

Technical Information Manual

Revision n. 5
02 April 2008

**CAEN RFID UHF
READERS**
*COMMUNICATION
PROTOCOL*

NPO:
00117/03:PROTx.MUTx/05

INDEX

1. INTRODUCTION	3
1.1. CAEN RFID UHF RFID READER NAMING CONVENTIONS	3
2. PROTOCOL SPECIFICATION	4
3. ASYNCHRONOUS NOTIFICATION: PROTOCOL SPECIFICATION	28
4. DEFAULT CONFIGURATION	29
5. SAMPLES	31
6. REFERENCES	32

LIST OF TABLES

TABLE 1.1: CAEN RFID NAME ABBREVIATION	3
TABLE 2.1: ATTRIBUTE TYPES	5
TABLE 2.2: COMMAND CODES	11
TABLE 3.1: ATTRIBUTE TYPES: NOTIFICATION AVP LIST	28
TABLE 4.1: A928 AND A948 CONFIGURATION PARAMETERS DEFAULT VALUES	29
TABLE 4.2: A928 AND A948 DEFAULT COMPOSITION OF SOURCES	29
TABLE 4.3: A828, A829, A829, A946 AND A949 CONFIGURATION PARAMETERS DEFAULT VALUES	30
TABLE 4.4: A828, A829, A946 AND A949 DEFAULT COMPOSITION OF SOURCES	30

1. Introduction

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.

1.1. CAEN RFID UHF RFID Reader Naming conventions

The following table describes CAEN RFID reader name abbreviation used in the following paragraph

Table 1.1: CAEN RFID name abbreviation

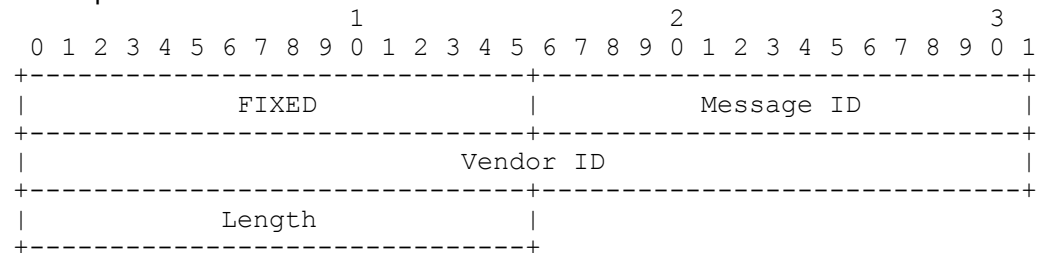
Name Abbreviation	CAEN RFID Reader
A528	A528 - OEM UHF multiregional Compact Reader
A828	A828EU - OEM UHF compact reader (ETSI EN 300 220) A828US - OEM UHF compact reader (FCC Part 15)
A828A	A828AEU - OEM UHF enhanced compact reader (ETSI EN 300 220)
A829	A829EU - Fully integrated UHF compact reader (ETSI EN 300 220) A829US - Fully integrated UHF compact reader (FCC Part 15)
A941	A941MEI - IP65 UHF stand alone reader (ETSI EN 302 208) A941MEO - OEM UHF stand alone reader (ETSI EN 302 208) A941MUI - IP65 UHF stand alone reader (FCC Part 15) A941MUO - OEM UHF stand alone reader (FCC Part 15)
A949	A949EU - IP65 UHF stand alone reader (ETSI EN 302 208)
A946	A946EU - OEM UHF stand alone reader (ETSI EN 302 208)
A928	A928EU - UHF long range reader (ETSI EN 300 220)
A948	A948EU - UHF long range reader (ETSI EN 302 208) A948EU - UHF long range reader DRM (ETSI EN 302 208) A948US - UHF long range reader (FCC part 15) A948EU - UHF long range reader HS (ETSI EN 302 208)

2. Protocol specification

CAEN RFID 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:



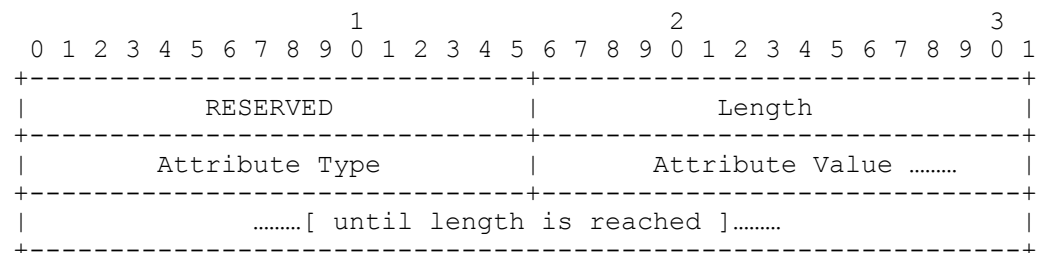
FIXED: Must be 0x8001 for commands and 0x0001 for responses.

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).

Table 2.1: 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.
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 12 bytes. For ISO18000 tags only the first 8 bytes are significant while for EPC tags all the 12 bytes are significant.
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.

Code	Description
0x22	ReadPointName: a string 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.
0x51	BitRate: a value representing the RF BitRate. Attribute value is 2 bytes long and can assume the following values: 0x0 – Transmit = 10kbit, Receive = 10kbit 0x1 – Transmit = 10kbit, Receive = 40kbit 0x2 – Transmit = 40kbit, Receive = 40kbit 0x3 – Transmit = 40kbit, Receive = 160kbit
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 = Bad: antenna is not connected or broken. 0x01 = Poor: antenna has a low quality connection. 0x02 = Good: antenna is well connected.

Code	Description
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 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 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 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 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.
0x5F	RFOOnOff: used to start the generation of a continuous wave for test purposes. Attribute value is 2 bytes long and can assume the following values: 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.

Code	Description
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 bitmask parameter of tag selection operations defined by the ISO18000-6B protocol. Attribute value is 2 bytes long (only 8 least significant bits are used).
0x68	REESERVED.

Code	Description
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 configuration 0x02 = Lost Threshold configuration
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.
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 has the form: '<tftpserverip>:<filename>'.

Code	Description
0x71	<p>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:</p> <p>0x00 = Reserved Memory Bank 0x01 = EPC Memory Bank 0x02 = TID Memory Bank 0x03 = User Memory Bank</p>
0x72	<p>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.</p>
0x73	<p>G2Password: a value representing the password parameter for the EPC Class 1 Gen 2 kill command (see the EPC Gen2 specification for details). Attribute value is 4 bytes long.</p>
0x74	<p>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.</p>
0x75	<p>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.</p>
0x76	<p>ReaderInfo: a string indicating the model and the serial number of the reader.</p>
0x77	<p>RFRegulation: a value representing the RF regulation to use. Attribute value is 2 bytes long and can assume the following values:</p> <p>0x00 = ETSI EN 302 208 0x01 = ETSI EN 300 220</p>
0x78	<p>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.</p>
0x96	<p>PowerSet: a value representing the RF power emitted during the communication with tags. Attribute value is 4 bytes long. (used to set a new current value).</p>

Code	Description
0xFB	SourceName: a string 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"

Table 2.2: Command codes

Code	Description	Comp.
0x12	RawReadIDs: permits to get all the tag's Ids that are under the RF field of the selected source. <u>Parameters:</u> <i>SourceNameIn:</i> [in] the name of the source to use. <i>SourceNameOut:</i> [out] the name of the source used. <i>ReadPointName:</i> [out] the name of the readpoint. <i>TimeStamp:</i> [out] the time at which the tags are detected. <i>TagIDLen:</i> [out] the ID length of the tags detected. <i>ListOfIDs:</i> [out] the list of Ids detected from the source. <i>ResultCode:</i> [out] the result code. Note: out parameters are repeated for each readpoint in the source.	A928 A948 A528 A828 A828A A829 A941 A946 A949
0x3F	AddReadTrigger: permits to add a trigger to a source. <u>Parameters:</u> <i>SourceName:</i> [in] the name of the source. <i>TriggerName:</i> [in] the name of the trigger. <i>ResultCode:</i> [out] the result code.	A928 A948
0x40	AddNotifyTrigger: permits to add a trigger to a notification channel. <u>Parameters:</u> <i>ChannelName:</i> [in] the name of the channel. <i>TriggerName:</i> [in] the name of the trigger. <i>ResultCode:</i> [out] the result code.	A928 A948
0x41	RemoveReadTrigger: permits to remove a trigger from a source. <u>Parameters:</u> <i>SourceName:</i> [in] the name of the source. <i>TriggerName:</i> [in] the name of the trigger. <i>ResultCode:</i> [out] the result code.	A928 A948

Code	Description	Comp.
0x42	RemoveNotifyTrigger: permits to remove a trigger from a notification channel. <u>Parameters:</u> <i>ChannelName:</i> [in] the name of the channel. <i>TriggerName:</i> [in] the name of the trigger. <i>ResultCode:</i> [out] the result code.	A928 A948
0x49	AllocateTrigger: permits to create a new trigger. <u>Parameters:</u> <i>TriggerName:</i> [in] the name of the trigger. <i>TriggerType:</i> [in] the type of the trigger. <i>ResultCode:</i> [out] the result code.	A928 A948
0x4A	DeallocateTrigger: permits to destroy an existing trigger. <u>Parameters:</u> <i>TriggerName:</i> [in] the name of the trigger. <i>ResultCode:</i> [out] the result code.	A928 A948
0x53	AllocateChannel: permits to create a notification channel. <u>Parameters:</u> <i>ChannelName:</i> [in] the name of the channel. <i>ChannelAddress:</i> [in] the address of the channel. <i>ResultCode:</i> [out] the result code.	A928 A948
0x54	DeallocateChannel: permits to destroy a notification channel. <u>Parameters:</u> <i>ChannelName:</i> [in] the name of the channel. <i>ResultCode:</i> [out] the result code.	A928 A948
0x5D	AddSourceToChannel: permits to add a source to a notification channel. <u>Parameters:</u> <i>SourceName:</i> [in] the name of the source. <i>ChannelName:</i> [in] the name of the channel. <i>ResultCode:</i> [out] the result code.	A928 A948

Code	Description	Comp.
0x5E	RemoveSourceFromChannel: permits to remove a source from a notification channel. <u>Parameters:</u> <i>SourceName:</i> [in] the name of the source. <i>ChannelName:</i> [in] the name of the channel. <i>ResultCode:</i> [out] the result code.	A928 A948
0x5F	AddReadPointToSource: permits to add a readpoint to a source. <u>Parameters:</u> <i>SourceName:</i> [in] the name of the source. <i>ReadPointName:</i> [in] the name of the readpoint. <i>ResultCode:</i> [out] the result code.	A928 A948
0x60	RemoveReadPointFromSource: permits to remove a readpoint from a source. <u>Parameters:</u> <i>SourceName:</i> [in] the name of the source. <i>ReadPointName:</i> [in] the name of the readpoint. <i>ResultCode:</i> [out] the result code.	A928 A948
0x64	SetPower: permits to set the RF power level. <u>Parameters:</u> <i>PowerSet:</i> [in] the power level to set. <i>ResultCode:</i> [out] the result code.	A928 A948 A941 A946 A949
0x6E	ReadTagData: permits to read data from the tag memory. <u>Parameters:</u> <i>SourceName:</i> [in] the name of the source to use. <i>TagIDLen:</i> [in] the ID length of the tag. <i>TagID:</i> [in] the ID of the tag. <i>TagAddress:</i> [in] the address from which read the data. <i>Length:</i> [in] the number of bytes to read. <i>TagValue:</i> [out] the data read from the tag memory. <i>ResultCode:</i> [out] the result code.	A928 A948 A528 A828 A828A A829 A941 A946 A949

Code	Description	Comp.
0x6F	WriteTagData: permits to write data to the tag memory.	A928
	<u>Parameters:</u>	A948
	<i>SourceName:</i> [in] the name of the source to use.	A528
	<i>TagIDLen:</i> [in] the ID length of the tag.	A828
	<i>TagID:</i> [in] the ID of the tag.	A828A
	<i>TagAddress:</i> [in] the address where to write the data.	A829
	<i>Length:</i> [in] the number of bytes to write.	A941
	<i>TagValue:</i> [in] the data to write to the tag memory.	A946
	<i>ResultCode:</i> [out] the result code.	A949
0x70	LockTag: permits to lock data into the tag memory.	A928
	<u>Parameters:</u>	A948
	<i>SourceName:</i> [in] the name of the source to use.	A528
	<i>TagIDLen:</i> [in] the ID length of the tag.	A828
	<i>TagID:</i> [in] the ID of the tag.	A828A
	<i>TagAddress:</i> [in] the address where to write the data.	A829
	<i>ResultCode:</i> [out] the result code.	A941
		A946
		A949
0x71	RESERVED	
0x72	SetBitRate: permits to set the BitRate to use.	A928
	<u>Parameters:</u>	A948
	<i>BitRate:</i> [in] the BitRate to set.	A528
	<i>ResultCode:</i> [out] the result code.	A828
		A828A
		A829
		A941
	A946	
	A949	
0x73	GetPower: permits to get the current RF power level.	A928
	<u>Parameters:</u>	A948
	<i>PowerGet:</i> [out] the current power level.	A941
	<i>ResultCode:</i> [out] the result code.	A946
		A949

Code	Description	Comp.
0x74	SetProtocol: permits to set the protocol to use.	A928
	<u>Parameters:</u>	A948
	<i>Protocol:</i> [in] the protocol to use.	A528
	<i>ResultCode:</i> [out] the result code.	A828
		A828A
		A829
		A941
		A946
	A949	
0x75	RESERVED	
0x76	CheckReadPointStatus: permits to check the quality of the antenna connection.	A928
	<u>Parameters:</u>	A948
	<i>ReadPointName:</i> [in] the name of the readpoint.	A941
	<i>ReadPointStatus:</i> [out] the quality of the connection.	A946
	<i>ResultCode:</i> [out] the result code.	A949
0x77	CheckSourceInChannel: permits to verify if a source is assigned to a notify channel.	
	<u>Parameters:</u>	A928
	<i>SourceName:</i> [in] the name of the source.	A948
	<i>ChannelName:</i> [in] the name of the channel.	
	<i>Value:</i> [out] a Boolean value meaning the belonging to the source.	
	<i>ResultCode:</i> [out] the result code.	
0x78	CheckReadPointInSource: permits to verify if a readpoint belongs to a givens source.	A928
	<u>Parameters:</u>	A948
	<i>ReadPointName:</i> [in] the name of the readpoint.	A941
	<i>SourceName:</i> [in] the name of the source.	A946
	<i>Value:</i> [out] a Boolean value meaning the belonging to the source.	A949
	<i>ResultCode:</i> [out] the result code.	

Code	Description	Comp.
0x79	GetProtocol: permits to get the protocol in use. <u>Parameters:</u> <i>Protocol:</i> [out] the protocol in use. <i>ResultCode:</i> [out] the result code.	A928 A948 A528 A828 A828A A829 A941 A946 A949
0x7A	SetNetwork: permits to set up the network configuration. <u>Parameters:</u> <i>IPAddress:</i> [in] the IP address to set. <i>IPNetMask:</i> [in] the IP netmask to set. <i>IPGateway:</i> [in] the IP gateway to set. <i>ResultCode:</i> [out] the result code.	A928 A948
0x7B	SetDESB: permits to enable or disable the “Data Exchange Status Bit” handling during the anti-collision algorithm when ISO 18000-6b air protocol is in use. <u>Parameters:</u> <i>DESBEnable:</i> [in] enable/disable value. <i>ResultCode:</i> [out] the result code.	A928 A948 A528 A828 A828A A829 A941 A946 A949
0x7C	GetFirmwareRelease: permits to get the firmware revision. <u>Parameters:</u> <i>FWRelease:</i> [in] the firmware release. <i>ResultCode:</i> [out] the result code.	A928 A948 A528 A828 A828A A829 A941 A946 A949

Code	Description	Comp.
0x7D	<p>GetDESB: permits to get the current setting of the “Data Exchange Status Bit” handling.</p> <p><u>Parameters:</u></p> <p><i>DESBStatus:</i> [in] enabled/disabled value.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948 A528 A828 A828A A829 A941 A946 A949
0x7E	<p>ProgramID: permits to program the ID in the EPC Class 1 Gen 1 tags.</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in] the name of the source.</p> <p><i>TagIDLen:</i> [in] the ID length of the tag.</p> <p><i>TagID:</i> [in] the ID of the tag.</p> <p><i>EPCPWD:</i> [in] the EPC password to set.</p> <p><i>Lock:</i> [in] a Boolean value; TRUE = lock the ID. FALSE = do not lock.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948
0x7F	<p>KillTag: permits to kill a EPC Class 1 Gen 1 tag.</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in] the name of the source.</p> <p><i>TagIDLen:</i> [in] the ID length of the tag.</p> <p><i>TagID:</i> [in] the ID of the tag.</p> <p><i>EPCPWD:</i> [in] the EPC password.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948
0x80	<p>RFOff: permits to start/stop the generation of a continuous wave. Used only for test and measurements purposes.</p> <p><u>Parameters:</u></p> <p><i>RFOff:</i> [in] = 0 → stop; != 0 → start</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948 A528 A828 A828A A829 A941 A946 A949

Code	Description	Comp.
0x81	<p>GetBitRate: permits to get the BitRate in use.</p> <p><u>Parameters:</u></p> <p><i>BitRate:</i> [out] the BitRate in.</p> <p><i>ResultCode:</i> [out] the result code.</p>	<p>A928</p> <p>A948</p> <p>A528</p> <p>A828</p> <p>A828A</p> <p>A829</p> <p>A941</p> <p>A946</p> <p>A949</p>
0x82	<p>BlockWriteTag: permits to write data to the tag memory. This function uses the ISO18000-6b Write4Byte command to speed up the writing of large amount of data at one time.</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in] the name of the source to use.</p> <p><i>TagIDLen:</i> [in] the ID length of the tag.</p> <p><i>TagID:</i> [in] the ID of the tag.</p> <p><i>TagAddress:</i> [in] the address where to write the data.</p> <p><i>Length:</i> [in] the number of bytes to write.</p> <p><i>TagValue:</i> [in] the data to write to the tag memory.</p> <p><i>ResultCode:</i> [out] the result code.</p>	<p>A928</p> <p>A948</p> <p>A528</p> <p>A828</p> <p>A828A</p> <p>A829</p> <p>A941</p> <p>A946</p> <p>A949</p>
0x83	<p>SetRS232: permits to modify the settings of the serial port.</p> <p><u>Parameters:</u></p> <p><i>Baudrate:</i> [in] the baud rate value.</p> <p><i>Databits:</i> [in] the data bits setting.</p> <p><i>Stopbits:</i> [in] the stop bits setting.</p> <p><i>Parity:</i> [in] the parity setting.</p> <p><i>Flowctrl:</i> [in] the flow control setting.</p> <p><i>ResultCode:</i> [out] the result code.</p>	<p>A928</p> <p>A948</p> <p>A528</p> <p>A828</p> <p>A828A</p> <p>A829</p> <p>A941</p> <p>A946</p> <p>A949</p>

Code	Description	Comp.
0x84	SetDateTime: permits to modify date and time.	A928
	<u>Parameters:</u>	A948
	<i>Datetime:</i> [in] the date and time to set up.	A528
	<i>ResultCode:</i> [out] the result code.	A828
		A828A
		A829
		A941
		A946
0x85	GroupSelectUnselect: permits to execute the tag selection commands defined by the ISO18000-6B protocol.	A928
	<u>Parameters:</u>	A948
	<i>SourceName:</i> [in, optional] the name of the source to use.	A528
	<i>Operation:</i> [in] the tag selection operation.	A828
	<i>Bytemask:</i> [in] the byte mask as defined by the protocol.	A828A
	<i>TagAddress:</i> [in] the address where to compare the data.	A829
	<i>TagValue:</i> [in] the data to compare with the tag memory.	A941
	<i>TagID:</i> [out] the ID of the tag.	A946
0x86	GetIO: permits to read the current status of the I/O lines.	A928
	<u>Parameters:</u>	A948
	<i>IORegister:</i> [out] the status of the I/O lines.	A528
	<i>ResultCode:</i> [out] the result code.	A828
		A828A
		A829
		A941
		A946
	A949	

Code	Description	Comp.
0x87	SetIO: permits to set the level of the output lines. <u>Parameters:</u> <i>IORegister:</i> [in] the value to set to the output lines. <i>ResultCode:</i> [out] the result code.	A928 A948 A528 A828 A828A A829 A941 A946 A949
0x88	SetIODirection: permits to define the direction of the I/O lines. (0 = input; 1 = output) <u>Parameters:</u> <i>IORegister:</i> [in] the direction to set to the I/O lines. <i>ResultCode:</i> [out] the result code.	A528 A828 A828A A829 A941 A946 A949
0x89	GetIODirection: permits to read the current status of the I/O lines. (0 = input; 1 = output) <u>Parameters:</u> <i>IORegister:</i> [out] the direction of the I/O lines. <i>ResultCode:</i> [out] the result code.	A528 A828 A828A A829 A941 A946 A949
0x8A	SetSourceConfig: permits to set a configure parameter for a logical source. <u>Parameters:</u> <i>SourceName:</i> [in] the name of the source to configure. <i>ConfigParameter:</i> [in] the code of the parameter. <i>ConfigValue:</i> [in] the value for the parameter. <i>ResultCode:</i> [out] the result code.	A928 A948

Code	Description	Comp.
0x8B	<p>GetSourceConfig: permits to read a configure parameter for a logical source.</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in] the name of the source to configure.</p> <p><i>ConfigParameter:</i> [in] the code of the parameter.</p> <p><i>ConfigValue:</i> [out] the value for the parameter.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948
0x8C	<p>GetTriggers: permits to read the names of the allocated triggers.</p> <p><u>Parameters:</u></p> <p><i>NoOfTriggers:</i> [out] the number of allocated triggers.</p> <p><i>ListOfTriggerNames:</i> [in] a list containing the names of the allocated triggers.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948
0x8D	<p>GetChannels: permits to read the names of the allocated notification channels.</p> <p><u>Parameters:</u></p> <p><i>NoOfChannels:</i> [out] the number of allocated channels.</p> <p><i>ListOfChannelNames:</i> [in] a list containing the names of the allocated channels.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948
0x8E	<p>CheckSourceInTrigger: permits to verify if the specified logical source is associated to the specified trigger.</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in] the name of the source.</p> <p><i>TriggerName:</i> [in] the name of the trigger.</p> <p><i>Boolean:</i> [out] 0 if they are associated, ≠0 if not.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948
0x8F	<p>CheckTriggerInChannel: permits to verify if the specified trigger is associated to the specified channel.</p> <p><u>Parameters:</u></p> <p><i>ChannelName:</i> [in] the name of the source.</p> <p><i>TriggerName:</i> [in] the name of the trigger.</p> <p><i>Boolean:</i> [out] 0 if they are associated, ≠0 if not.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948

Code	Description	Comp.
0x90	<p>CheckChannelInTrigger: permits to verify if the specified channel is associated to the specified trigger.</p> <p><u>Parameters:</u></p> <p><i>ChannelName:</i> [in] the name of the source.</p> <p><i>TriggerName:</i> [in] the name of the trigger.</p> <p><i>Boolean:</i> [out] 0 if they are associated, ≠0 if not.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948
0x91	<p>SetEventMode: permits to set the event generation mode for the reader notification channels.</p> <p><u>Parameters:</u></p> <p><i>EventMode:</i> [in] the event mode.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948
0x92	<p>GetEventMode: permits to read the event generation mode for the reader notification channels.</p> <p><u>Parameters:</u></p> <p><i>EventMode:</i> [out] the event mode.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948
0x93	<p>FirmwareUpgrade: permits to upgrade the reader firmware.</p> <p><u>Parameters:</u></p> <p><i>UpgradeType:</i> [in] the type of the upgrade.</p> <p><i>UpgradeArg:</i> [in] the argument needed by the upgrade procedure.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948
0x94	<p>E119ProgramID: permits to write the EPC into a EPC1.19 tag.</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in] the name of the source to use.</p> <p><i>TagID:</i> [in] the old EPC of the tag.</p> <p><i>TagValue:</i> [in] the EPC to write into the tag memory.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948 A528 A828 A828A A829 A941 A946 A949

Code	Description	Comp.
0x95	<p>G2ProgramID: permits to write the EPC in a Class 1 Gen 2 tag.</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in] the name of the source to use.</p> <p><i>TagIDLen:</i> [in] the ID length of the tag (must be an even number).</p> <p><i>TagID:</i> [in] the EPC to write into the tag memory.</p> <p><i>G2NSI:</i> [in] the EPC numbering system.</p> <p><i>ResultCode:</i> [out] the result code.</p>	<p>A928</p> <p>A948</p> <p>A528</p> <p>A828</p> <p>A828A</p> <p>A829</p> <p>A941</p> <p>A946</p> <p>A949</p>
0x96	<p>G2Read: permits to read data from anyone of the Gen2 tag memory banks.</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in, optional] the name of the source to use.</p> <p><i>TagIDLen:</i> [in] the ID length of the tag.</p> <p><i>TagID:</i> [in] the ID of the tag.</p> <p><i>MemoryBank:</i> [in] the memory bank.</p> <p><i>TagAddress:</i> [in] the address where to read the data.</p> <p><i>Length:</i> [in] the number of bytes to read (must be an even number).</p> <p><i>TagValue:</i> [out] the data read from the tag memory.</p> <p><i>ResultCode:</i> [out] the result code.</p>	<p>A928</p> <p>A948</p> <p>A528</p> <p>A828</p> <p>A828A</p> <p>A829</p> <p>A941</p> <p>A946</p> <p>A949</p>
0x97	<p>G2Write: permits to write data into anyone of the Gen2 tag memory banks.</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in, optional] the name of the source to use.</p> <p><i>TagIDLen:</i> [in] the ID length of the tag.</p> <p><i>TagID:</i> [in] the ID of the tag.</p> <p><i>MemoryBank:</i> [in] the memory bank.</p> <p><i>TagAddress:</i> [in] the address where to write the data.</p> <p><i>Length:</i> [in] the number of bytes to write (must be an even number).</p> <p><i>TagValue:</i> [in] the data to write to the tag memory.</p> <p><i>ResultCode:</i> [out] the result code.</p>	<p>A928</p> <p>A948</p> <p>A528</p> <p>A828</p> <p>A828A</p> <p>A829</p> <p>A941</p> <p>A946</p> <p>A949</p>

Code	Description	Comp.
0x98	<p>G2Lock: permits to execute the tag lock command defined by the EPC Class 1 Gen 2 protocol.</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in, optional] the name of the source to use.</p> <p><i>TagIDLen:</i> [in] the ID length of the tag.</p> <p><i>TagID:</i> [in] the ID of the tag.</p> <p><i>G2Payload:</i> [in] the lock payload.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948 A528 A828 A828A A829 A941 A946 A949
0x99	<p>G2Kill: permits to execute the tag kill command defined by the EPC Class 1 Gen 2 protocol.</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in, optional] the name of the source to use.</p> <p><i>TagIDLen:</i> [in] the ID length of the tag.</p> <p><i>TagID:</i> [in] the ID of the tag.</p> <p><i>G2Password:</i> [in] the kill password.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948 A528 A828 A828A A829 A941 A946 A949
0x9A	<p>G2Query: 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).</p> <p><u>Parameters:</u></p> <p><i>SourceName:</i> [in] the name of the source to use.</p> <p><i>ResultCode:</i> [out] the result code.</p>	A928 A948 A528 A828 A828A A829 A941 A946 A949

Code	Description	Comp.
0x9B	G2SetQ: permits to change the initial value of the Q parameter used in the Gen2 anticollision algorithm. <u>Parameters:</u> <i>QParameter:</i> [in] the value of the Q parameter. <i>ResultCode:</i> [out] the result code.	A928 A948 A528 A828 A828A A829 A941 A946 A949
0x9C	G2GetQ: permits to read the initial value of the Q parameter used in the Gen2 anticollision algorithm. <u>Parameters:</u> <i>QParameter:</i> [out] the value of the Q parameter. <i>ResultCode:</i> [out] the result code.	A928 A948 A528 A828 A828A A829 A941 A946 A949
0x9D	G2QueryAck: permits to execute the tag query and ack command defined by the EPC Class 1 Gen 2 protocol. If a tag is in the field result code is ERROR_SUCCESS (0x00) and the command returns the EPC code stored in the tag else the result code is ERROR_TAGNOTPRESENT (0xCA). <u>Parameters:</u> <i>SourceName:</i> [in] the name of the source to use. <i>TagID:</i> [out] the ID of the tag. <i>ResultCode:</i> [out] the result code.	A928 A948 A528 A828 A828A A829 A941 A946 A949

Code	Description	Comp.
0x9E	GetReaderInfo: permits to read some information about the reader itself. <u>Parameters:</u> <i>ReaderInfo:</i> [out] a string with information about the reader. <i>ResultCode:</i> [out] the result code.	A928 A948 A528 A828 A828A A829 A941 A946 A949
0x9F	SetLBTMode: permits to enable or disable the Listen Before Talk capability on ETSI EN 302 208 compatible readers. <u>Parameters:</u> <i>Boolean:</i> [in] 0 to disable LBT and ≠0 to enable LBT. <i>ResultCode:</i> [out] the result code.	A941 A946 A948 A949
0xA0	GetLBTMode: permits to read the current setting for the Listen Before Talk capability on ETSI EN 302 208 compatible readers. <u>Parameters:</u> <i>Boolean:</i> [out] 0 if LBT is disabled, ≠0 if LBT is enabled. <i>ResultCode:</i> [out] the result code.	A941 A946 A948 A949
0xA1	SetRFRegulation: permits to change the RF regulation used by the reader. <u>Parameters:</u> <i>RFRegulation:</i> [in] the desired RF regulation. <i>ResultCode:</i> [out] the result code.	A941 A946 A948 A949
0xA2	GetRFRegulation: permits to read the RF regulation used by the reader. <u>Parameters:</u> <i>RFRegulation:</i> [out] the desired RF regulation. <i>ResultCode:</i> [out] the result code.	A941 A946 A948 A949
0xA3	SetRFChannel: permits to set the RF channel where the reader emits the RF field. <u>Parameters:</u> <i>RFChannel:</i> [in] the RF channel. <i>ResultCode:</i> [out] the result code.	A941 A946 A948 A949

Code	Description	Comp.
0xA4	GetRFChannel: permits to read the RF channel currently in use.	A941
	<u>Parameters:</u>	A946
	<i>RFChannel:</i> [out] the RF channel.	A948
	<i>ResultCode:</i> [out] the result code.	A949
0xFFFF	RESERVED	

3. Asynchronous Notification: Protocol specification

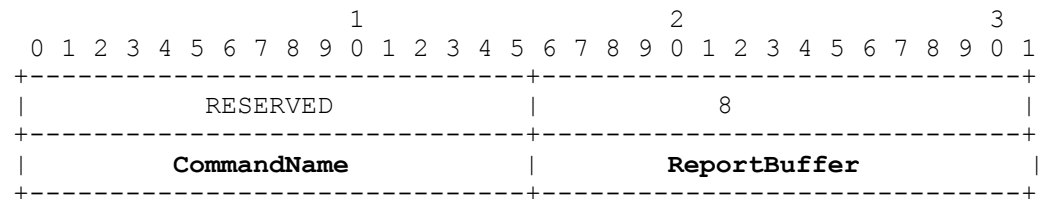
The notification channels are implemented only with sockets.

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:

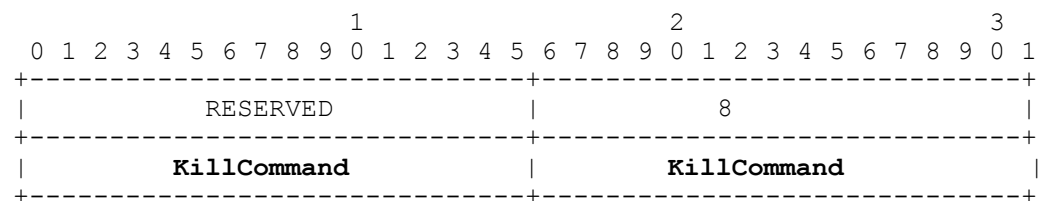


Table 3.1: Attribute types: Notification AVP List.

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

4. Default configuration

CAEN A928 and A948E UHF RFID Reader protocol has various configuration parameters; in the following table are summarized the default values.

Table 4.1: A928 and A948 Configuration parameters default values

Parameter	Default value
IP Address	192.168.0.125
IP Netmask	255.255.255.0
IP Gateway	192.168.0.1
Sources	"Source_0", "Source_1", "Source_2", "Source_3"
Readpoints	"Ant0", "Ant1", "Ant2", "Ant3"
Baud Rate	115200
Data Bits	8
Stop Bits	1
Parity	None
Flow Control	None

The default composition of sources for A928E and A948 reader is the following:

Table 4.2: A928 and A948 Default composition of sources

Source	Readpoints
Source_0	Ant0
Source_1	Ant1
Source_2	Ant2
Source_3	Ant3

CAEN A828, A829, A946 and A949 UHF RFID Reader protocol has various configuration parameters; in the following table are summarized the default values.

Table 4.3: A828, A829, A829, A946 and A949 Configuration parameters default values

Parameter	Default value
Sources	"Source_0"
Readpoints	"Ant0"
Baud Rate	115200
Data Bits	8
Stop Bits	1
Parity	None
Flow Control	None

The default composition of sources for the A828, A829, A946 and A949 reader is the following:

Table 4.4: A828, A829, A946 and A949 Default composition of sources

Source	Readpoints
Source_0	Ant0

5. Samples

In the following sample a RawReadIDs command is examined together with the response coming from the reader.

Command sent:

```

0x8001          (Fixed)
0x0000          (Message ID)
0x00005358     (Vendor ID = CAEN SpA)
0x0021         (Message Length)
0x0000         (Reserved)
0x0008         (AVP Length)
0x0001         (AVP Type = CommandName)
0x0012         (AVP Value = RawReadIDs)
0x0000         (Reserved)
0x000F         (AVP Length)
0x00FB         (AVP Type = SourceName)
"Source_0"     (AVP Value)
  
```

Response received:

```

0x0001          (Fixed)
0x0000          (Message ID)
0x00005358     (Vendor ID = CAEN SpA)
0x0066         (Message Length)
0x0000         (Reserved)
0x0008         (AVP Length)
0x0001         (AVP Type = CommandName)
0x0012         (AVP Value = RawReadIDs)
0x0000         (Reserved)
0x000F         (AVP Length)
0x00FB         (AVP Type = SourceName)
"Source_0"     (AVP Value)
0x0000         (Reserved)
0x000B         (AVP Length)
0x0022         (AVP Type = ReadPointName)
"Ant0"         (AVP Value)
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)
0x000F         (AVP Type = TagIDLen)
0x0008         (AVP Value)
0x0000         (Reserved)
0x000E         (AVP Length)
0x0011         (AVP Type = TagID)
0x05629FFF    (AVP Value)
0xC0113001    (AVP Value)

0x0000         (Reserved)
0x000E         (AVP Length)
0x0011         (AVP Type = TagID)
0xE0040F0E    (AVP Value)
0x06010000    (AVP Value)
0x0000         (Reserved)
0x0008         (AVP Length)
0x0002         (AVP Type = ResultCode)
0x0000         (AVP Value = Success)
  
```

6. References

- Reader Protocol 1.0 – Working Draft Version of 25 August 2004 – Document revision 33 - EPCGlobal
- EPC Radio Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz – 960MHz – Version 1.0.9 – EPCGlobal
- 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