Ez – Alert Wired Interface Command Reference

Ez Alert Technology Series

Version 3.0.1

Table of Contents

Overview

This document describes the command interface and organization used in the Ez-Alert product line for those devices using a hard wired interface. For Ez-Alert devices using the XEL wireless interface, please refer to the *Ez-Alert Wireless Interface Command Reference*.

Ez-Alert Overview

Ez-Alert is a family of products available from XscapeEz Ltd which monitor the vehicle data-links to report the status and health of a vehicle. It is available in multiple packages ranging from chipset solutions to fully enclosed, stand alone devices with either wired or wireless interfaces.

In the wired devices, the serial communications configuration used is as follows:
Baud Rate – 115200, 8 start bits, 2 stop bits, no parity. Make sure the host is et to two stop its or undesirable operation can occur.

Ez-Alert solutions provide vehicle status through monitoring of active and historic J1939 SPNs and J1708 fault codes, user-defined parameter reporting, transmit requests for non-broadcast parameters, user-defined trigger events, and user-defined message transmission. Table 1-1 details the different Ez-Alert Monitoring features available to the programmer.

Monitor Type	Description
Fault Code	J1708 and J1939 Active and historic fault codes. Can be filtered as well.
Triggers	Triggers are event monitors that watch for a user-defined parameter to go above or below a set value. The trigger will report these occurrences and their duration.
Parameters	Both broadcast and requested J1708and J1939 parameters can be recorded and reported to the host system.

Table 1-1: Types of Monitoring available in Ez-Alert

The system is designed to give the user control over the system timing, data content, and data size. A typical application might use the device in the following manner:

1. Call the *Get Version* command to verify communications with the module.
2. Call the *Set Calibration* command to set the J1708 MID and the J1939 Source address of the device as well as the 16-bit Configuration Flag which controls how data is collected and what data is reported to the host.
3. Call any of the Write Table commands to configure triggers, fault monitoring, parameter monitoring, parameter requests, and transmit messages for the application.
4. Call the *Read Status* command at periodic intervals to return collected data and events from the Ez-Alert module. Note that this command optionally takes the same 16-bit flag as *Set Calibration* and, when used with *Read Status*, will override the default setting and only collect and return data as specified for one and only one call to this command. This can be very useful when only a subset of the configured data is needed for a particular operation and then the system will automatically reset to the “Calibrated” mode of operation.

Ez-Alert Command Set Organization

Ez-Alert Commands are those provided to configure the device, monitor vehicle status including faults and triggers while the vehicle is operating, perform file transfers, and more.

Command Formats

The general format is defined in Table 1-2 below. Each command is defined in its own section with subtle differences, including whether or not they illicit a response from the device. The programmer is encouraged to review each command to understand its parameters, effects, and responses.

Ez-Alert Command Format

Header **Footer**

# of bytes	**1**	**2**	**1**	**1**	**1-N**	**1**
Byte Definition	Header Byte	Packet Count	Protocol	Command	Data	Checksum

Table 1-2: Command Format

Payload

Header Byte – Used for baud rate detect and error checking on each packet This byte is always set to '0x7E'.

Packet Count– The command packet's byte count. It represents the total number of bytes in a packet from the protocol byte to the end of data. It does not include the header byte, two packet count bytes, or the checksum. (First byte MSB, Second Bytes LSB)

Protocol – Protocol for which this packet is encoded

Protocol Enumerations
0x00 –Device Operation
0x01 – J1708
0x02 – J1939

Table 1-3: Protocol Numbers

Command– The command byte defines which actions should be taken by the module.

Data–. When the data is variable length, the first byte of the data packet may be a single byte, data byte count. Refer to the each command definition for specifics. Byte count is defined as the number of subsequent bytes following the byte count. This will be shown in the individual command description when applicable.

Checksum– The command packet's 1's complement checksum excluding the header byte and two byte packet count.

Device Protocol Command Reference

These commands are for support of the device specific functions inside the Ez-Alert module. All status monitoring and configuration commands use the *Device* protocol (0x00) for wired interface implementations. All commands and examples described herein show the data payload only and assume the use of the header and footer shown in Table 1-2

It should be noted that all 16 and 32-bit values are in big endian format, or sent down MSB first as shown in each command section unless otherwise noted. All information taken from the data links is left in data link defined data formats. The examples will show this format as well as the governing specifications.

All command numbers are sent down exactly as shown; however, all responses use the same command number with the upper most bit set to differentiate a request from a response. As an example, a command 0x41 comes back as a 0xC1 (0x41 with bit 7 set) in a response. See individual examples for more information.

Read Status

Command used to retrieve active fault, active trigger status, and vehicle monitoring information. This command elicits multiple responses. It is the fundamental command used to gather health information about the vehicle.

This command carries an optional configuration flag parameter. This parameter is a 16-bit value that overrides the current calibration setting for what data will be reported and how that data will be collected. If not used, the 16-bit flag set using the *'Set Calibration'* command, defined in this document, controls how data is collected and reported. The bit definition of the flag can be found in this command definition as well as the Set Calibration definition. A detailed description of the effect of each bit is shown in Table 1-4.

Bit Name	Bit Position	Bit Function
Reserved	0	Internal Use, set to 0
Clear Triggers	1	Ez-Alert will automatically clear trigger data after reporting
Report Triggers	2	Ez-Alert send the user defined trigger data to the host on a status request
Reserved	3	Internal Use set to 0
Reserved	4	Internal Use set to 0
Clear Parameters	5	Ez-Alert will automatically clear them after reporting
Report Parameters	6	Ez-Alert send the user defined parameters to the host on a status request
Send TX requests	7	Ez-Alert will transmit the data link request message for the preset parameters
Reserved	8	Internal Use, set to 0
Clear Faults	9	Ez-Alert will automatically clear fault codes in the table after reporting
Report Table Faults	10	Ez-Alert send the faults recorded in the fault table to the host on a status request
Send Request for Faults	11	Ez-Alert will transmit a request for all nodes to report their active faults before sending up the reported fault list
Reserved	12	Internal Use set to 0
Send User Transmits	13	Ez-Alert will transmit the user defined data link messages
Filter Faults	14	Ez-Alert will apply the users fault filter to the active received faults Note that recorded faults are filtered by how the user sets up recorded fault table rules.
Report Current Faults	15	User would use this bit with bit 11 to generate a data link request for current faults and then this bit will cause the active faults, reported through the request, up to the host. The system will report all active faults received in a one second window after receipt of the Status Request message. Note that this is independent of collected faults in the table although the same fault filter applies.

Table 1-4: Bit definition and effects of configuration Flag on data collection and reporting

As mentioned above, this command can generate several responses based on the setting of the 16-bit *Configuration Flag*. The first response is a series of trigger status messages describing if triggers matching the user's configuration have been detected. There is one message per active trigger as described in the response section below.

The remaining different types of responses come back with a 'Table Response' command number and the following byte indicating which type of table data is contained. Note that if the upper bit of the Table Identifier is set the response is reported (live) data, if it is clear the response is recorded data. Table 1-5 shows the format of all table responses.

Sequence of Read Status events:

Send Fault requests on data links	(Bit 11)
Send Parameter Requests on data links	(Bit 7)
Send User defined messages on data links	(Bit 13)
Report Trigger Data	(Bit 2)

Report Recorded Faults	(Bit 10)
Report current faults	(Bits 15)
Clear Fault Data Tables	(Bit 9)
Clear Trigger Data Tables	(Bit 1)

Protocol	Table Response Command	Table Identifier	Protocol for table data	Table Data
0x00	0x44	0x01 – Recorded Fault Table 0x02 – Recorded Parameter Table 0x04 – Trigger Table 0x11 – Reported Fault Table	0x01 – J1708 0x02 – J1939	Defined per table type below

Table 1-5: Table data response format

Note that the fault message format for currently active, reported faults is defined by the respective SAE® specification.

Example Status Request

Request Format: [Protocol] [Command] [Configuration Flag]
Protocol: 0x00
Command: 0x00
Optional Configuration Flag 16-bit value - see Table 1-4 above

Example: Request Global Status using the 16-bit Config Flag value from *Set Calibration*
[Header] [0x00] [0x00] [Footer]

Response Type 0- Status Message – Active Device and Status Information
This message is always present regardless of the 16-bit configuration flag setting. All parameters indicating vehicle module activity use the following bit definition.

Bit 0 - J1708 Engine
Bit 1 - J1708 Transmission
Bit 2 - J1708 Brakes
Bit 3 - Not Used
Bit 4 - J1939 Engine
Bit 5 - J1939 Transmission
Bit 6 - J1939 Brakes
Bit 7 - Unused

Protocol: 0x00
Command: 0x80
Devices active on link - Conveys what devices are active on the data links
Devices reporting inactive fault(s) Conveys what devices are reporting inactive faults
Devices reporting active fault(s) Conveys what devices are reporting active faults
Active triggers - Any non-zero value indicates that at least one trigger is active and should be read

Response Type 1- Reported Active Triggers
Response Format: Table Response per Table 1-5
Protocol: 0x00
Command: 0x44
Table Type: per Table 1-5
Trigger Data Identical to the *Read Trigger Data* command response per page 31

Response Type 2- Current Reported Active J1708 Faults
Response Format: Table Response per Table 1-5
Protocol: 0x00
Command: 0x44
Table Type: per Table 1-5
Data List of single byte Fault Codes

Response Type 3- Current Reported Active J1939 Faults (SPNs)
Response Format: Table Response per Table 1-5
Protocol: 0x00
Command: 0x44
Table Type: per Table 1-5
Data List of four byte Suspect Parameter Numbers

Response Type 4- Recorded J1708 Faults from the fault table
Response Format: Table Response per Table 1-5
Protocol: 0x00
Command: 0x44
Table Type: per Table 1-5
Data List of single byte Fault Codes

Response Type 5- Recorded J1939 Faults from the fault table (SPNs)
Response Format: Table Response per Table 1-5
Protocol: 0x00
Command: 0x44
Table Type: per Table 1-5
Data List of four byte Suspect Parameter Numbers

Response Type 6- Recorded Parameters from the J1939 user defined parameter table
Response Format: Table Response per Table 1-5
Protocol: 0x00
Command: 0x44
Table Type: per Table 1-5
MSB PGN: Most significant byte of the16-bit PGN
LSB PGN: Least significant byte of the16-bit PGN
Byte Count: Number of data bytes that follow
Data: Associated data

Response Type 7- Recorded Parameters from the J1708 user defined parameter table
Response Format: Table Response per Table 1-5
Protocol: 0x00
Command: 0x44

Table Type:	per Table 1-5
MIDn:	J1708 MID associated with this reported parameter
PIDn:	J1708 PID for this parameter
Datan:	Data associated with this parameter, Note the data size is controlled by the user setting in the parameter rules table

Get Version

Used to read version information from the Ez-Alert module. This command returns five response messages containing version numbers for the overall system firmware and hardware installed in the system as well as the individual component modules that make up the system. These responses are described in the order they are received below.

Protocol(s): 0x00
Command: 0x1B
Type: Fixed Format, Fixed Length
Format:
[Protocol] [Command]

Response 1:
Command: 0x1B - General Version Number
[Protocol] [Command][XY][AB][CD][EF]

[X].[Y].[A].[B] System hardware version. There is one decimal value per nibble
[C].[D].[E].[F] Firmware version. There is one decimal value per nibble

Example: Get Version information from active device
[Header] [0x00] [0x1B] [Footer]

Response 1: System Hardware Version 1.6.1.0 Firmware Version 2.6.4.0
[Header] [0x00] [0x9B] [0x16] [0x10] [0x26] [0x40] [Footer]

Read Calibration

Used to read Ez-Alert identification and configuration information.

Protocol(s): 0x00
Command: 0x12
Type: Fixed Format, Variable length
Format: [Protocol][Command]

Response:
Type: Fixed Format, Variable Length
[MID] J1708 Source MID for the tool
[SA] J1939 Source Address for the tool

Response Format:
[Protocol] [Command] [MID] [SA] [Configuration Flag]

Example: Request info from Ez-Alert module.
[Header] [0x00] [0x12] [Footer]

Response: J1708 MID =0xAC and SA = 0x0D
[Header] [0x00] [0x92] [0xAC] [0x0D] [0x3C] [0xCE] [Footer]

Set Calibration

This command is used to allow users to set configuration and power up parameters in the Ez-Alert module

Protocol: 0x00
Command: 0x13
Format: [Protocol] [Command] [MID] [SA] [Configuration Flag]

[MID] Source J1708 MID for the tool
[SA] Source Address for J1939 for the tool
[Configuration Flag] 16-bit configuration flag defined in Table 1-4.
Example: a J1708 MID of 0xF0 and a J1939 SA of 0xC4 with a config flag of 0x3CCE
[Header] [0x00] [0x13] [0xF0] [0xC4] [0x3C] [0xCE] [Footer]

Response: None

Reset Device

This command is used to allow users to reset the device without disconnecting power. Certain settings must be followed with a reset for those changes to take effect. The user must issue the reset command after updating trigger information and parameters for the changes to take effect.

Protocol: 0x00
Command: 0x1F
Format: [Protocol] [Command]

Example: a device reset to restart the system
[Header] [0x00] [0x1F] [Footer]

Response: None

Ez-Alert Monitor Table Commands

Ez-Alert's vehicle monitoring functionality is controlled through the use of five calibration tables controlling the settings for Triggers, Faults, Parameters, Parameter Requests, and Message Transmission. All tables are configured and read using four commands, *Read Table Rules, Write Table Rules, Read Table Data, and Clear Table Data.*

Rules Tables

The Rules Tables are used to control what J-Link data is requested, stored, and reported in the data tables for Faults, Parameter, or Triggers. There are also rules tables to define what actions are to be taken when status is queried such as the Parameter Request Table and the Transmit Table.

Write Table Rules

To configure the monitoring settings for a particular table, the user sends the *Write Table Rules* command as shown in each section below. The general format of the read command is shown in Table 1-6.

Protocol	Table Write Command	Table Identifier	Protocol for table data	Table Data
0x00	0x43	0x01 – Recorded Fault Table 0x02 – Recorded Parameter Table 0x03 – Parameter Request Table 0x04 – Trigger Table 0x05 – Transmit Table 0x11 – Reported Fault Table	0x01 – J1708 0x02 – J1939	Defined per table type below

Table 1-6: Write Table Rules format

Filter Rules Table (1)

The Fault Filter Rules Table is a list a SAE® defined Faults loaded into flash by the user. The entries in these tables are compared to J-link reported faults. If a match is found that parameter is stored or rejected, based on the include or exclude flag, in RAM for reporting the next time a report recorded faults or report filtered faults is valid.

Write J1708 Fault Filter rules

Write a J1708 fault filter table to the device. The device can accommodate up to 40 MID/PID faults to filter on. If a user wants to include all reported faults, set the exclude flag for the table and put no MID/PID combinations in the table. The table can hold up to 40 fault codes. To get no faults, set the table to include and put no PIDs in the table.

Protocol: 0x00
Command: 0x43
Format
[Protocol] [Command] [Table Type] [Table Protocol] [Filter Type] [1-N Filter PIDs]

Table Type: per Table 1-6
Table Protocol: 0x01 – J1708
Filter Type: 0 – Include If Fault matches an active Fault
1 – Exclude: if Fault matches
Filter PIDs: J1708/J1587 PIDs to filter on
Response: None

Example: Set to include filter on Fault PID 0x35, 0x36, and 0x37
[Header] [0x00] [0x43] [0x01] [0x01] [0x00] [0x35] [0x36] [0x37] [Footer]

Write J1939 Fault Filter Rules

Write a J1939 fault filter table to the device. The device can accommodate up to 20 SPNs to filter on. If a user wants to include all reported faults, set the exclude flag in the table and put no SPNs in the table. The table can hold up to 20 SPNs.

Protocol: 0x00
Command: 0x43
Format
[Protocol] [Command] [Table Type] [Table Protocol] [Filter Type] [1-N Filter SPNs]

Table Type: per Table 1-6
Table Protocol: 0x02 – J1939
Filter Type: 00 = Include
01 = Exclude
Filter SPNs: J1939 SPNs to filter on
Response: None

Example: Set to only filter on SPNs 0x135 and 36.
[Header] [0x00] [0x43] [0x01] [0x02] [0x00] [0x00] [0x00] [0x01] [0x35] [0x00] [0x00] [0x00] [0x36] [Footer]

Parameter Rules Table (2)

The Parameter Rules Table is a list of parameters loaded into flash by the user. The entries in these tables are compared to all received J-link parameters. If a match is found, that parameter is stored in RAM for reporting at the next request.

Write J1708 Parameter Rules

This function allows the host application to write the current Parameter Table stored in the Ez-Alert Module. The Parameter Table determines what data link quantities will be reported to the

host. The table can hold up to 72 bytes of data. Note that the user must issue the reset command after updating trigger information and parameters for the changes to take effect.

Protocol(s): 0x00
Command: 0x43
Format:
[Protocol] [Command] [Table Type] [Table Protocol] [MIDn] [MSB PIDn] [LSB PIDn]
Table Type: per Table 1-6
Table Protocol: 0x01 - J1708
MIDn:
PID MSBn:
PID LSBn:
Response: None

Example: J1708 Parameter Table with two J1708 Parameters to report
The first is for MID 0x80, PID 0x35, the second is MID 0x30 PID 0x20
[Header] [0x00] [0x43] [0x02] [0x01] [0x80] [0x00] [0x35] [0x30] [0x00] [0x20] [Footer]

Write J1939 Parameter Rules

This function allows the host application to write the current J1939 Parameter Table stored in the Ez-Alert Module. The Parameter Table determines what data link quantities will be reported to the host. The Parameters read from the table are the middle 16-bits of the J1939 PGN. Note that J1939 PGNs can contain more than one parameter. The user must therefore specify the offset and size of the data they would like per PGN. i.e. if you would like to see only data bytes 3 and 4 for a PGN, you would send an offset of 2 (0 referenced) with a size of 2. The table can hold up to 72 bytes of data. Note that the user must issue the reset command after updating trigger information and parameters for the changes to take effect.

Protocol(s): 0x00
Command: 0x43
Format:
[Protocol] [Command] [Table Type] [Table Protocol] [16-bit PGN] [Data Offset] [Data Size]

Table Type: per Table 1-6
Table Protocol: 0x02 - J1939
MSB PGNn: MSB of the J1939 parameter number
LSB PGNn: LSB of the J1939 parameter number
Data Offsetn: Offset of the parameter in the message, in bytes, zero referenced
Data Sizen: Size of the parameter in bytes.
Response: None

Example: Write J1939 parameter table with PGN 0xF009 collecting all 8 data bytes
[Header] [0x00] [0x43] [0x02] [0x02] [0xF0] [0x09] [0x01] [0x08] [Footer]

Parameter Request Table (3)

Since not all parameters are broadcast, or in some cases, just not broadcast at a high enough frequency, the Parameter Request Table provides the ability to request parameters. Each parameters stored in the table will be requested using the appropriate J-Link request sequence. All responses will be filtered using the Parameter Rules Table.

Write J1708 Parameter Request Table

This function allows the host application to write the current Parameter Request Table stored in the Ez-Alert Module. The Parameter Request Table defines what parameters the user would like to automatically request from the data links that are not automatically broadcast. Up to 8 PIDs can be requested.

Protocol(s): 0x00
Command: 0x43
Format: Protocol] [Command] [Table Type] [Table Protocol] [PID1]…[PIDn]

Table Type: per Table 1-6
Table Protocol: 0x01 - J1708
[Data] PID list for Ez-Alert to request, collect, and report to the host
Response: None

Example: Set J1708 Parameter Request Table for PIDs 0x35 and 0x42
[Header] [0x00] [0x43] [0x03] [0x01] [0x35] [0x42] [Footer]

Write J1939 Parameter Request Table

This function allows the host application to write the current Parameter Request Table stored in the Ez-Alert Module. The Parameter Request Table defines what parameters the user would like to request from the data links that are not automatically broadcast. J1939 parameters are passed in using the middle 16-bits of the J1939 identifier, i.e. Ez-Alert only uses the parameter portion and not the addresses, priority, etc. Up to 8 PGNs can be collected.

Protocol(s): 0x00
Command: 0x43
Format: [Protocol] [Command] [Table Type] [Table Protocol] [16-bit PGN1]… [16-bit PGNn]

Table Type: per Table 1-6
Table Protocol: 0x02 - J1939
Data: PGN list for Ez-Alert to request, collect, and report to the host
Response: None

Example: Parameter Request Table set for requesting J1939 PGNs (0xF009, 0xF13C)
[Header] [0x00] [0x43] [0x03] [0x02] [0xF0] [0x09] [0xF1] [0x3C] [Footer]

Write Trigger Rules Table (4)

Due to the complex nature of Triggers they are treated somewhat differently than the other Rules Tables. Instead of having one table of rules for each protocol, Triggers have 10 independent tables. Each table contains the rules for a single trigger, and each is assigned to a specific protocol. All triggers are referenced beginning with zero creating a trigger number range of 0-9 corresponding to trigger numbers 1-10.

J1587 and J1939 protocols are unique enough that the data required for valid trigger detection is significantly different for each of the protocols. However since any of the triggers may be used for either protocol, some of the entries are either not used, or re-defined based on the specified protocol. Note that the user must issue the reset command after updating trigger information and parameters for the changes to take effect.

The generic description is…

Protocol: [0x00]
Command: [0x43]

Data Format:	[Trigger Data as follows…]
[Table Type]	see Table 1-6
[Trigger #]	Trigger number of table (0-9)
[Protocol]	Protocol for which this trigger applies (see Table 1-3 for values)
[Max/Min]	00 - MAX - Trigger above the base value 01 - MIN - Trigger below the base value
[MID/Page]	J1708 MID or J1939 Page
[PID/PGN1]	MSB of J1708 PID/ J1939 PGN
[PID/PGN2]	LSB of J1708 PID/ J1939 PGN
[Size]	Size of the parameter in bits
[Index]	Index into the message data to the parameter to monitor
[Offset MSB]	Data Offset as defined by SAE J1939 or J1708/J1587 (MSB)
[Offset LSB]	Data Offset as defined by SAE J1939 or J1708/J1587 (LSB)
[Scale MSB]	Defined by SAE J1939 or J1708/J1587 multiplied by 1000
[Scale LSB]	
[Base MSB]	Base parameter Value to trigger around
[Base LSB]	
[Min Duration]	Minimum duration the event must last to be considered active
["Name"]	Null Terminated ASCII Name of the trigger (12 characters Max)
["Units"]	Null Terminated ASCII Units of the trigger (12 characters Max)

All of the entries in the table are used to monitor for a trigger condition except for 'Name' and 'Units' These ASCII values are reported so the monitoring stations do not have to correlate trigger table numbers and entries across a fleet. Each trigger may be given a user defined Name and measurement type that is reported each time a valid report trigger data is initiated.

Response: None

Write J1587 Trigger Rule

Due to the nature of the protocol, J1587 message formats are more tightly defined as far as PID size and location. This results in some of the trigger entries being unused, or reserved. In the case of reserved data, any value may be entered, so long as some data is present to provide a place holder. Note that the user must issue the reset command after updating trigger information and parameters for the changes to take effect.

Protocol: [0x00]
Command: [0x43]

Data Format:	[Trigger Data as follows…]
[Table Type]	see Table 1-6
[Trigger #]	Trigger number of table (0-9)
[Protocol]	0x01
[Max/Min]	00 - MAX - Trigger above the base value 01 - MIN - Trigger below the base value
[MID]	J1587 MID
[PID]	MSB of J1587 PID (Current only PIDs 7 - 191 are supported for triggers)
[PID]	LSB of J1587 PID
[Size]	Size - Reserved
[Index]	Index - Reserved
[Offset MSB]	Data Offset as defined by SAE J1708/J1587 (MSB)
[Offset LSB]	Data Offset as defined by SAE J1708/J1587 (LSB)
[Scale MSB]	Defined by SAE J1708/J1587 multiplied by 1000
[Scale LSB]	
[Base MSB]	Base parameter Value to trigger around
[Base LSB]	
[Min Duration]	Minimum duration the event must last to be considered active
["Name"]	Null Terminated ASCII Name of the trigger (12 characters Max)
["Units"]	Null Terminated ASCII Units of the trigger (12 characters Max)

Example: Set trigger 2 rule for J1708 trigger MID 80, PID 34
[Header] [0x00][[0x43] [0x04] [0x01] [0x01] [0x00] [0x80] [0x00] [0x34] [0x08] [0x03] [0x00] [0x08] [0x00] [0x01] [0x00] [0x90] [{'Ret Temp'}] [0x00] [{'Degrees C'}] [0x00] [Footer]

Response: None

Write J1939 Trigger Rule

Due to J1939s large and dynamic data base, the user is required to define the size and location of data that is to be monitored by the trigger logic. As a result, all fields inside the trigger data are used. Note that the user must issue the reset command after updating trigger information and parameters for the changes to take effect.

Protocol: [0x00]
Command: [0x43]

Data Format:	[Trigger Data as follows…]
[Table Type]	see Table 1-6
[Trigger #]	Trigger number of table (0-9)
[Protocol]	0x02
[Max/Min]	00 – MAX – Trigger above the base value 01 – MIN – Trigger below the base value
[Page]	J1939 Page
[PGN1]	MSB of J1939 PGN
[PGN2]	LSB of J1939 PGN
[Size]	Size of the parameter in bits – 1st bit is bit 0.
[Index]	Index into the message data to the parameter to monitor in bits- 1st bit is bit 0.
[Offset MSB]	Data Offset as defined by SAE J1939 (MSB)
[Offset LSB]	Data Offset as defined by SAE J1939 (LSB)
[Scale MSB]	Defined by SAE J1939 multiplied by 1000
[Scale LSB]	
[Base MSB]	Base parameter Value to trigger around
[Base LSB]	
[Min Duration]	Minimum duration the event must last to be considered active
["Name"]	Null Terminated ASCII Name of the trigger (12 characters Max)
["Units"]	Null Terminated ASCII Units of the trigger (12 characters Max)

Example: Set trigger 5 for J1939 trigger Page 0 PGN 0xF000, two byte value at byte 2
[Header] [0x00][[0x43] [0x04] [0x04] [0x02] [0x01] [0x00] [0xF0] [0x00] [0x0F] [0x08] [0x03] [0x00] [0x08] [0x00] [0x01] [0x00] [0x90] [{'Ret Temp'}] [0x00] [{'Degrees C'}] [0x00] [Footer]

Response: None

Message Transmit Table (5)

There are some instances where the user may want specific messages sent on the link when the status is queried. In that case, the message to be sent is stored in the Message Transmit Table and is sent upon receipt of the status request message. In order to facilitate multiple messages, each message entry is made up of byte count, Identifier, and data. The entries in this table are the only instance where the J1708 MID and J1939 Source ID set in calibration may be overridden.

Write J1708 User Transmit Table

This function allows the host application to rite the current J1708 User Transmit Table stored in the Ez-Alert Module. The User Transmit Table controls what messages are sent out on the data link when commanded by the user through the issuing of the *Read Status* message. This table can hold up t o 48 bytes of transmit data.

Protocol(s):	0x00
Command:	0x43
Format:	[Protocol] [Command] [Table Type] [Table Protocol] [Byte Count] [Message] ….

Table Type: per Table 1-6
Table Protocol: 0x01 – J1708
Byte Count: Number of bytes in the message to transmit
Message: J1708 Message to transmit (MID,PID,Data)
Response: None

Example: Write J1708 Transmit Table to send MID: 0xF9, PID: 0x34, Data: 0,1,2,3
[Header] [0x00] [0x43] [0x05] [0x01] [0x05] [0xF9] [0x34] [0x01] [0x02] [0x03] [Footer]

Write J1939 User Transmit Table

This function allows the host application to write the current J1939 User Transmit Table stored in the Ez-Alert Module. The User Transmit Table controls what messages are sent out on the data link when commanded by the user through the issuing of the *Read Status* message. . This table can hold up t o 48 bytes of transmit data.

Protocol(s): 0x00
Command: 0x43
Format: [Protocol] [Command] [Table Type] [Table Protocol] [Byte Count] [Message] ….

Table Type: per Table 1-6
Table Protocol: 0x02 – J1939
Byte Count: Number of bytes in the message to transmit
Message: J1939 Message to transmit (4 Byte ID + Data)
Response: None

Example: J1939 Transmit Table return with one J1939 Message to transmit
ID = 0x1CF00900 Data = 0x01, 0x02, 0x03, 0x04

[Header] [0x00] [0x42] [0x05] [0x02] [0x08] [0x1C] [0xF0] [0x09] [0x00] [0x01] [0x02] [0x03] [0x04] [Footer]

Read Table Rules

To read a particular tables monitoring configuration, the user sends the *Read Table Rules* command as shown in each section below. The general format of the read command is shown in Table 1-7.

Protocol	Table Read Rules Command	Table Identifier	Protocol for table data	Table Data
0x00	0x42	0x01 – Recorded Fault Table 0x02 – Recorded Parameter Table 0x03 – Parameter Request Table 0x04 – Trigger Table 0x05 – Transmit Table 0x11 – Reported Fault Table	0x01 – J1708 0x02 – J1939	Defined per table type below

Table 1-7: Read Table Rules format

Read J1708 Fault Filter Table

This function allows the host application to read the current Fault Filter Table stored in the Ez-Alert Module. The Fault Filter Table determines what faults will be reported to the host.

Protocol(s): 0x00
Command: 0x42
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-7
Table Protocol: 0x01 - J1708

Response: Filter Table
Format: [Protocol] [Command] [Table Type] [Table Protocol][Filter Type] [1-n Data]

Table Type: per Table 1-7
Table Protocol: 0x01 - J1708
Filter Type: 0 - Include if Fault matches an active Fault
1 - Exclude if Fault matches
Data: Fault Filter Data

Example: Read Fault table
[Header] [0x00] [0x42] [0x01] [0x01] [Footer]

Response: Fault filter table return with two J1708 Faults, PIDs 0x35 and 0x42
[Header] [0x00] [0xC2] [0x01] [0x01] [0x00] [0x35] [0x42] [Footer]

Read J1939 Fault Filter Table

This function allows the host application to read the current Fault Filter table stored in the Ez-Alert Module. The fault filter table determines what faults will be reported to the host. Note that SPNs are the only value returned and not the entire DTC which also includes over/under warning, count, and more. SPNs in J1939 are analogous to J1708 PIDs and therefore it is the only data returned. Many PIDs are mapped directly to SPNs but SPNs can be up to 4 bytes long.

Protocol(s): 0x00
Command: 0x42
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-7
Table Protocol: 0x02 - J1939

Response: Filter Table
Format: [Protocol] [Command] [Table Type] [Table Protocol] [Filter Type] [1-n Data]

Table Type: per Table 1-7
Table Protocol: 0x02 - J1939
Filter Type: 0 - Include if Fault matches an active Fault
1 - Exclude if Fault matches
Data: Fault Filter SPN Data

Example: Read J1939 Fault Table
[Header] [0x00] [0x42] [0x01] [0x02] [Footer]

Response: Fault filter table return with one filter for SPN 0x00000035
[Header] [0x00] [0xC2] [0x01] [0x02] [0x00] [0x00] [0x00] [0x00] [0x35] [Footer]

Read J1708 Parameter Table

This function allows the host application to read the current Parameter Table stored in the Ez-Alert Module. The Parameter Table determines what data link quantities will be reported to the host. PID data size is defined by the J1587 specification however the data string reserves two bytes per PID.

PID 7 – 127	1 Byte of Data
PID 128 – 191	2 Bytes of Data
PID 192 –254	1st Byte of Data is data length

Protocol(s): 0x00
Command: 0x42
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-7
Table Protocol: 0x01 - J1708

Response: J1708 Parameter Table
Format:
[Protocol] [Command] [Table Type] [Table Protocol] [MID1] [MSB PID1] [LSB PID 1] [MSB MIDn] [LSB PIDn]

Table Type: per Table 1-7
Table Protocol: 0x01 – J1708
Data: J1708 MID/PIDs to collect and report

Example: Read J1708 Parameter Table
[Header] [0x00] [0x42] [0x02] [0x01] [Footer]

Response: J1708 MID/PIDs to collect and report table return with two J1708 Parameters
The first is for MID 0x80, PID 0x35, the second is MID 0x30 PID 0x20

[Header] [0x00] [0xC2] [0x02] [0x01] [0x80] [0x00] [0x35] [0x30] [0x00] [0x20] [Footer]

Read J1939 Parameter Table

This function allows the host application to read the current J1939 Parameter Table stored in the Ez-Alert Module. The Parameter Table determines what data link quantities will be reported to the host. The Parameters read from the table are the middle 16-bits of the J1939 PGN. Note that J1939 PGNs can contain more than one parameter. The user must therefore specify the offset and size of the data they would like per PGN. i.e. if you would like to see only data bytes 3 and 4 for a PGN, you would send an offset of 3 with a size of 2.

Protocol(s): 0x00
Command: 0x42
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-7
Table Protocol: 0x02 – J1939

Response: J1939 Parameter Table
Format:
[Protocol] [Command] [Table Type] [Table Protocol] [16-bit PGN] [Data Offset] [Data Size]

Table Type: per Table 1-7
Table Protocol: 0x02 – J1939
Data: J1939 PGNs to collect and report

Example: Read J1939 Parameter Table
[Header] [0x00] [0x42] [0x02] [0x02] [Footer]

Response: J1939 parameter table return with PGN 0xF009 collecting all 8 data bytes
[Header] [0x00] [0xC2] [0x02] [0x02] [0xF0] [0x09] [0x01] [0x08] [Footer]

Read J1708 Parameter Request Table

This function allows the host application to read the current Parameter Request Table stored in the Ez-Alert Module. The Parameter Request Table defines what parameters the user would like to request from the data links that are not automatically broadcast.

Protocol(s): 0x00
Command: 0x42
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-7
Table Protocol: 0x01 – J1708

Response: Parameter Request Table
Format: [Protocol] [Command] [Table Type] [Table Protocol] [PID1]…[PIDn]

Table Type: per Table 1-7
Table Protocol: 0x01 – J1708
Data: PID list for Ez-Alert to request, collect, and report to the host

Example: Read J1708 Parameter Request Table
[Header] [0x00] [0x42] 0x03] [0x01] [Footer]

Response: Parameter Request Table return with two J1708 PIDs (0x35, 0x42) to request
[Header] [0x00] [0xC2] [0x03] [0x01] [0x35] [0x42] [Footer]

Read J1939 Parameter Request Table

This function allows the host application to read the current Parameter Request Table stored in the Ez-Alert Module. The Parameter Request Table defines what parameters the user would like to request from the data links that are not automatically broadcast. J1939 parameters are passed in using the middle 16-bits of the J1939 identifier, i.e. Ez-Alert only uses the parameter portion and not the addresses, priority, etc.

Protocol(s): 0x00
Command: 0x42
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-7
Table Protocol: 0x02 – J1939

Response: Parameter Request Table
Format: [Protocol] [Command] [Table Type] [Table Protocol] [16-bit PGN1]… [16-bit PGNn]
Table Type: per Table 1-7
Table Protocol: 0x02 – J1939
Data: PGN list for Ez-Alert to request, collect, and report to the host

Example: Read J1939 Parameter Request Table

[Header] [0x00] [0x42] [0x03] [0x02] [Footer]

Response: Parameter Request Table return with two J1939 PGNs (0xF009, 0xF13C)
[Header] [0x00] [0xC2] [0x03] [0x02] [0xF0] [0x09] [0xF1] [0x3C] [Footer]

Read Trigger Rules Table

Used to read the rules each user defined trigger is operating on.

Protocol: 0x00
Command: [0x42]
Data Format: [Trigger #]
[Trigger #] Trigger number (0-9)

Example: Request Trigger 8 Rules
[Header] [0x00] [0x42] [0x04] [0x01] [0x08] [Footer]

Response:

Data Format:	[Trigger Data as follows…]
[Table Type]	per Table 1-7
[Trigger #]	Trigger number (0-9)
[Protocol]	Protocol for which this trigger applies (see Table 1-2 for values)
[Max/Min]	00 – MAX – Trigger above the base value 01 – MIN – Trigger below the base value
[MID/Page]	J1708 MID or J1939 Page
[PID/PGN1]	MSB of J1708 PID/ J1939 PGN
[PID/PGN2]	LSB of J1708 PID/ J1939 PGN
[Size]	Size of the parameter in bits
[Index]	Index into the message data, in bytes, to the parameter to monitor
[Offset MSB]	Data Offset as defined by SAE J1939 or J1708/J1587 (MSB)
[Offset LSB]	Data Offset as defined by SAE J1939 or J1708/J1587 (LSB)
[Scale MSB]	Defined by SAE J1939 or J1708/J1587 times 1000
[Scale LSB]	
[Base MSB]	Base parameter Value to trigger around MSB
[Base LSB]	Base parameter Value to trigger around LSB
[Min Duration]	Minimum duration the event must last to be considered active
[Name]	12 character ASCII name of the trigger
[Units]	12 character ASCII name of the units defining the parameter

Example: Read trigger 3 rule response for J1708 trigger MID 80, PID 34
[Header] [0x00][[0xC2] [0x04] [0x02] [0x01] [0x00] [0x80] [0x00] [0x34] [0x08] [0x03] [0x00] [0x08] [0x00] [0x01] [0x00] [0x90] [{'Ret Temp'}] [0x00] [{'Degrees C'}] [0x00] [Footer]

Read J1708 User Transmit Table

This function allows the host application to read the current J1708 User Transmit Table stored in the Ez-Alert Module. The User Transmit Table controls what messages are sent out on the data link when commanded by the user through the issuing of the *Read Status* message.

Protocol(s): 0x00

Command: 0x42
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-7
Table Protocol: 0x01 – J1708

Response: Transmit Table
Format: [Protocol] [Command] [Table Type] [Table Protocol] [Byte Count] [Message] ….

Table Type: per Table 1-7
Table Protocol: 0x01 – J1708
Byte Count: Number of bytes in the message to transmit
Message: J1708 Message to transmit (MID,PID,Data)

Example: Read J1708 Transmit Table
[Header] [0x00] [0x42] [0x05] [0x01] [Footer]

Response: J1708 Transmit Table return with one J1708 Message 0x80,0x34,0x01
[Header] [0x00] [0xC2] [0x05] [0x01] [0x03] [0x80] [0x34] [0x01] [Footer]

Read J1939 User Transmit Table

This function allows the host application to read the current J1939 User Transmit Table stored in the Ez-Alert Module. The User Transmit Table controls what messages are sent out on the data link when commanded by the user through the issuing of the *Read Status* message.

Protocol(s): 0x00
Command: 0x42
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-7
Table Protocol: 0x02 – J1939

Response: Transmit Table
Format: [Protocol] [Command] [Table Type] [Table Protocol] [Byte Count] [Message] ….

Table Type: per Table 1-7
Table Protocol: 0x02 – J1939
Byte Count: Number of bytes in the message to transmit
Message: J1939 Message to transmit (4 Byte ID + Data)

Example: Read J1939 Transmit Table
[Header] [0x00] [0x42] [0x05] [0x02] [Footer]

Response: J1939 Transmit Table return with one J1939 Message to transmit
ID = 0x1CF00900 Data = 0x01, 0x02, 0x03, 0x04
[Header] [0x00] [0xC2] [0x05] [0x02] [0x08] [0x1C] [0xF0] [0x09] [0x00] [0x01] [0x02] [0x03] [0x04] [Footer]

Read Table Data

To read a particular table's current collected data, the user sends the *Read Table Data* command as shown in each section below. The Transmit Tables and the Parameter Request Tables do not have table data associated with them. It should be noted that the Parameter Request Table does control the parameters requested by the Ez-Alert module and hence any responses to those requests will be reported as table data in the Recorded Parameter Table.

The general format of the read command is shown in Table 1-8.

Protocol	Table Read Data Command	Table Identifier	Protocol for table data	Table Data
0x00	0x44	0x01 – Recorded Fault Table 0x02 – Recorded Parameter Table 0x04 – Trigger Table 0x11 – Reported Fault Table	0x01 – J1708 0x02 – J1939	Defined per table type below

Table 1-8: Read Table Data format

Read J1708 Recorded Fault Table Data

This function allows the host application to read the current Fault Table Data stored in the Ez-Alert Module. This table holds historic or recorded fault codes. Use the Reported Fault Table for only active faults. The format of the fault code and its data is defined by SAE J1587, Section A.194 and should be referenced for data formatting information.

Protocol(s): 0x00
Command: 0x44
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-8
Table Protocol: 0x01 - J1708

Response: Recorded J1708 Fault Codes
Format:
[Protocol] [Command] [Table Type] [Table Protocol] [FCn] [FC Data1 n] [FC Data2 n]……

Table Type: per Table 1-8
Table Protocol: 0x01 - J1708
Fault Code n: J1587 Fault Code for the parameter
Fault Code n Data 1: Diagnostic Code Character as defined by J1587 standard
Fault Code n OC: Occurrence Count, only present if bit 7 of previous byte is set, as defined by J1587 standard

Example: Read Fault table

[Header] [0x00] [0x44] [0x01] [0x01] [Footer]

Response: Recorded fault data table return with two J1708 Faults 0x35 DCC of 0x27 no occurrence count
[Header] [0x00] [0xC4] [0x01] [0x01] [0x35] [0x27] [Footer]

Read J1939 Recorded Fault Table Data

This function allows the host application to read the current J1939 Recorded Fault Table data stored in the Ez-Alert Module. These recorded DTCs are historic and may or may not still be active on the vehicle. Use the Reported Fault Table to see only currently active DTCs/SPNs . SPNs in J1939 are analogous to J1708 PIDs and therefore it is the only data returned. Many PIDs are mapped directly to SPNs but DTCs are 4 bytes long. The J1939-73 standard should be referenced for data formats. The DTC is well defined and reported here exactly as seen in the specification and hence on the data link.

Protocol(s): 0x00
Command: 0x44
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-8
Table Protocol: 0x02 - J1939

Response: J1939 Recorded DTCs
Format: [Protocol] [Command] [Table Type] [Table Protocol] [DTC1] … [DTCn]

Table Type: per Table 1-8
Table Protocol: 0x02 - J1939
Data: J1939 Recorded 4-byte Diagnostic Trouble Codes [DTC]

Example: Read J1939 Recorded Fault Table Data
[Header] [0x00] [0x44] [0x01] [0x02] [Footer]

Response: Recorded Fault Table return with one DTC 0x04030135
[Header] [0x00] [0xC4] [0x01] [0x02] [0x35] [0x01] [0x03] [0x04] [Footer]

Read J1708 Parameter Table Data

This function allows the host application to read the current Parameter Table Data stored in the Ez-Alert Module. The Parameter Table Data contains the J1708 quantities that the host has asked the module to store and report. This data also contains the parameters returned in response to the Request Table parameter requests on the data link.

Protocol(s): 0x00
Command: 0x44
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-8
Table Protocol: 0x01 – J1708

Response: Recorded J1708 Parameters
Format:
[Protocol] [Command] [Table Type] [Table Protocol] [MID1] [PID1] [Data1] [MIDn] [PIDn] [Data n]

Table Type: per Table 1-8
Table Protocol: 0x01 – J1708
MIDn: J1708 MID associated with this reported parameter
PIDn: J1708 PID for this parameter
Datan: Data associated with this parameter, Note the data size is controlled by the user setting in the parameter rules table.

Example: Read J1708 Parameter Table Data
[Header] [0x00] [0x44] [0x02] [0x01] [Footer]

Response: J1708 Parameters (MID,PID,Data) collected and report return with two Parameters. The first is for MID 0x80, PID 0x35, Data-0x04, the second is MID 0x30 PID 0x20 Data = 0x01, 0x02

[Header] [0x00] [0xC4] [0x02] [0x01] [0x80] [0x35] [0x04] [0x30] [0x20] [0x01] [0x02] [Footer]

Read J1939 Parameter Table Data

This function allows the host application to read the current J1939 Parameter Table Data stored in the Ez-Alert Module. The Parameters read from the table are the middle 16-bits of the J1939 PGN followed by the data requested. Note that J1939 PGNs can contain more than one parameter.

Protocol(s): 0x00
Command: 0x44
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-8
Table Protocol: 0x02 – J1939

Response: J1939 Parameter Table Data
Format:
[Protocol] [Command] [Table Type] [Table Protocol] [MSB PGN] [LSB PGN] [Byte Count] [Data]…..

Table Type: per Table 1-8
Table Protocol: 0x02 – J1939
MSB PGN: Most significant byte of the16-bit PGN

LSB PGN: Least significant byte of the16-bit PGN
Byte Count: Number of data bytes that follow
Data: Collected J1939 PGNs and associated data

Example: Read J1939 Parameter Table
[Header] [0x00] [0x44] [0x02] [0x02] [Footer]

Response: J1939 parameter table data return with PGN 0xF009 and 8 data bytes
[Header] [0x00] [0xC4] [0x02] [0x02] [0xF0] [0x09] [0x08] [0x01] [0x02] [0x03] [0x04] [0x05] [0x06] [0x07] [0x08] [Footer]

Read Trigger Table Data

Used to read active trigger data.

Protocol: [0x00]
Command: [0x44]
Data Format:

Example: Request active trigger data for J1708 Trigger 1
[Header] [0x00] [0x44] [0x04] [0x01] [0x01] [Footer]

Response:
Data Format:
[Protocol] [Byte Count] [Table Type] [Trigger #] [MSB Duration] [LSB Duration] [MSB Offset] [LSB Offset]

[Trigger #] Trigger number being reported
[16-bit Duration] Trigger event duration: one, two byte value per occurrence in seconds
[16-bit Offset] Trigger value offset from the trigger point

Example: Trigger 1 response for J1708 oil pressure lasting 0x264 seconds with an offset from the trigger point of 0x21.
[Header] [0x00] [0xC4] [0x04] [0x01] [0x02] [0x64] [0x00] [0x21] [Footer]

Read J1708 Reported Fault Table Data

This function allows the host application to read the current Reported Fault Table Data stored in the Ez-Alert Module. This table reports only fault codes that are currently active on the vehicle.

Protocol(s): 0x00
Command: 0x44
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-8
Table Protocol: 0x01 - J1708

Response: Active J1708 Fault Codes

Format: [Protocol] [Command] [Table Type] [Table Protocol] [FC1] .. [FCn]

Table Type: per Table 1-8
Table Protocol: 0x01 - J1708
Data: [Fault Code 1]…[Fault Code n]

Example: Read Fault table
[Header] [0x00] [0x44] [0x11] [0x01] [Footer]

Response: Recorded fault data table return with two J1708 Faults 0x35 and 0x42
[Header] [0x00] [0xC4] [0x11] [0x01] [0x35] [0x42] [Footer]

Read J1939 Reported Fault Table Data

This function allows the host application to read the current J1939 Reported Fault Table Data stored in the Ez-Alert Module. These recorded SPNs are only the currently active SPNs reported by different vehicle systems. Note that SPNs are the only value returned and not the entire DTC which also includes over/under warning, count, and more. SPNs in J1939 are analogous to J1708 PIDs and therefore it is the only data returned. Many PIDs are mapped directly to SPNs but SPNs can be up to 4 bytes long.

Protocol(s): 0x00
Command: 0x44
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-8
Table Protocol: 0x02 - J1939

Response: J1939 Recorded SPNs
Format: [Protocol] [Command] [Table Type] [Table Protocol] [SPN1] … [SPN n]

Table Type: per Table 1-8
Table Protocol: 0x02 - J1939
Data: J1939 Recorded 4-byte Suspect Parameter Numbers [SPN]

Example: Read J1939 Recorded Fault Table Data
[Header] [0x00] [0x44] [0x11] [0x02] [Footer]

Response: Recorded Fault Table return with one SPN 0x00000035
[Header] [0x00] [0xC4] [0x11] [0x02] [0x00] [0x00] [0x00] [0x35] [Footer]

Clear Table Data

To read a particular table's current collected data, the user sends the *Read Table Data* command as shown in each section below. The Transmit Tables and the Parameter Request Tables do not have table data associated with them. It should be noted that the Parameter Request Table does control the parameters requested by the Ez-Alert module and hence any responses to those requests will be reported as table data in the Recorded Parameter Table.

The general format of the clear command is shown in Table 1-9.

Protocol	Table Clear Command	Table Identifier	Protocol for table data	Table Data
0x00	0x45	0x01 – Recorded Fault Table 0x02 – Recorded Parameter Table 0x04 – Trigger Table 0x11 – Reported Fault Table	0x01 – J1708 0x02 – J1939	Defined per table type below

Table 1-9: Clear Table Data format

Clear Table Data

This function allows the host application to clear the specified table's data. This is an asynchronous command and can be issued at any time to begin collecting new data.

Protocol(s): 0x00
Command: 0x45
Format: [Protocol] [Command] [Table Type] [Table Protocol]
Table Type: per Table 1-8
Table Protocol: 0x02 - J1939

Response: None

Example: Clear J1939 Recorded Fault Table Data
[Header] [0x00] [0x45] [0x11] [0x02] [Footer]

Ez-Alert File Storage

Ez Alert modules provide a 64Kbyte file storage feature. This feature allows the user to read and write user data to the on-board Ez-Alert device. This non-volatile memory may be used for any purpose the user wants, such as vehicle maintenance records, fault tacking, or any other data.

The file storage architecture is shown in Figure 1-1.

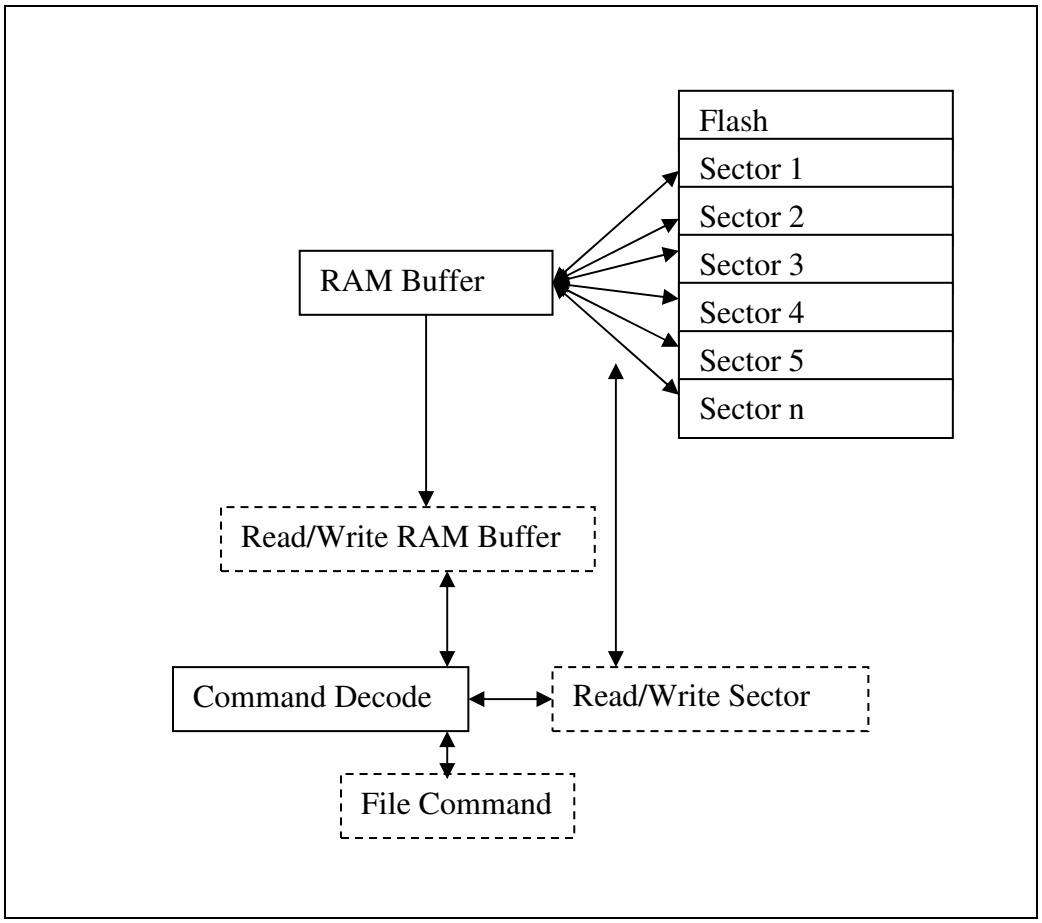

Figure 1-1: File storage flash memory architecture

Because of communication overhead issues and flash erase cycles the following scheme has been adapted for file transfer.

Write – The application will write data into a temporary RAM buffer the size of the current flash sector using the *Write Data* command. Once the entire temporary buffer is filled (up to the current sector size). The application will issue a *Write Sector* command and the RAM buffer will be transferred into the specified flash sector.

Read – The application will issue a *Read Sector* command and the specified sector will be transferred into the temporary RAM buffer from flash memory. The application may then read the RAM buffer at will while adhering to any communications limitations.

Read Storage Properties

Reads the on-board memory format and size in a remote module. This command can be used by the host system to determine the architecture of the file storage area. Note that if the sector and block size are equal, the module will not return a [Sectors per block] parameter indicating this arrangement. Because different cores can have multiple blocks with different sizes, the command will return a [Sectors per block] parameter for each of the 'N' blocks called out by the [Number of Sectors] parameter.

Protocol(s): 0x00
Command: [0x50]
[0x50] Format Read Storage Properties
[Header] [Protocol] [Command] [Footer]

Response: Format
[Protocol] [Command] [Number of Sectors][2 byte Sector Size] [Sectors per Block]

Type: Fixed Format, Variable length

Example: Request ID

Response: Device has 8 sectors each 0x200 bytes long, 1 sector per block
[Header] [0x00] [0xD0] [0x08] [0x20] [0x00] [0x01] [Footer]

Erase Block

Erases the block of on-board memory.

Protocol(s): 0x00
Command: [0x51]
Format
[Header] [Protocol] [Command] [Sector] [Block] [Footer]

Command Type: Fixed Format, Fixed length

Read Buffer

Reads data from specified location in RAM Buffer

Protocol(s): 0x00
Command: [0x52]
Byte Count: Number of bytes to be read
Address: Address of data in Buffer to be read

Format:
[Protocol] [Command] [Byte Count] [RAM Address MSB] [RAM Address LSB]

Command Type: Fixed Format, Fixed length

Response:
[Header] [Protocol] [Command] [Byte Count] [Address MSB] [Address LSB] [Data] [Footer]

Write Buffer

Writes data to specified location in RAM Buffer

Protocol(s): 0x00
Command: [0x53]
Format:
[Header] [Protocol] [Command] [Byte Count] [Address MSB] [Address LSB] [Data] [Footer]

Byte Count: Number of bytes to be written
Address MSB: Address of Buffer to be written
Address LSB:

Read Sector

Read Data in specified sector into RAM buffer

Protocol(s): 0x00
Command: [0x54]
[0x01] Read ID
Format:
[Header] [Protocol] [Command] [Sector] [Footer]

Sector: Sector number to be read into buffer

Response: None

Write Sector

Writes Data in RAM buffer to specified sector

Protocol(s): 0x00
Command: [0x55]

Format: [Header] [Protocol] [Command] [Sector] [Footer]

Sector: Sector number to write RAM buffer into.

Data Set

Set entire buffer to specified data value

Protocol(s): 0x00
Command: [0x57]

Format: [Header] [Protocol] [Command] [Data Value] [Footer]

Data Value: Unsigned char that buffer is to be loaded with

Response: None

Read Sector CRC

Reads the CRC of a requested sector to verify data content

Protocol(s): 0x00
Command: [0x58]
Format
[Header] [Protocol][Command] [sector #] [Footer]

Response:
[XX] Firmware Major Version
[YY] Firmware Minor Version

Format
[Protocol] [Command] [CRCMSB] [CRCLSB] [Footer]

Read File Firmware Version

Reads the on-board file transfer application version

Protocol(s): 0x00
Command: [0x5B]
Format
[Header] [Protocol] [Command] [Footer]

Response Format: [Header] [Protocol] [Command] [MSB Version] [LSB Version] [Footer]

[XX] Firmware Major Version
[YY] Firmware Minor Version

Flash Application

Copies a new application to the microprocessor overwriting the existing functionality. The application file must first be transferred to the system memory from the host in the same manner as any file the host would like to store in the module. This file is very unique and must be obtained directly from XscapeEz Ltd. Note that flashing an unapproved file can cause undesirable operation or even render the device completely inoperable.

Protocol(s): 0x00
Command: [0x5F]
Format
[Header] [Protocol] [Command] [Footer]

Response Format: None

Ez-Alert RS-232 Wiring Information

Ez Alert devices packaged with XELs RS-232 interface will use a DB-15 female connector with the following pin mapping:

Pin Name	Pin Number	Description
Ground	6	Digital Ground
Power	8	8-24V Power
J1939 -	12	
J1939 +	13	
J1708 (ATA) -	14	
J1708 (ATA) +	15	