**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x8D	**GetChannels**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x8E	**CheckSourceInTrigger**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x8F	**CheckTriggerInChannel**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x90	**CheckChannelInTrigger**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x91	**SetEventMode**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x92	**GetEventMode**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x93	**FirmwareUpgrade**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	

Code	Description	Comp.
0x94	**E119ProgramID**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0x95	**ProgramID_EPC_C1G2**: permits to write the EPC in a Class 1 Gen 2 tag. Parameters: *SourceName*: [in] the name of the source to use. *TagIDLen*: [in] the ID length of the tag (must be an even number). *TagID*: [in] the EPC to write into the tag memory. *G2NSI*: [in] the EPC numbering system. *G2Password*: [in] the EPC Access password (optional). *ResultCode*: [out] the result code. (See *§ Tab. 2.3: Commands with Optional Parameters Table pag. 24* to know the CAEN RFID readers that support them)	**A941M** **R1230CB** **R1170I** **R1170INF** **R1270** **R1250I**
0x96	**ReadTagData_EPC_C1G2**: permits to read data from anyone of the Gen2 tag memory banks. Parameters: *SourceName*: [in, optional] the name of the source to use. *Bank:* [in] the number of the bank to use (optional). *TagAddress*: [in] Filter Mask Start Address (optional) *TagIDLen*: [in] the ID length of the tag. *TagID*: [in] the ID of the tag. *MemoryBank*: [in] the memory bank. *TagAddress*: [in] the address where to read the data. *Length*: [in] the number of bytes to read (must be an even number). *TagValue*: [out] the data read from the tag memory. *G2Password*: [in] the EPC Access password (optional). *ResultCode*: [out] the result code. (See *§ Tab. 2.3: Commands with Optional Parameters Table pag. 24* to know the CAEN RFID readers that support them).	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x97	**WriteTagData_EPC_C1G2**: permits to write data into anyone of the Gen2 tag memory banks. Parameters: *SourceName*: [in, optional] the name of the source to use. *Bank:* [in] the number of the bank to use (optional). *TagAddress*: [in] Filter Mask Start Address (optional) *TagIDLen*: [in] the ID length of the tag. *TagID*: [in] the ID of the tag. *MemoryBank*: [in] the memory bank. *TagAddress*: [in] the address where to write the data. *Length*: [in] the number of bytes to write (must be an even number). *TagValue*: [in] the data to write to the tag memory. *G2Password*: [in] the EPC Access password (optional). *ResultCode*: [out] the result code. (See *§ Tab. 2.3: Commands with Optional Parameters Table pag. 24* to know the CAEN RFID readers that support them)	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**

Code	Description	Comp.
0x98	**LockTag_EPC_C1G2**: permits to execute the tag lock command defined by the EPC Class 1 Gen 2 protocol. Parameters: *SourceName*: [in, optional] the name of the source to use. BankMask: [in] filter mask for the bank (optional). PositionMask: [in] filter mask start address (optional). *TagIDLen*: [in] the ID length of the tag to lock or the filter mask length *TagID*: [in] the ID of the tag or the filter mask to use (optional). *G2Payload*: [in] the lock payload. *G2Password*: [in] the EPC Access password (optional). *ResultCode*: [out] the result code. (See *§ Tab. 2.3: Commands with Optional Parameters Table pag. 24* to know the CAEN RFID readers that support them)	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x99	**KillTag_EPC_C1G2**: permits to execute the tag kill command defined by the EPC Class 1 Gen 2 protocol. Parameters: *SourceName*: [in, optional] the name of the source to use. BankMask: [in] filter mask for the bank (optional). PositionMask: [in] filter mask start address (optional). *TagIDLen*: [in] the ID length of the tag. *TagID*: [in] the ID of the tag or the filter mask to use (optional). *G2Password*: [in] the kill password. *ResultCode*: [out] the result code. (See *§ Tab. 2.3: Commands with Optional Parameters Table pag. 24* to know the CAEN RFID readers that support them)	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x9A	**Query_EPC_C1G2**: permits to execute the tag query command defined by the EPC Class 1 Gen 2 protocol. If a tag is in the field result code is ERROR_SUCCESS (0x00) else result code is ERROR_TAGNOTPRESENT (0xCA). Parameters: *SourceName*: [in] the name of the source to use. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x9B	**SetQ_EPC_C1G2**: permits to change the initial value of the Q parameter used in the Gen2 anticollision algorithm. Parameters: *QParameter*: [in] the value of the Q parameter. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**

Code	Description	Comp.
0x9C	**GetQ_EPC_C1G2**: permits to read the initial value of the Q parameter used in the Gen2 anticollision algorithm. Parameters: *QParameter*: [out] the value of the Q parameter. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x9D	**QueryAck_EPC_C1G2**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)*ResultCode*: [out] the result code.	
0x9E	**GetReaderInfo**: permits to read some information about the reader itself. Parameters: *ReaderInfo*: [out] a string[10] with information about the reader. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0x9F	**SetLBTMode**: permits to enable or disable the Listen Before Talk capability on ETSI EN 302 208 compatible readers. Parameters: *Boolean*: [in] 0 to disable LBT and ≠0 to enable LBT. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**

[10] Regarding the string format our convention is to use a NULL terminate string, i.e. all the string end with 0x00.

Code	Description	Comp.
0xA0	**GetLBTMode**: permits to read the current setting for the Listen Before Talk capability on ETSI EN 302 208 compatible readers. Parameters: *Boolean*: [out] 0 if LBT is disabled, ≠0 if LBT is enabled. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0xA2	**GetRFRegulation**: permits to read the RF regulation used by the reader. Parameters: *RFRegulation*: [out] the desired RF regulation. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **A528B** **R1240I** **R1240IE** **R1240IU** **R1170I** **R1170INF** **R1270** **R1250I**
0xA3	**SetRFChannel**: permits to set the RF channel where the reader emits the RF field. Parameters: *RFChannel*: [in] the RF channel. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **R1170I** **R1170INF** **R1270** **R1250I**
0xA4	**GetRFChannel**: permits to read the RF channel currently in use. Parameters: *RFChannel*: [out] the RF channel. *ResultCode*: [out] the result code.	**A941M** **R1230CB** **R1260I** **R1260U** **R1260E** **R4300P** **R4301P** **R1170I** **R1170INF** **R1270** **R1250I**
0xA7	**GetChannelData**: [Obsolete]	
0xB0	**GetBufferedData**: obsolete (*please refer to the previous revisions of the manual that can be downloaded in the Manuals and Documents area of the CAEN RFID web site*)	
0xB1	**LockBlockPermaLock_EPC_C1G2**	**A528B** **R1240I** **R1240IE** **R1240IU**
0xB2	**ReadBLockPermalock_EPC_C1G2**	**A528B** **R1240I** **R1240IE** **R1240IU**

Code	Description	Comp.
0xFFFF	**RESERVED**	

Tab. 2.2: Command codes

Commands with Optional Parameters

The following table shows a list of EPC C1G2 commands with optional parameters and the CAEN RFID readers that support them (for information about the **obsolete readers**, please refer to the previous revisions of the manual that can be downloaded in the *Manuals and Documents* area of the CAEN RFID web site).

Command \ Reader		A941EU	A528B	R1230CB	R1260I R1260U R1260E	R4300P R4301P	R1240I R1240IE R1240IU	R1270	R1170I R1170INF	R1250I
InventoryTag		√	√	√	√	√	√	√	√	√
+optional parameters	InventoryTag + SourceNameIn + bank + Length + TagID + TagAddress + Bitmask+ flag	x	√	√	√	x	x	x	x	x
	InventoryTag + SourceNameIn + bank + Length + TagID + TagAddress + Bitmask	x	√	√	√	x	x	x	x	x
	InventoryTag + SourceNameIn+ Length + TagID + TagAddress + Bitmask+ flag	√	√	√	√	√	√	√	√	√
	InventoryTag + SourceNameIn+ Length + TagID + TagAddress + Bitmask	√	√	√	√	√	√	√	√	√
KillTag_EPC_C1G2		√	√	√	√	√	√	√	√	√
+optional parameters	KillTag_EPC_C1G2 + BankMask + PositionMask + TagId	x	√	x	x	x	x	x	x	x
LockTag_EPC_C1G2		√	√	√	√	√	√	√	√	√
+optional parameters	LockTag_EPC_C1G2 + BankMask + PositionMask + TagID + G2Password	x	√	x	x	x	x	x	x	x
ReadTagData_EPC_C1G2		√	√	√	√	√	√	√	√	√
+optional parameters	ReadTagData_EPC_C1G2 + G2Password	√	√	√	√	√	√	√	√	√
	ReadTagData_EPC_C1G2 + Bank + TagAddress	x	√	x	x	x	x	x	x	x
	ReadTagData_EPC_C1G2 + Bank + TagAddress + G2Password	x	√	x	x	x	x	x	x	x
WriteTagData_EPC_C1G2		√	√	√	√	√	√	√	√	√
+optional parameters	WriteTagData_EPC_C1G2 + G2Password	√	√	√	√	√	√	√	√	√
	WriteTagData_EPC_C1G2 + Bank + TagAddress	x	√	x	x	x	x	x	x	x
	WriteTagData_EPC_C1G2 + Bank + TagAddress + G2Password	x	√	x	x	x	x	x	x	x
ProgramID_EPC_C1G2		√	√	√	√	√	√	√	√	√

Command \ Reader		A941EU	A528B	R1230CB	R1260I R1260U R1260E	R4300P R4301P	R1240I R1240IE R1240IU	R1270	R1170I R1170INF	R1250I
+optional parameters	ProgramID_EPC_C1G2 + G2Password	√	√	√	√	√	√	√	√	√

Tab. 2.3: Commands with Optional Parameters Table

Renamed Commands Table

Some commands have been renamed to align the nomenclature in this manual and in the *CAEN RFID API Reference* manual (you can download this manual from the CAEN RFID Web Site).

The following table shows the equivalence between old and new name of the renamed commands.

Old name	New name
NewRawReadIDs	InventoryTag
G2Kill	KillTag_EPC_C1G2
G2Write	WriteTagData_EPC_C1G2
G2Read	ReadTagData_EPC_C1G2
G2Lock	LockTag_EPC_C1G2
G2ProgramID	ProgramID_EPC_C1G2
G2Query	Query_EPC_C1G2
G2SetQ	SetQ_EPC_C1G2
G2GetQ	GetQ_EPC_C1G2
G2QueryAck	QueryAck_EPC_C1G2
G2ReadBlockPermalock	ReadBLockPermalock_EPC_C1G2
G2LockBlockPermablock	LockBLockPermalock_EPC_C1G2

Tab. 2.4: Renamed Commands Table

3 ASYNCHRONOUS NOTIFICATION

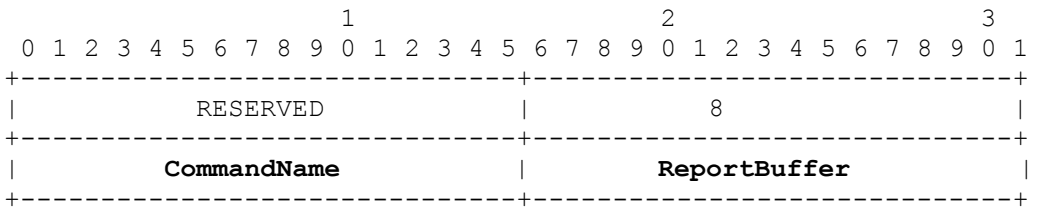
The notification channels are implemented only with sockets.

All the messages notifications are composed by a header and a body. In all cases the body of the message is a list of attribute-value pairs. The first AVP of the body is fixed and called **NotifyMessage**.

All the packets for notification channel share the same header format of other packet as described at § 3.

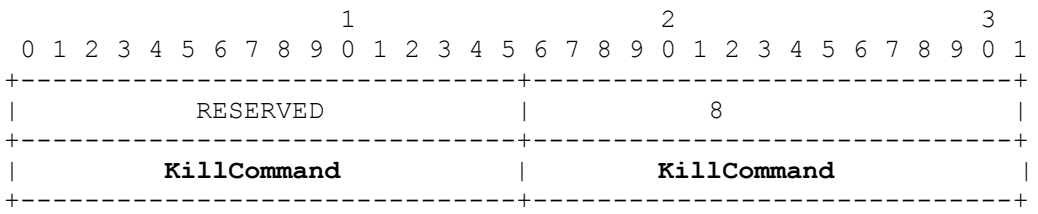
The first AVP (NotifyMessage) is followed by a list of AVPs, the number of which depends on how many tags should be notified. Each AVP has the same format of the AVP described in § 3.

The **NotifyMessage** has the following fixed format:



After the **NotifyMessage** AVP we can receive:

a) a list of AVPs (as described in table 3) followed by an AVP with 'Attribute Type' ResultCode
b) a single AVP called KillMessage with the following fixed format:



Description
TimeStamp: the timestamp of the notification TagIDLen: the ID length of the tag. TagID: the ID of the tag. SourceName: the name of the source to use. EventType: the type of the notified event

Tab. 3.3.1: Attribute types: Notification AVP List

4 EXAMPLES

Set Protocol Command

Action: Set Reader Protocol to EPC C1G2

Result: Reader selects EPC C1G2 protocol.

Command sent:

```
0x8001                  (Fixed)
0x0000                  (Message ID)
0x00005358              (Vendor ID = CAEN SpA)
0x001C                  (Message Length)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0001                  (AVP Type = CommandName)
0x0074                  (AVP Value = SetProtocol)

0x0000                  (Reserved)
0x000A                  (AVP Length)
0x0054                  (AVP Type = Protocol)
0x00000003              (AVP Value = EPC C1G2)
```

Response received:

```
0x0001                  (Fixed)
0x0000                  (Message ID)
0x00005358              (Vendor ID = CAEN SpA)
0x001A                  (Overall Message Length)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0001                  (AVP Type = CommandName)
0x0074                  (AVP Value = SetProtocol)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0002                  (AVP Type = ResultCode)
0x0000                  (AVP Value = Success)
```

InventoryTag Command

Action: Execute an inventory cycle on the logical source Source0

Result: Two EPCC1G2 tags are returned as being inventory by the reader on Ant0.
Tag1 Id = 0102030405060708091011121314151617181919 20 (160 bit)
Tag2 Id = 300833B2DDD9014035050000 (96 bit)

```
Command sent:
0x8001                      (Fixed)
0x0000                      (Message ID)
0x00005358                  (Vendor ID = CAEN SpA)
0x0021                      (Message Length)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0001                      (AVP Type = CommandName)
0x0013                      (AVP Value = InventoryTag)

0x0000                      (Reserved)
0x000F                      (AVP Length)
0x00FB                      (AVP Type = SourceName)
0x536F757263655F3000        (AVP Value = "Source_0")

Response received:
0x0001                      (Fixed)
0x0000                      (Message ID)
0x00005358                  (Vendor ID = CAEN SpA)
0x00B6                      (Overall Message Length)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0001                      (AVP Type = CommandName)
0x0013                      (AVP Value = InventoryTag)

0x0000                      (Reserved)
0x000F                      (AVP Length)
0x00FB                      (AVP Type = SourceName)
0x536F757263655F3000        (AVP Value = "Source_0")

0x0000                      (Reserved)
0x000B                      (AVP Length)
0x0022                      (AVP Type = ReadPointName)
416E743000                  (AVP Value = "Ant0")

0x0000                      (Reserved)
0x000E                      (AVP Length)
0x0010                      (AVP Type = TimeStamp)
0x00000578                  (AVP Value = Thu Jan  1 01:23:20 1970)
0x00000000                  (AVP Value)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0012                      (AVP Type = TagType)
0x0003                      (AVP Value = EPCC1G2)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x000F                      (AVP Type = TagIDLen)
0x0014                      (AVP Value = 160 bit)

0x0000                      (Reserved)
0x001A                      (AVP Length)
0x0011                      (AVP Type = TagID)
0x0102030405060708091011121314151617181919 20

0x0000                      (Reserved)
0x000F                      (AVP Length)
0x00FB                      (AVP Type = SourceName)
```

```
0x536F757263655F3000      (AVP Value)

0x0000                    (Reserved)
0x000B                    (AVP Length)
0x0022                    (AVP Type = ReadPointName)
0x416E743000              (AVP Value = "Ant0")

0x0000                    (Reserved)
0x000E                    (AVP Length)
0x0010                    (AVP Type = TimeStamp)
0x00000578                (AVP Value = Thu Jan  1 01:23:20 1970)
0x00000000                (AVP Value)

0x0000                    (Reserved)
0x0008                    (AVP Length)
0x0012                    (AVP Type = TagType)
0x0003                    (AVP Value = EPCC1G2)

0x0000                    (Reserved)
0x0008                    (AVP Length)
0x000F                    (AVP Type = TagIDLen)
0x000C                    (AVP Value = 96 bit)

0x0000                    (Reserved)
0x0012                    (AVP Length)
0x0011                    (AVP Type = TagID)
0x300833B2DDD9014035050000

0x0000                    (Reserved)
0x0008                    (AVP Length)
0x0002                    (AVP Type = ResultCode)
0x0000                    (AVP Value = Success)
```

WriteTagData_EPC_C1G2 Command

Command sent:

```
0x8001                      (Fixed)
0x001A                      (Message ID)
0x00005358                  (Vendor ID = CAEN SpA)
0x005d                      (Message Length)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0001                      (AVP Type = CommandName)
0x0097                      (AVP Value = WriteTagData_EPC_C1G2)

0x0000                      (Reserved)
0x000F                      (AVP Length)
0x00FB                      (AVP Type = SourceName)
0x536F757263655F3000        (AVP Value)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x000F                      (AVP Type = TagIDLen)
0x000C                      (AVP Value = 96 bit)

0x0000                      (Reserved)
0x0012                      (AVP Length)
0x0011                      (AVP Type = TagID)
0x300833B2DDD9014035050000


0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0071                      (AVP Type = Memory Bank)
0x0003                      (AVP Value = User Memory Bank)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x004e                      (AVP Type = Tag Address)
0x0000                      (AVP Value = Address)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0050                      (AVP Type = Length)
0x0004                      (AVP Value = # of bytes)

0x0000                      (Reserved)
0x000a                      (AVP Length)
0x004d                      (AVP Type = Tag Value)
0x00000000                  (AVP Value = bytes to be written)
```

Response received:

```
0x0001                      (Fixed)
0x001A                      (Message ID)
0x00005358                  (Vendor ID = CAEN SpA)
0x001A                      (Overall Message Length)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0001                      (AVP Type = CommandName)
0x0097                      (AVP Value = WriteTagData_EPC_C1G2)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0002                      (AVP Type = ResultCode)
0x0000                      (AVP Value = Success)
```

ReadTagData_EPC_C1G2 Command

Command sent:

```
0x8001                      (Fixed)
0x000E                      (Message ID)
0x00005358                  (Vendor ID = CAEN SpA)
0x0053                      (Message Length)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0001                      (AVP Type = CommandName)
0x0096                      (AVP Value = ReadTagData_EPC_C1G2)

0x0000                      (Reserved)
0x000F                      (AVP Length)
0x00FB                      (AVP Type = SourceName)
0x536F757263655F3000        (AVP Value)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x000F                      (AVP Type = TagIDLen)
0x000C                      (AVP Value = 96 bit)

0x0000                      (Reserved)
0x0012                      (AVP Length)
0x0011                      (AVP Type = TagID)
0x300833B2DDD9014035050000

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0071                      (AVP Type = Memory Bank)
0x0003                      (AVP Value = User Memory Bank)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x004e                      (AVP Type = Tag Address)
0x0000                      (AVP Value = Address)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0050                      (AVP Type = Length)
0x0004                      (AVP Value = # of bytes)
```

Response received:

```
0x0001                      (Fixed)
0x000E                      (Message ID)
0x00005358                  (Vendor ID = CAEN SpA)
0x0024                      (Overall Message Length)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0001                      (AVP Type = CommandName)
0x0096                      (AVP Value = ReadTagData_EPC_C1G2)

0x0000                      (Reserved)
0x000A                      (AVP Length)
0x004d                      (AVP Type = Tag Value)
0x00000000                  (AVP Value = bytes to be read)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0002                      (AVP Type = ResultCode)
0x0000                      (AVP Value = Success)
```

Set Power Command

Command sent:

```
0x8001                  (Fixed)
0x0000                  (Message ID)
0x00005358              (Vendor ID = CAEN SpA)
0x001C                  (Message Length)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0001                  (AVP Type = CommandName)
0x0064                  (AVP Value = Set Power)

0x0000                  (Reserved)
0x000A                  (AVP Length)
0x0096                  (AVP Type = Power Set)
0x000003E8              (AVP Value = 1000 mW)
```

Response received:

```
0x0001                  (Fixed)
0x0000                  (Message ID)
0x00005358              (Vendor ID = CAEN SpA)
0x001A                  (Overall Message Length)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0001                  (AVP Type = CommandName)
0x0064                  (AVP Value = Set Power)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0002                  (AVP Type = ResultCode)
0x0000                  (AVP Value = Success)
```

LockTag Command

Command sent:

```
0x8001                      (Fixed)
0x0009                      (Message ID)
0x00005358                  (Vendor ID = CAEN SpA)
0x004F                      (Message Length)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0001                      (AVP Type = CommandName)
0x0098                      (AVP Value = LockTag_EPC_C1G2)

0x0000                      (Reserved)
0x000F                      (AVP Length)
0x00FB                      (AVP Type = SourceName)
0x536F757263655F3000        (AVP Value)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x000F                      (AVP Type = TagIDLen)
0x000C                      (AVP Value = 96 bit)

0x0000                      (Reserved)
0x0012                      (AVP Length)
0x0011                      (AVP Type = TagID)
0x300833B2DDD9014035050000

0x0000                      (Reserved)
0x000A                      (AVP Length)
0x0072                      (AVP Type = Payload)
0x00000C02                  (AVP Value = User memory accessible on secure)

0x0000                      (Reserved)
0x000A                      (AVP Length)
0x0073                      (AVP Type = G2Password)
0x12345678                  (AVP Value = Password)
```

Response received:

```
0x0001                      (Fixed)
0x0009                      (Message ID)
0x00005358                  (Vendor ID = CAEN SpA)
0x001A                      (Overall Message Length)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0001                      (AVP Type = CommandName)
0x0098                      (AVP Value = LockTag_EPC_C1G2)

0x0000                      (Reserved)
0x0008                      (AVP Length)
0x0002                      (AVP Type = ResultCode)
0x0000                      (AVP Value = Success)
```

Continuous Inventory Command

Command sent (set read cycle to 0):

```
0x8001                  (Fixed)
0x0003                  (Message ID)
0x00005358              (Vendor ID = CAEN SpA)
0x0035                  (Message Length)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0001                  (AVP Type = CommandName)
0x008A                  (AVP Value = Set Source Config)

0x0000                  (Reserved)
0x000F                  (AVP Length)
0x00FB                  (AVP Type = SourceName)
0x536F757263655F3000    (AVP Value)

0x0000                  (Reserved)
0x000A                  (AVP Length)
0x006A                  (AVP Type = Config Parameter)
0x00000000              (read cycle)

0x0000                  (Reserved)
0x000A                  (AVP Length)
0x006B                  (AVP Type = Config Value)
0x00000000              (read cycle = 0)
```

Response received:

```
0x0001                  (Fixed)
0x0003                  (Message ID)
0x00005358              (Vendor ID = CAEN SpA)
0x001A                  (Overall Message Length)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0001                  (AVP Type = CommandName)
0x008A                  (AVP Value = Set Source Config)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0002                  (AVP Type = ResultCode)
0x0000                  (AVP Value = Success)
```

Command sent (Continuous inventory):

```
0x8001                  (Fixed)
0x0002                  (Message ID)
0x00005358              (Vendor ID = CAEN SpA)
0x0040                  (Message Length)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0001                  (AVP Type = CommandName)
0x0013                  (AVP Value = InventoryTag)

0x0000                  (Reserved)
0x000F                  (AVP Length)
0x00FB                  (AVP Type = SourceName)
0x536F757263655F3000    (AVP Value)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0050                  (AVP Type = Length)
0x0000                  (AVP Value = 0 byte)
```

```
0x0000                  (Reserved)
0x0007                  (AVP Length)
0x0011                  (AVP Type = TagID)
0x00

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x004e                  (AVP Type = Tag Address)
0x0000                  (AVP Value = Address)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0067                  (AVP Type = Bit Mask)
0x0006                  (AVP Value = Flags: FRAMED and CONTINUOS)
```

Response received:

```
0x0001                  (Fixed)
0x0002                  (Message ID)
0x00005358              (Vendor ID = CAEN SpA)
0x0000                  (Overall Message Length not defined)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0001                  (AVP Type = CommandName)
0x0013                  (AVP Value = InventoryTag)

0x0000                  (Reserved)
0x0008                  (AVP Length)
0x0002                  (AVP Type = ResultCode)
```

0x0000 (AVP Value = Success)

Now the reader will send a data packet every time it will detect a valid tag in the field.

To Exit from the Continuous Inventory mode the following byte shall be sent to the reader:

```
0xAB                    (Stop the continuous acquisition mode)
```

The Reader does not reply with any byte.