Protocol for Ethernet Communications

Revision 5.2

1-17-2012

Table of Contents

Overview:

Introduction:

This document defines the Communications protocol used by Ashly Audio Ethernet Devices, such as the PE Series Amplifiers, to communicate over a standard UDP/ IP network. The Protocol will sit atop the UDP protocol on a typical IP Stack (this is the application layer). This allows PCs and other devices to communicate with the Ashly Device without specialized hardware or software/firmware. Software that can use UDP can communicate with the device. The specific ports for the UDP protocol are defined in the various Protocol definitions.

Unit Identification:

Each unit is identified on the network by its "MAC" Address. This value is set by the factory and cannot be changed. This value is purchased from IEEE and guarantees there are no duplicates in the world. Also, the unit will require other configuration parameters as defined by the TCP/IP protocol, such as an IP address. Configuration of these parameters is discussed in section, "**Configuring and establishing Communications."**

Initial State:

The initial state of a device will have DHCP enabled as well as the IP Parameters cleared. Thus it will require IP Configuration.

Configuring and establishing Communications:

Although each device is shipped with a predefined MAC Address, specific parameters for IP communications must be set by the end user. The Ashly Device provides several ways to do this: Manually using Ashly System software, Automatically using Ashly System Software or by a DHCP server. To use Ashly System Software to configure a device's IP Parameters, the PC hosting the software "MUST" be connected to the network. Typically, this means that the PC must be connected to the same "Subnet". However, if this is not possible, the PC may also be connected to the Ashly Device directly via "Crossover Cable." IP parameters may be preset in this manner prior to connecting the device to a network.

If a simple connection between a PC and a device is desired this may be accomplished with a "Crossover Cable." For this type of connection IP Configuration is not necessary. Leaving the parameters set blank (0.0.0.0) will allow the software and device to communicate.

Protocol Breakdown:

The Ashly Protocol can actually be broken down into several smaller protocols. Each is associated with a specific server that exists on the Ashly Device. These servers are the "Get Parameter Server", "Set Parameter Server" and "Update Request Server." These specific protocols define the Transport Layer (UDP) information as well as its purpose and message structure. Although the Header information varies between protocols, the "Options" or data payloads carried by the protocols is standardized. This is defined in the Section titled, "**Ashly Options.**" It is worth noting that each protocol may define the way in which the option is used as well as if the option is used at all. The receiving server will ignore any Option that is not understood (or used) by a particular server. This allows for flexibility as well as future expansion of the protocol. It is also worth noting that more than one option may be used in any given message up to a defined maximum length (MTU) defined by the Ethernet network or 1200 bytes whichever is smaller.

NOTE: THIS PROTOCOL USES UDP NOT TCP. TCP Clients WILL NOT WORK (Please see "TCP Extensions" for more information)

Get Parameter Server

Overview:

This server handles any requests for the current state of parameters within the device, such as the current mute state of channel one. To accomplish this, a PC must send this message to a device with the request/reply byte set to “request,” as well as a payload containing all requested options. Since these options contain no data, it is acceptable to set their length to 0 and not include dummy data (Some Parameters Such as Channel number may be required). The device server will respond to the sending server with the reply message containing the options with their correct data.

Dynamic Update Overview:

This protocol also supports dynamic updates. These are updates that are sent out when a device’s parameters change (excluding meters). Two servers handle dynamic updates. The “Request Updates Server” handles administration of who wishes to receive updates, while the “Get Parameter Server” is used to send the updates. Any PC that has requested updates will receive a Get Parameter reply containing updates every time a device parameter changes. Note that changes made from a PC will not be updated on that PC. This is to avoid jitter in the controls.

Heartbeat Overview:

This protocol also supports a heartbeat. The heartbeat is a request reply mechanism to ensure a device is still functional. To accomplish this, send a Request Parameter message with no options (except the end option). The server will reply with no options (except the end option). If this is done periodically it achieves the functionality of a heartbeat for the device.

Transport Layer:

This server will listen for requests on Port number 3100. Responses to requests received are sent back to the originating port and IP/MAC). Any parameters requested that are not available will be ignored.

Message Structure:

This structure supports both the request and reply. A Request message may set the length of all options to 0 and thus omit the option data fields. The reply message should not do this. In a request message the options denote request information where in the reply message the options correspond to the current state of the device.

Byte#	Value	Description
1-4	0x8F	Header Identifier
5-10	xx	Destination MAC Address (XX-XX-XX-XX-XX-XX)
11	0y	Request/reply (0 = request, 1 = reply)
12	xx	Source 00 – Default (working Settings), 1 to X = Presets 1 to X
13	0x00	Reserved, Inter-box Communications
14	0x00	Reserved, Value ignored should be set to 0x00
15-??	--	Option Payload must be terminated with END_OPT (0xFF)

Set Parameter Server

Overview:

The Set Parameter Server is a password secured server that is used to remotely modify the state of a device. The user must know a user name and password to use this protocol. Also, this protocol uses an acknowledgment to guarantee arrival of messages.

Transport Layer:

This server will use UDP port number 3100. All servers using this protocol must listen for updates on Port 3100. When an update message is received the device will update its parameters and send an acknowledgment back to the originating server and port number.

Message Structure:

For this message any option that does not pertain to a device will be ignored. If a message fails, a special security failure acknowledgment will be returned.

Byte#	Value	Description
1-4	0xAA	Header Identifier
5-10	xx	Destination MAC Address (XX-XX-XX-XX-XX-XX)
11-18	yy	User Name
19-26	yy	Password
27	zz	Message Number (byte 1)
28	zz	Message Number (byte 2)
29	aa	Acknowledgment status
30	0x00	Reserved
31-??	--	Option Payload must be terminated with END_OPT (0xFF)

Message Number:

A number associated with a message. This should be implemented as a circular counter between 0 and 65535. For correct messages received the acknowledgment may contain only the Header information sent.

Acknowledgment Status:

The status of the acknowledgment may be any of the following:
0 – indicates that it is the original message (not the acknowledgment)
1 – indicates the message was received okay
2 – indicates the message was not passed due to insufficient security
3 – indicates that at least one parameter in the previous message was not accepted due to insufficient resources (namely DSP horsepower)
4 – indicates a DSP Error has occurred (reserved byte 31 indicates the Error Code)
5 – indicates that it is already in bulk update mode (returned only for OPT_BULK_UPDATE)
16 - Save to Temp Buffer (original message) (Processed)

User Name and Password:

These two fields denote the user name a password to be used when determining the security of a message. If the message contains parameters for which it does not have security the entire message is ignored and an acknowledgment stating security failure is sent. For the default user, the user name should equal “default”. For the Administrative user, the user name should equal “admin”. Also note, the password field is ignored for the default user. Any characters in the name not used should be set to 0x00.

Request Updates Server

Overview:

The Request Updates Server allows a PC to subscribe to changes in the Ashly device. A PC, or other device, can request to be updated for a specific amount of time (specified by a timeout). This ensures that the number of PCs does not grow due to PCs that have changed IP's or gone offline. When this timeout expires, the Ashly device will send a message using this structure to the receiving PC's IP/Port specified for the updates. All data in this message may be ignored except the Destination MAC, which will contain the MAC Address of the Device requesting the PC to re-login. At this point, the PC should request Updates again if it is still alive.

Transport Layer:

This Protocol will use UDP port number 3100. All servers using this protocol must listen for Client Requests on UDP port 3100.

Message Structure:

This structure is used by a PC to request dynamic updates from a device. In this message, the Timeout specifies the amount of time for which the device will update the PC before the device will ask the PC to re-request updates. The Port Number specifies the UDP Port number to which to send updates. The payload is not used by this message. However, as a matter of convention the END_OPT should still be appended to the message. The IP to which the updates will be sent is the message source IP received by the device. This means that if it goes through a firewall, it will correctly use the firewall's IP (NOT the local IP). When an update is sent from a device, it will use the Get Parameter format and be sent to the port specified in this message (its header info will be 0x8F).

Byte#	Value	Description
1-4	0x55	Header Identifier
5-10	xx	Destination MAC Address (XX-XX-XX-XX-XX-XX)
11	yy	Timeout (byte 1) MSB
12	yy	Timeout (byte 2) LSB
13-16	0x00	Reserved
17	aa	Port (byte 1) MSB
18	aa	Port (byte 2) LSB
19-20	0x00	Reserved, Value ignored should be set to 0x00
21-??	--	Option Payload must be terminated with END_OPT (0xFF)

Appendix A:

Ashly Options

Overview:

The following are a standard set of Options that may be used in Ethernet communications with Ashly products. All Options follow the convention of Identifier followed by length (excluding identifier and length byte) followed by any channel or card identifiers followed by data. The start of the options is specified by the particular protocols. To parse, read the identifier and length. If the option is not understood skip the specified length to read the next option. The END_OPT signifies the end of the options field.

0x02	OPT_MUTE
0x04	OPT_CHANNEL_NAME
0x05	OPT_AMP_MODE
0x06	OPT_POWER_STATUS
0x09	OPT_ATTENUATION
0x0A	OPT_GAIN_INC_DEC
0x0D	OPT_STANDBY
0x0F	OPT_INPUT_CONFIG
0x10	OPT_DVCA_LEVEL
0x11	OPT_DVCA_LINK_GROUP
0x12	OPT_DVCA_NAME
0x16	OPT_COBRANET_RECEIVER
0x17	OPT_COBRANET_RECEIVER_MAP
0x18	OPT_INPUT_SOURCE
0x19	OPT_LOGIC_BLOCK
0x1A	OPT_COBRANET_TX
0x1B	OPT_COBRANET_TX_MAP
0x1C	OPT_LOGIC_PIN
0x25	OPT_FR_CONFIG
0x26	OPT_FR_FADER_CONFIG
0x2D	OPT_RTC
0x2E	OPT_RTC_EVENT
0x2F	OPT_FAULT_LOG
0x30	OPT_PRESET_INFO
0x31	OPT_PRESET_SAVE
0x32	OPT_PRESET_RECALL
0x33	OPT_DEVICE_MODIFIED
0x34	OPT_TOTAL_PRESETS
0x35	

0x40	OPT_METER_INPUT
0x41	OPT_METER_OUTPUT
0x42	OPT_METER_TEMP
0x43	OPT_METER_CURRENT
0x45	OPT_METER_ATTENS
0x46	OPT_CHANNEL_PROTECT
0x47	OPT_AES_RECEIVER
0x48	OPT_LOGIC_STATE
0x49	OPT_WORDCLOCK_STATUS
0x4A	OPT_AMP_METER_PROTECT
0x4B	OPT_AMP_GAIN_SETTING
0x50	OPT_REMOTE_RD8C
0x53	OPT_REMOTE_LEVEL
0x6E	OPT_DEVICE_NAME
0x6F	OPT_DEVICE_GROUP
0x70	OPT_ILL_FRONT
0x71	OPT_DEVICE_INFO
0x78	OPT_COBRANET_INFO
0x79	OPT_COBRANET_STATUS
0x7A	OPT_COBRANET_RX_STATUS
0x81	OPT_DSP_FUNC_PARAMS
0x82	OPT_DSP_FUNC_METERS
0x88	OPT_DSP_CHANNEL_METER
0xF9	OPT_PAD
0xFF	OPT_END

Note: “0x” indicates hexadecimal value.

<table>
<tr><td>Name</td><td>OPT_MUTE</td></tr>
<tr><td>Option #</td><td>0x02</td></tr>
<tr><td>Description</td><td>Mute control from Protea NE Products. Will Mute/Unmute an Input or Output Channel as selected. Will mute only one channel. If more than one channel must be muted, use multiple Options.

Channel Type (byte aa)
- 0 = Output Channel (Amplifier Channels are Outputs)
- 1 = Input Channel

Channel Number (byte bb)
- 0 to 23 = Channels 1 to 24

Mute Status (byte cc)
- 0x00 = Mute Off (unmuted)
- 0x01 = Mute On (muted)

For Example to Mute PE Amplifier Channel 1 the option would be
0x02 0x03 0x00 0x00 0x01</td></tr>
<tr><td>Implemented Products</td><td>All
The maximum Input and output Channels do change per product.
Unimplemented channels will be ignored.</td></tr>
<tr><td>Attributes</td><td>Read/Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x02</td><td>Option Mute Identifier</td></tr>
<tr><td>2</td><td>3</td><td>Bytes to follow</td></tr>
<tr><td>3</td><td>aa</td><td>Channel Type</td></tr>
<tr><td>4</td><td>bb</td><td>Channel Number</td></tr>
<tr><td>5</td><td>cc</td><td>Mute status</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_CHANNEL_NAME</td></tr>
<tr><td>Option #</td><td>0x04</td></tr>
<tr><td>Description</td><td>Channel Name for a particular Channel. Can be written to if a channel name needs to be changed.

Channel Type (byte aa)
- 0 = Output Channel (Amplifier Channels are Outputs)
- 1 = Input Channel

Channel Number (byte bb)
- 0 to 23 = Channels 1 to 24

Channel Name (20 character string cc)
- Valid ASCII Characters 0x20 to 0x7A and 0x00
- Fill (pad) unused characters with 0x00
- String should be left justified (MSB justified)</td></tr>
<tr><td>Implemented Products</td><td>All
Note: Not All Protea products have 24 inputs or 24 outputs</td></tr>
<tr><td>Attributes</td><td>Read Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x04</td><td>Option Channel Name</td></tr>
<tr><td>2</td><td>22</td><td>Bytes to Follow</td></tr>
<tr><td>3</td><td>aa</td><td>Channel Type</td></tr>
<tr><td>4</td><td>bb</td><td>Channel Number</td></tr>
<tr><td>5-25</td><td>cc</td><td>Channel Name</td></tr>
</table>
</td></tr>
</table>

Name	OPT_AMP_MODE
Option #	0x05
Description	Amp Mode allows the Amplifier Model and Amplifier Mode to be read. However these values cannot be changed via software. The Amplifier Mode may be changed via a switch on the rear of the Amplifier Amp Model (byte aa) - Valid Models (1 = 800, 2 = 1200, 3 = 1800, 4 = 2400, 5 = 3000, 6 = 3800) *(7 = 4250, 8 = 8250, 9 = 4070, 10= 8070) - Can not be changed Amp Mode (byte bb) - Valid Modes (0 = Stereo, 1 = Bridged Mono, 2 = Parallel Mono) - *Bitwise for additional channels (4 = BM 3 & 4, 8 = BM 5 & 6, 16 = BM 7 & 8) PM is NOT VALID for a MultiChannel Amplifier - Can only be changed via physical switch on rear of amplifier. - Programmable in software in some amplifiers (Zone Mixer) * Valid only for MultiChannel (>2) PE Series Amplifiers)
Implemented Products	All PE, NE Series Amplifiers Inputs, Standard, DSP, CobraNet
Attributes	Read Only (Read/ Write in Zone Mixer)
Byte Description	Byte # / Value / Description 1 / 0x05 / Option Amp Mode 2 / 2 / Bytes to follow 3 / aa / Amp Model 4 / bb / Amp Mode

Name	OPT_POWER_STATUS
Option #	0x06
Description	Power Status of an Amplifier. This indicates if the Amplifier section is currently powered On or not (standby). This is read only. Status: (byte xx) - 1 = Standby Power Only - 0 = Power On
Implemented Products	All PE Series Amplifiers Inputs, Standard, DSP, CobraNet
Attributes	Read Only
Byte Description	Byte # / Value / Description 1 / 0x06 / Option Power Status 2 / 1 / Bytes to Follow 3 / xx / Status

<table>
<tr><td>Name</td><td>OPT_ATTENUATION</td></tr>
<tr><td>Option #</td><td>0x09</td></tr>
<tr><td>Description</td><td>The Attenuation Option allows the Amplifier's Digitally Controlled Analog attenuators to be set for each Amplifier Channel

Channel Type (byte aa)
- 0 = Amp Channel (currently only supported channel Type)
- 1-255 – ignored

Channel (byte bb)
- Amplifier Channel Number to modify
- 0 to 1 = Amplifier Channels 1 & 2

Attenuation (byte cc)
- 0 to 40 = 0dB to –40dB in 1dB steps
- 41 to 255 = off (typically 255 used as OFF)

Polarity (byte dd)
- 0 = normal,
- 1 to 255 = inverted

Link Group (byte ee)
- Link Group for Offset Linking
- 0 = None
- 1-8 = Link Group 1-8

Offset (byte ff)
- Offset Linking Attenuation
- 0 to 40 = 0dB to –40dB in 1dB steps
- 41 to 255 = off (typically 255 used as OFF)</td></tr>
<tr><td>Implemented Products</td><td>All PE Series Amplifiers Inputs, Standard, DSP, CobraNet</td></tr>
<tr><td>Attributes</td><td>Read/ Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x09</td><td>Option Attenuation</td></tr>
<tr><td>2</td><td>6</td><td>Bytes to Follow</td></tr>
<tr><td>3</td><td>aa</td><td>Channel Type</td></tr>
<tr><td>4</td><td>bb</td><td>Channel</td></tr>
<tr><td>5</td><td>cc</td><td>Attenuation</td></tr>
<tr><td>6</td><td>dd</td><td>Polarity</td></tr>
<tr><td>7</td><td>ee</td><td>Link Group</td></tr>
<tr><td>8</td><td>ff</td><td>Offset</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_GAIN_INC_DEC</td></tr>
<tr><td>Option #</td><td>0x0A</td></tr>
<tr><td>Description</td><td>Function that allows Gain Increment/Decrement Commands.

In PE Products this will modify the attenuators Only. It does not affect the DSP Gain settings.

In NE Products this will modify the attenuators if No DSP is installed. If a DSP is installed this will affect the DSP Gain Block. For this function to have any affect in a NE Product with DSP the DSP Gain Block must be inserted on the channel adjusted.

This Command can be modified to adjust the amount in dB of the change as well as the direction

cc - Increment/ Decrement:
0 - Decrement (Decrease Gain, Increase Attenuation). Quieter
1 – Increment (Increase Gain, Decrease Attenuation). Louder

dd – Amount of Change
- Will determine how large of a change to make
- 0 = 0.5dB – Not Valid for Attenuators which only support whole dB changes
- 1 = 1.0dB
- 2 = 2.0dB
- 3 = 3.0dB
- Other values – Not Supported.</td></tr>
<tr><td>Implemented Products</td><td>All PE Series Amplifiers Inputs (Amplifier Channels Only), Standard, DSP, CobraNet, All NE Series Products (Amplifiers & Signal Processors)</td></tr>
<tr><td>Attributes</td><td>Write Only</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x0A</td><td>Option Gain/ Attenuation Increment & Decrement</td></tr>
<tr><td>2</td><td>4</td><td>Length of data field</td></tr>
<tr><td>3</td><td>aa</td><td>Channel Type (0 = Amp Channel/Output, 1 = Input)</td></tr>
<tr><td>4</td><td>bb</td><td>Channel</td></tr>
<tr><td>5</td><td>cc</td><td>Increment/ Decrement (1 = Inc., 0 = Dec.)</td></tr>
<tr><td>6</td><td>dd</td><td>amount of Change (0-3 = .5dB, 1dB, 2dB, 3dB)</td></tr>
</table>

Note: For PE Series Amplifiers this will change the Attenuation set via Software.</td></tr>
</table>

Name	OPT_STANDBY
Option #	0x0D
Description	Option to read/set the standby state of the amplifier and front-panel disable. When used as part of a "set" message, the length may be either one or two. If the length is one, the front-panel disable state is not sent (and not modified). If the length is two, the front-panel disable state provided is used to set the target device. When used as part of a "get" message, the target device will always send both the standby state and front-panel disable state (thus, the length will be two).
Implemented Products	All PE Series Amplifiers Inputs Standard, DSP, CobraNet NE Amplifiers
Attributes	Read/Write
Byte Description	Byte # / Value / Description 1 / 0x0D / Option Standby 2 / 1 or 2 / Length of following data 3 / xx / Standby state (0 = On, 1 = standby) 4 / yy / OPTIONAL: Front-panel disable state (0 = enabled, 1 = disabled)

Name	OPT_INPUT_CONFIG
Option #	0x0F
Description	Option allows the Input Configuration to be set. This includes the sampling rate as well as the source. Input Configuration - 0 = Analog Input 48Khz - 1 = Analog Input 96Khz - 2 = AES3/EBU Input 44.1Khz - 3 = AES3/EBU Input 48Khz - 4 = AES3/EBU Input 88.2Khz - 5 = AES3/EBU Input 96Khz - 6 = AES3/EBU w/ Analog Backup 48Khz - 7 = AES3/EBU w/ Analog Backup 96Khz
Implemented Products	PE Series DSP Input (NOT Supported in NE Series)
Attributes	Read/Write
Byte Description	Byte # / Value / Description 1 / 0x0F / Option Start Delay 2 / 1 / Length of data field 3 / xx / Input Configuration

Name	OPT_DVCA_LEVEL
Option #	0x10
Description	Option to set the current DVCA Level for a given DVCA. Current products support only 4 DVCA's
Implemented Products	PE DSP Input Card, NE Series.
Attributes	Read/Write
Byte Description	Byte # / Value / Description 1 / 0x10 / Option DVCA Level 2 / 4 / Length of data field 3 / xx / DVCA Num (0-3 = DVCAs 1-4) 4 / yy / DVCA Mute (0 = Unmuted, 1-FF = Muted) 5-6 / xxxx / VCA Gain (-50 to +12 = 7792 to 8312, Off = 0) (.1dB Step size)

Name	OPT_DVCA_LINK_GROUP
Option #	0x11
Description	
Implemented Products	PE DSP Input Card, NE Series
Attributes	Read / Write
Byte Description	

Byte #	Value	Description
1	0x11	Option DVCA Link Group
2	2	Length of data field
3	xx	DVCA Num (0-3 = DVCAs 1-4)
3	yy	DVCA Link Group (0 = None, 1-8 = Link Groups 1-8)

Name	OPT_DVCA_NAME
Option #	0x12
Description	
Implemented Products	PE DSP Input Card, NE Series
Attributes	Read/Write
Byte Description	

Byte #	Value	Description
1	0x12	Option DVCA Name
2	21	Length of data field
3	xx	DVCA Num (0-3 = DVCAs 1-4)
4-23	zz	DVCA Name (20 char length)

Name	OPT_COBRANET_RECEIVER
Option #	0x16
Description	CobraNet Receiver Option will allow each CobraNet Receiver's bundle to be configured. The Bundle specifies which bundle on a CobraNet net the Receiver will request and receive if available. Bundles < 255 are broadcast on network and not normally recommended unless your application needs to send audio to multiple receivers. Receiver Number (byte xx) - 0-3 = Receivers A to D - Not all receivers need be configured only those to be used. - All other receiver numbers ignored. Receiver Bundle (16 bit unsigned integer yy) - Valid Range = 0 to 65535. - 0 = None - 1-255 are Multicast (not recommended) - 256-65279 Unicast (typical usage) - 65280-65535 Private (see CobraNet Programmer's Reference, page 10)
Implemented Products	PE CobraNet Input Card, NE Series
Attributes	Read/Write
Byte Description	

Byte #	Value	Description
1	0x16	Option CobraNet Receiver
2	3	Bytes to Follow
3	xx	Receiver Number
4-5	yy	Receiver Bundle

<table>
<tr><td>Name</td><td>OPT_COBRANET_RECEIVER_MAP</td></tr>
<tr><td>Option #</td><td>0x17</td></tr>
<tr><td>Description</td><td>Option to get/set the Receiver Mapping. This mapping allows a particular Receiver sub Channel to be mapped to a Specific Audio Routing Channel Number. The “Audio Routing Channel Number” Channel on a Protea Device. Typically Audio Routing channels are Mapped
1 to x = Channels 1 to x

Audio Output Channel
- CobraNet Audio Output Channel (ARChannel + 33)
- For PE Amplifier use 0 & 1 as Channels 1 & 2

Receiver Number
- 0 = None Assigned to channel
- 1-8 = Receivers 1 to 8
-
Bundle Sub Index
- 0-7 = Sub Index 1 to 8</td></tr>
<tr><td>Implemented Products</td><td>PE CobraNet Input Card, NE Series</td></tr>
<tr><td>Attributes</td><td>Read/Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x17</td><td>Option CobraNet Receiver Map</td></tr>
<tr><td>2</td><td>3</td><td>Bytes to Follow</td></tr>
<tr><td>3</td><td>aa</td><td>Audio Output Channel</td></tr>
<tr><td>4</td><td>cc</td><td>Receiver Number (0-3 = 1-4)</td></tr>
<tr><td>5</td><td>dd</td><td>Bundle Sub Channel</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_INPUT_SOURCE</td></tr>
<tr><td>Option #</td><td>0x18</td></tr>
<tr><td>Description</td><td>Option to get/set the Audio Source for a particular Channel.

Channel Type (aa)
- Set to 0 for PE Amplifiers
- 0 = Outputs,
- 1 = Inputs

Audio Channel
- 0 & 1 = Channels 1 & 2
- 0 to x-1, where x = number of inputs or outputs in the device.
- Values greater than the number of inputs – 1 are ignored.

Audio Source
- 0 = None
- 1 = Analog
- 2 = AES
- 3 = Network Audio (CobraNet)
- 16 = AES w/ Analog Backup
- 17 = CobraNet w/ Analog Backup
- 32 = Auto Mode (Net, AES, Analog)

Current Audio Source (dd)**
- Supported by PE MultiChannel & NE Rackmount Only
- Only Valid on Reads, Should not be included on Writes
- Will Cause Bytes to Follow to be Adjusted to 4
- Contains the Current Source for the specific channel.
- This is normally the same as the Audio Source Except in Auto Modes. Where this will the selected source.
- Will also be sent with Updates.
- 0 = None, 1 = Analog, 2 = AES3/EBU, 3 = Network Audio</td></tr>
<tr><td>Implemented Products</td><td>PE CobraNet Input Card, NE MultiChannel Amplifiers, NE Rackmount</td></tr>
<tr><td>Attributes</td><td>Read/Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x18</td><td>Option Input Source</td></tr>
<tr><td>2</td><td>3 (4)</td><td>Bytes to Follow **</td></tr>
<tr><td>3</td><td>aa</td><td>Channel Type</td></tr>
<tr><td>4</td><td>Bb</td><td>Audio Input Channel</td></tr>
<tr><td>5</td><td>cc</td><td>Audio Source</td></tr>
<tr><td>6</td><td>dd</td><td>Current Audio Source **</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_LOGIC_BLOCK</td></tr>
<tr><td>Option #</td><td>0x19</td></tr>
<tr><td>Description</td><td>Option to Setup a Logic Block. A logic Block may have several Logic I/O Pins. However, all the Pins in the Logic Block are of the same direction. The length of this message is dependent on the number of pins in the logic I/O Block.

Logic I/O Block Number
- Block Number of the logic I/O block this represents

Logic I/O Block Direction
- A logic I/O Block may be configured either as an input or and output
- 0 = Logic Direction Output
- 1 = Logic Direction Input

Logic IO Pin Function
- 0 = NO Function
- 1 = Input Function preset, or Output Function GPO

Logic IO Pin Parameter
- Parameter defined by the function</td></tr>
<tr><td>Implemented Products</td><td>NE Rackmount</td></tr>
<tr><td>Attributes</td><td>Read/Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x16</td><td>Option Logic IO</td></tr>
<tr><td>2</td><td>10</td><td>Bytes to Follow</td></tr>
<tr><td>3</td><td>aa</td><td>Logic I/O Block Number</td></tr>
<tr><td>4</td><td>bb</td><td>Logic I/O Block Direction</td></tr>
<tr><td>5</td><td>cc</td><td>Logic I/O Pin 1 Function</td></tr>
<tr><td>6</td><td>dd</td><td>Logic I/O Pin 1 Parameter</td></tr>
<tr><td>7</td><td>cc</td><td>Logic I/O Pin 2 Function</td></tr>
<tr><td>8</td><td>dd</td><td>Logic I/O Pin 2 Parameter</td></tr>
<tr><td>9</td><td>cc</td><td>Logic I/O Pin 3 Function</td></tr>
<tr><td>10</td><td>dd</td><td>Logic I/O Pin 3 Parameter</td></tr>
<tr><td>11</td><td>cc</td><td>Logic I/O Pin 4 Function</td></tr>
<tr><td>12</td><td>dd</td><td>Logic I/O Pin 4 Parameter</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_COBRANET_TX</td></tr>
<tr><td>Option #</td><td>0x1A</td></tr>
<tr><td>Description</td><td>CobraNet Transmitter Option will allow each CobraNet Transmitter's bundle to be configured. The Bundle specifies which bundle on a CobraNet net the Bundle will send audio data. This will also allow the user to specify the format for each sub bundle.

Bundles < 255 are Broadcast and not recommended.

Transmitter Number (byte xx)
- 0-3 = Receivers A to D
- Not all receivers need be configured only those to be used.
- All other receiver numbers ignored.

TX Bundle (16 bit unsigned integer yy)
- Valid Range = 0 to 65535.
- 0 = None
- 1-255 are Multicast (not recommended)
- 256-65279 Unicast (typical usage)
- 65280-65535 Private (see CobraNet Programmer's Reference pg10)

TX Format (8bit unsigned integer zz)
- 0x04 = 16 Bit
- 0x05 = 20 Bit
- 0x06 = 24 Bit</td></tr>
<tr><td>Implemented Products</td><td>NE Rackmount</td></tr>
<tr><td>Attributes</td><td>Read/Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x1A</td><td>Option CobraNet Transmitter</td></tr>
<tr><td>2</td><td>4</td><td>Bytes to Follow</td></tr>
<tr><td>3</td><td>aa</td><td>Tx Number</td></tr>
<tr><td>4-5</td><td>bb</td><td>Tx Bundle</td></tr>
<tr><td>6</td><td>cc</td><td>Bundle Tx Format</td></tr>
</table>
</td></tr>
</table>

Name	OPT_COBRANET_TX_MAP
Option #	0x1B
Description	Option to get/set the Transmitter Mapping. This mapping allows a particular Transmitter sub Channel to be mapped to a Specific Audio Routing Channel Number. The "Audio Routing Channel Number" Channel on a Protea Device. Typically Audio Routing channels are Mapped 1 to x = Channels 1 to x Audio Input Channel - CobraNet Audio Input Channel (ARChannel) - For Ne Rackmount use 1-8 Transmitter Number - 0 = None Assigned to channel - 1-8 = Transmitter 1 to 8 Bundle Sub Index - 0-7 = Sub Index 1 to 8
Implemented Products	NE Rackmount
Attributes	Read/Write
Byte Description	Byte # | Value | Description 1 | 0x1B | Option CobraNet Transmitter Map 2 | 3 | Bytes to Follow 3 | aa | Audio Input Channel 4 | cc | Transmitter Number (0-3 = 1-4) 5 | dd | Bundle Sub Channel

Name	OPT_LOGIC_PIN
Option #	0x1C
Description	Option to allow a single Pin to be modified w/o specifying the other pins in the bank. Pin Number - logic pin number relative to all pins in the box. (in block 2 pin 1 would be pin 5) Pin State - 0 = OFF, 1 = ON
Implemented Products	NE Series (with logic outputs)
Attributes	Write Only
Byte Description	Byte # | Value | Description 1 | 0x1C | Option Logic Pin 2 | 2 | Bytes to Follow 3 | aa | Pin Number 4 | cc | Pin State

<table>
<tr><td>Name</td><td>OPT_FR_CONFIG</td></tr>
<tr><td>Option #</td><td>0x25</td></tr>
<tr><td>Description</td><td>Option to configure system settings for FR remote.

aa – Brightness Level
0 = Quarter
1 = Half
2 = Three Quarter
3 = Full

bb – Faders Present / Lockout Status (read-only)
bits 0..6 = number of controllable faders present (8 or 16)
bit 7 = status of lockout switch
This is a read-only byte and is ignored on write.

cc, dd – Target User Name and Password
These null-padded strings are sent to all devices being controlled by the FR.</td></tr>
<tr><td>Implemented Products</td><td>FR Remotes</td></tr>
<tr><td>Attributes</td><td>Read/Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x25</td><td>OPT_FR_CONFIG</td></tr>
<tr><td>2</td><td>19</td><td>Bytes to Follow</td></tr>
<tr><td>3</td><td>0x00</td><td>(reserved for future use)</td></tr>
<tr><td>4</td><td>aa</td><td>Brightness Level</td></tr>
<tr><td>5</td><td>bb</td><td>Faders Present/Lockout Status</td></tr>
<tr><td>6 ..13</td><td>cc</td><td>Target User Name</td></tr>
<tr><td>14 .. 21</td><td>dd</td><td>Target Password</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_FR_FADER_CONFIG</td></tr>
<tr><td>Option #</td><td>0x26</td></tr>
<tr><td>Description</td><td>Option for per-fader configuration for FR remotes.

aa – Fader Number (zero-based)

bb – Fader Mode
0 = Disabled
1 = Mixer
2 = I/O Level
3 = Source Select

cc – Target MAC address

dd – Channel Type
0 = Output
1 = Input

ee – Channel Number (zero-based)

ff – Sub-Channel A
Ignored in I/O Level mode.
Specifies the mixer channel being controlled in Mixer mode.
Specifies the “A” channel in Source Select mode.

gg – Master Enable
0 = Master fader does not affect this channel
1 = Channel's fader is affected by Master fader

hh – Fader Scaling Low
ii – Fader Scaling High
The range in dB of the channel's fader, from low to high.
Expressed as 16-bit value: (dB * 10) + 8192
Range is 7692 (-50dB) to 8312 (+12dB).

jj – Meter Threshold Low
kk – Meter Threshold High
In both Mixer and I/O Level modes, these are the points at which the target device's audio levels light the LED green (low threshold) or orange (high threshold).
Expressed as 16-bit value: (dBu * 10) + 8192
Range is 7892 (-30dBu) to 8392 (+20dBu).

ll – Sub-Channel B
Ignored in I/O Level and Mixer modes.
Specifies the “B” channel in Source Select mode.</td></tr>
<tr><td>Implemented Products</td><td>FR Remotes</td></tr>
<tr><td>Attributes</td><td>Read/Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x26</td><td>OPT_FR_FADER_CONFIG</td></tr>
<tr><td>2</td><td>21</td><td>Bytes to Follow</td></tr>
<tr><td>3</td><td>aa</td><td>Fader Number</td></tr>
<tr><td>4</td><td>bb</td><td>Fader Mode</td></tr>
<tr><td>5 .. 10</td><td>cc</td><td>Target MAC Address</td></tr>
<tr><td>11</td><td>dd</td><td>Channel Type</td></tr>
<tr><td>12</td><td>ee</td><td>Channel Number</td></tr>
<tr><td>13</td><td>ff</td><td>Sub-Channel A</td></tr>
<tr><td>14</td><td>gg</td><td>Master Enable</td></tr>
<tr><td>15 .. 16</td><td>hh</td><td>Fader Scaling Low</td></tr>
<tr><td>17 .. 18</td><td>ii</td><td>Fader Scaling High</td></tr>
<tr><td>19 .. 20</td><td>jj</td><td>Meter Threshold Low</td></tr>
<tr><td>21 .. 22</td><td>kk</td><td>Meter Threshold High</td></tr>
<tr><td>23</td><td>ll</td><td>Sub-Channel B</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_RTC</td></tr>
<tr><td>Option #</td><td>0x2D</td></tr>
<tr><td>Description</td><td>Option to adjust the RTC Settings For the current time of the amplifier

AA – Adjust for DST,
- 0x00 – Do not automatically adjust for DST (Daylight savings time)
- 0x01 – Automatically adjust for DST

BB – Current Year (0 .. 99 = 2000 to 2099)

CC – Current Month (1 .. 12 = January to December)

DD – Current Date (1 .. 31, depending on month)

EE – Current Hour (0 .. 23, 0 = Midnight)

FF – Current Minute (00 – 59)

GG – Current Second (00 – 59)

HH – Current Day of Week (1 .. 7 = Monday to Sunday)</td></tr>
<tr><td>Implemented Products</td><td>NE Series Products w/ RTC Events</td></tr>
<tr><td>Attributes</td><td>Read/Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x2D</td><td>Option RTC</td></tr>
<tr><td>2</td><td>8</td><td>Bytes to Follow</td></tr>
<tr><td>3</td><td>AA</td><td>Adjust for DST</td></tr>
<tr><td>4</td><td>BB</td><td>Year</td></tr>
<tr><td>5</td><td>Cc</td><td>Month</td></tr>
<tr><td>6</td><td>Dd</td><td>Day</td></tr>
<tr><td>7</td><td>Ee</td><td>Hour</td></tr>
<tr><td>8</td><td>Ff</td><td>Minute</td></tr>
<tr><td>9</td><td>Gg</td><td>Second</td></tr>
<tr><td>10</td><td>HH</td><td>Day</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_RTC_EVENT</td></tr>
<tr><td>Option #</td><td>0x2E</td></tr>
<tr><td>Description</td><td>
Option to adjust the settings of a specific RTC event.

AA – Event Number (0 .. 99)

BB – Event Name (20 characters, zero-padded)

CC – Event Type
- 0 = None
- 1 = Preset Recall
- 2 = Power State Change
- 3 = Mute Change
- 4 = Source Change
- 5 = Level Change

DD – Event Hour (0 .. 23, 0 = Midnight)

EE – Event Minute (0 to 59)

FF – Event Day Mask
- Bitwise mask of days this event will run:
<table>
<tr><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>day</td><td>N/A</td><td>Sun</td><td>Sat</td><td>Fri</td><td>Thur</td><td>Wed</td><td>Tue</td><td>Mon</td></tr>
</table>
GG – Channel Type (0 = output, 1 = input)
HH – Channel (0 .. max channel -1)
- Not used on all events.

II, JJ - Event Parameter 1 and 2:
<table>
<tr><td>Parameter 1</td><td>Description</td><td>Parameter 2</td></tr>
<tr><td>0</td><td>None</td><td>N/A</td></tr>
<tr><td>1</td><td>Preset recall</td><td>0 .. max preset-1</td></tr>
<tr><td>2</td><td>Power Change</td><td>0 = normal (power on)
1 = standby</td></tr>
<tr><td>3</td><td>Mute Change</td><td>0 = unmuted
1 = muted</td></tr>
<tr><td>4</td><td>Source Change</td><td>0 = source enabled
1 = source disabled</td></tr>
<tr><td>5</td><td>WR5 Level Change</td><td>0 = Mute (full attenuation)
1 .. 99 = -49.0dB to
0.0dB" (half dB steps)</td></tr>
</table>
</td></tr>
<tr><td>Implemented Products</td><td>NE Series Products w/ RTC Events</td></tr>
<tr><td>Attributes</td><td>Read/Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x2E</td><td>Option RTC Event</td></tr>
<tr><td>2</td><td>35</td><td>Bytes to Follow</td></tr>
<tr><td>3</td><td>AA</td><td>Event Number</td></tr>
<tr><td>4-23</td><td>Bb</td><td>Event Name</td></tr>
<tr><td>24</td><td>Cc</td><td>Event Type</td></tr>
<tr><td>25</td><td>Dd</td><td>Hour</td></tr>
<tr><td>26</td><td>Ee</td><td>Minute</td></tr>
<tr><td>27</td><td>Ff</td><td>Day Mask</td></tr>
<tr><td>28</td><td>Gg</td><td>Chan Type</td></tr>
<tr><td>29</td><td>HH</td><td>Chan</td></tr>
<tr><td>30-33</td><td>II</td><td>Event Param 1</td></tr>
<tr><td>34-37</td><td>JJ</td><td>Event param 2</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_FAULT_LOG</td></tr>
<tr><td>Option #</td><td>0x2F</td></tr>
<tr><td>Description</td><td>Option to Access the Fault Log (and also clear it with a special key)

AA – Fault Number
- 16 bit fault number 0 to Max Faults
- 0xFFAA is reserved to indicate Fault Log Clear.
- If 0xFFAA is sent to the device the fault log will clear
- If 0xFFAA is received via update this indicates the fault log was cleared.

BB – Fault Type
- Value indicates the type of fault
- 0 = None – No data is sent with a none type
- 1 = Amplifier Channel Thermal
- 2 = Amplifier Channel Protect fault
- 3 = Amplifier Supply Protect fault
- 4 = Amplifier Fuse Protect fault
- 5 = Amplifier Rail protect Fault
- 6 = Amplifier Power On/Off Event
- 7 = DSP Fault
- 8 = Memory Fault
- 9 = Network Fault
- 10 = Network Audio Fault
- 11 = Host microcontroller Fault
- 12 = Security Change
- 13 = Time Change

CC – Fault Time
- 1st byte is year (00 – 99),
- 2nd byte is Month (1-12)
- 3rd byte is Date (1-31)
- 4th byte is Hour (0-23)
- 5th byte is Minute (0-59)
- 6th byte is Second (0-59)

DD – Fault Channel
- 0 to Maximum Input Channels + Maximum Output Channels
- < Maximum Input Channel value is input channel
- else value – max input is output channel number.
- For some faults this indicates other values such as DSP number

EE – Event Description
- String description

FF – Data Length
- Bytes in data field

GG – Data
- Data sent with fault. This varies by fault type.</td></tr>
<tr><td>Implemented Products</td><td>NE Series Products w/ Fault Logs</td></tr>
<tr><td>Attributes</td><td>Read/Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x2F</td><td>Option Fault Log</td></tr>
<tr><td>2</td><td>xx</td><td>Bytes to Follow, depends on Fault and data</td></tr>
<tr><td>3-4</td><td>AA</td><td>Fault Number</td></tr>
<tr><td>5</td><td>Bb</td><td>Fault Type</td></tr>
<tr><td>6-11</td><td>Cc</td><td>Event Time</td></tr>
<tr><td>12</td><td>Dd</td><td>Event Channel (Host Channel Numbering)</td></tr>
<tr><td>13-42</td><td>Ee</td><td>Event Description</td></tr>
<tr><td>43</td><td>Ff</td><td>Data Length</td></tr>
<tr><td>44 - ??</td><td>Gg</td><td>Data</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_PRESET_INFO</td></tr>
<tr><td>Option #</td><td>0x30</td></tr>
<tr><td>Description</td><td>Option Preset Info allows a user to access the Name of a given preset. This option may also be used to access and modify the “Working Preset” Name.

The working Preset is the current settings in the device.

Preset Number (byte xx)
- The Preset Number to Access
- 1 to X = Presets 1 to X where X is the maximum number of presets
- 0x00 = Working Preset (or working Settings)
- 0xFF = Temp Preset
- Invalid Values are ignored

Preset Name (string zz)
- 20 Character String
- Left Justified
- Fill unused Characters with 0x00
- Valid Characters are 0x00 & 0x20 to 0x7A

Sub Preset (cc)*
- 0 = Not a Sub Preset
- 1 –255 Sub Preset
- Only Valid on PE Multi Channel Amps, NE Rackmount and other products that support Sub Presets.</td></tr>
<tr><td>Implemented Products</td><td>PE DSP Input Card, PE MultiChannel Amp, NE Rackmount</td></tr>
<tr><td>Attributes</td><td>Read/Write for working & temp Preset, Read Only for all other Preset Numbers</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x30</td><td>Option Preset Info</td></tr>
<tr><td>2</td><td>21</td><td>Bytes to Follow</td></tr>
<tr><td>3</td><td>xx</td><td>Preset Number</td></tr>
<tr><td>4-23</td><td>zz</td><td>Preset Name</td></tr>
<tr><td>24</td><td>cc</td><td>SubPreset*</td></tr>
</table>

* - when sub-preset bytes to follow = 22</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_RECALL_PRESET</td></tr>
<tr><td>Option #</td><td>0x32</td></tr>
<tr><td>Description</td><td>Option to recall a preset stored in the Protea NE Unit. This preset may be either a full preset or a sub preset.

Preset Number (byte xx)
- 1 to x = presets 1 to x
- x = maximum number of presets supported

Mute Outputs (byte yy)
- 0 = outputs will not be muted unless directed by preset
- 1-0xFF = outputs will be muted regardless of what is in the preset</td></tr>
<tr><td>Implemented Products</td><td>PE DSP Input Card</td></tr>
<tr><td>Attributes</td><td>Write Only (Reads are Ignored)</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x31</td><td>Option Recall Preset</td></tr>
<tr><td>2</td><td>2</td><td>Bytes to Follow</td></tr>
<tr><td>3</td><td>xx</td><td>Preset Number</td></tr>
<tr><td>4</td><td>yy</td><td>Mute</td></tr>
</table>
</td></tr>
</table>

Name	OPT_SAVE_PRESET
Option #	0x31
Description	Option to save the current settings to a preset location. The settings to be saved may either come from the Working Preset aka Working Settings or the may come from the Temp Preset. Preset Number (byte xx) - 0 to x – 1 = presets 1 to x - x = maximum number of presets supported Source (byte yy) - 0 = Working Preset, all data in the working Preset will be saved - 1 = Temp Preset, all data in the Temp Preset will be saved
Implemented Products	PE DSP Input Card
Attributes	Write Only (Reads are Ignored)
Byte Description	Byte # / Value / Description 1 / 0x32 / Option Save Preset 2 / 2 / Bytes to Follow 3 / xx / Preset Number 4 / yy / Source

Name	OPT_DEVICE_MODIFIED
Option #	0x33
Description	(Request All parameters again)
Implemented Products	Protea DSP Input Card & Protea CobraNet Input Card
Attributes	PC Receive Only, Can not Read or Write
Byte Description	Byte # / Value / Description 1 / 0x33 / Option Save Preset 2 / 0 or 1 / Length of data field (if Preset Recall use 1, else 0) 3 * / xx / Preset Number Recalled (Optional) * Adjust length to 1 if optional argument present

Name	OPT_TOTAL_PRESETS
Option #	0x34
Description	Option to obtain the total number of presets supported by a device. Not supported by all products Sub Presets Supported: (aa) - 0 = Sub Presets are not supported - 1-255 = Sub Presets are supported Total Presets Available: (bb) - 0 = None, - 1- 255 = Total Number of Presets available.
Implemented Products	Protea NE Rackmount, PE MultiChannel.
Attributes	PC Receive Only, Can not Read or Write
Byte Description	Byte # / Value / Description 1 / 0x34 / Option Save Preset 2 / 3 / Length of data field (if Preset Recall use 1 else 0) 3 / aa / Sub Presets Supported 4 / bb / Total Presets Available. 5 / Cc / Current Working Preset Number* * Supported by products that support neWR5

<table>
<tr><td>Name</td><td>OPT_METER_INPUT</td></tr>
<tr><td>Option #</td><td>0x40</td></tr>
<tr><td>Description</td><td>The meter Input option can be used to Analog Inputs to a device. For example this meter can be used to measure the input to the PE Series Amplifiers. Note: PE Series DSP Inputs should use the DSP Input Meter option.

*Channel Type (byte xx)
- 0 = Amp Channel (only supported value)

*Channel (byte yy)
- Amplifier Channel 0 = Channel 1, 1 = Channel 2

Clip (zz)
- 0 = Signal Not Clipping
- 1 = Signal Clipping
- There is a Clip hold on the clip value

Input Meter
-

* Required for Valid Request</td></tr>
<tr><td>Implemented Products</td><td>PE Series Standard Input Card & CobraNet Input Card</td></tr>
<tr><td>Attributes</td><td>Read Only</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x40</td><td>Option Meter Input</td></tr>
<tr><td>2</td><td>4</td><td>Bytes to follow</td></tr>
<tr><td>3</td><td>xx</td><td>Channel Type</td></tr>
<tr><td>4</td><td>xx</td><td>Channel</td></tr>
<tr><td>5</td><td>yy</td><td>Clip</td></tr>
<tr><td>6</td><td>zz</td><td>Input Meter</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_METER_OUTPUT</td></tr>
<tr><td>Option #</td><td>0x41</td></tr>
<tr><td>Description</td><td>The Output meter option allows the output level of the amplifier to be metered (in dB Down from clip).

*Channel Type:
- 0 = Amp Channel
- 1-0xFF are invalid

*Channel:
- 0 & 1 = Amp Channel 1 & 2
- 2 – 0xFF are invalid

Clip:
- 0 Channel Not Clipping
- 1 Channel is Clipping
- 2-0xFF not used
- (Clip hold of approximately 0.25 sec)

Output Meter:
- 0 to 40 = dB down from Clip
- 1dB Steps
- 41 to 255 are not used

* Required for a valid Request.</td></tr>
<tr><td>Implemented Products</td><td>All PE Series Input Cards</td></tr>
<tr><td>Attributes</td><td>Read Only</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x41</td><td>Option Meter Output</td></tr>
<tr><td>2</td><td>4</td><td>Length of data field</td></tr>
<tr><td>3</td><td>xx</td><td>Channel Type: 0 = Amp Channel;</td></tr>
<tr><td>4</td><td>xx</td><td>Channel</td></tr>
<tr><td>5</td><td>yy</td><td>Clip (0 = not Clipped, 1-FF = Clipped)</td></tr>
<tr><td>6</td><td>yy</td><td>Output Meter</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_METER_TEMP</td></tr>
<tr><td>Option #</td><td>0x42</td></tr>
<tr><td>Description</td><td>Option to access the current temperature for an Amplifier Channel. This option returns a value between 0 and 1023. 0 indicates 0 deg C, while 1023 indicates 94 deg C (with a linear scale in between).

Channel Type (aa)
- 0 = Amp Channel
- All other values currently Invalid

Channel (bb)
- 0 to (Maximum Channels available – 1)

Temp Meter (cc)
- Temp value between 0 and 1023
- 0 = 0degC
- 1023 = 94 deg C</td></tr>
<tr><td>Implemented Products</td><td>All PE Series Amplifier Inputs (Std, DSP, CobraNet)</td></tr>
<tr><td>Attributes</td><td>Read Only</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x42</td><td>Option Meter Temp</td></tr>
<tr><td>2</td><td>4</td><td>Length of data field</td></tr>
<tr><td>3</td><td>aa</td><td>Channel Type: 0 = Amp Channel</td></tr>
<tr><td>4</td><td>bb</td><td>Channel</td></tr>
<tr><td>5- 6</td><td>cc</td><td>Temp Meter</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_METER_CURRENT</td></tr>
<tr><td>Option #</td><td>0x43</td></tr>
<tr><td>Description</td><td>Option to access the Current of a PE Series Amplifier. This option returns a current value between 0 and 1023.

Channel Type (aa)
- 0 = Amp Channel
- All other values currently Invalid

Channel (bb)
- 0 to (Maximum Channels available – 1)

Current Meter (cc)
- Current value between 0 and 1023</td></tr>
<tr><td>Implemented Products</td><td>All PE Series Inputs (Std, DSP, CobraNet)</td></tr>
<tr><td>Attributes</td><td>Read Only</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x43</td><td>Option Meter Current</td></tr>
<tr><td>2</td><td>4</td><td>Length of data field</td></tr>
<tr><td>3</td><td>aa</td><td>Channel Type: 0 = Amp Channel</td></tr>
<tr><td>4</td><td>bb</td><td>Channel</td></tr>
<tr><td>5-6</td><td>cc</td><td>Current Meter</td></tr>
</table>
</td></tr>
</table>

Name	OPT_METER_ATTENS
Option #	0x45
Description	Option to return the current Attenuation values of the Front and Rear Attenuators. This allows software to know how much the amp is attenuating the signal. Channel Type (aa) - 0 = Amp Channel - All other values currently Invalid Channel (bb) - 0 to (Maximum Channels available – 1) Front Panel Attenuation & Rear Panel Attenuation (cc) - Amount of front panel attenuation (in dB) - 0 = 0 dB 40 = 40dB of Attenuation - 41-255 = MUTE (full attenuation)
Implemented Products	All PE Series Amps (Std, DSP, CobraNet)
Attributes	Read Only
Byte Description	Byte # / Value / Description 1 / 0x45 / Option Meter Attenuators 2 / 4 / Length of data field 3 / aa / Channel Type (Not used, set to 0) 4 / bb / Channel 5 / cc / Front Panel Attenuation 6 / cc / Rear Panel Attenuation

Name	OPT_CHANNEL_PROTECT
Option #	0x46
Description	Channel Protect option allows the current “Protect” Status of an amplifier channel to be read. If either channel is in protect, the respective front panel Mute Led will light and the protect led will light. Note that while the amp is in standby, the protect status is always false. Channel Type (aa) - 0 = Amp Channel - All other values currently Invalid Channel (bb) - 0 to (Maximum Channels available – 1) Protect (cc) - 0 = Not in Protect - 1 to 0xFF = Channel in Protect
Implemented Products	All PE Series Amps (Std, DSP, CobraNet)
Attributes	Read Only
Byte Description	Byte # / Value / Description 1 / 0x46 / Option Meter Attenuators 2 / 3 / Length of data field 3 / xx / Channel Type (Not used, set to 0) 4 / xx / Channel 5 / xx / Protect (1 = channel in protect)

Name	OPT_AES_RECEIVER
Option #	0x47
Description	Option AES Receiver reports the current status for a given AES Receiver. This includes current locked frequency as well as any current error Codes. AES Receiver Number - 0 for PE Series Amplifiers - All other values ignored. AES Frequency - Valid Frequencies - 44K1=0 - 48K=2 - 32K=3 - 22K=4 - 11K=5 - 24K=6 - 16K=7 - 88K2=8 - 8K=9 - 96K=10 - 64K=11 - 176K4=12 - 192K=14 - INVALID=15-255 AES Error Code (bitwise value) - ERROR_PARITY = 0x01 - ERROR_NON_AUD = 0x02 - ERROR_FREQ_CHNG = 0x04 - ERROR_INVALID = 0x08 - ERROR_UNLOCK = 0x10 - ERROR_FREQ_MISMATCH = 0x20 - ERROR_NON_PCM = 0x40 - ERROR_MCLK_FAIL = 0x80 - ERROR_NONE = 0x00
Implemented Products	PE DSP Input Card, PE Multi, NE Rackmount
Attributes	(Read Only)
Byte Description	Byte # / Value / Description 1 / 0x47 / Option AES3 Receiver Status 2 / 3 / Bytes To Follow 3 / yy / AES Receiver Num (0 for amplifier) 4 / xx / AES Freq 5 / xx / AES Error Code

Name	OPT_LOGIC_STATE
Option #	0x48
Description	Option to return the current Pin State for a Logic IO Block. The Pin State ignores the current pin Direction
Implemented Products	NE Rackmount
Attributes	(Read Only)
Byte Description	Byte # / Value / Description 1 / 0x48 / Option LOGIC State 2 / 5 / Bytes To Follow 3 / yy / Logic IO Block Number 4 / xx / Pin 1 State 5 / xx / Pin 2 State 6 / xx / Pin 3 State 7 / xx / Pin 4 State

Name	OPT_WORDCLOCK_STATUS
Option #	0x49
Description	Option to return current status of the word clock input.
Implemented Products	NE Rackmount
Attributes	(Read Only)
Byte Description	Byte # / Value / Description 1 / 0x49 / Option word clock status 2 / 2 / Bytes To Follow 3 / yy / Lock Status 4 / xx / Detected Sampling Rate

Name	OPT_AMP_METER_PROTECT
Option #	0x4A
Description	Option to access the current protect status of an amplifier (NE Series).
Implemented Products	NE Amplifiers
Attributes	Read Only
Byte Description	Byte # / Value / Description 1 / 0x4A / Option User Info Identifier 2 / 5 / Length of data field 3 / aa / Channel Protect 4 / bb / Channel Thermal 5 / bb / Fuse Protect 6 / bb / Rail Fault 7 / bb / Supply Protect

Name	OPT_AMP_GAIN_SETTING
Option #	0x4B
Description	Option to access the current gain switch of an amplifier (NE Series).
Implemented Products	NE Amplifiers
Attributes	Read Only
Byte Description	Byte # / Value / Description 1 / 0x4A / Option User Info Identifier 2 / 1 / Length of data field 3 / aa / Gain switch setting

<table>
<tr><td>Name</td><td>OPT_REMOTE_RD8C</td></tr>
<tr><td>Option #</td><td>0x50</td></tr>
<tr><td>Description</td><td>This option works with both the RD8C and with the FR-8/16.

RD8C:
- The option length is 9 bytes.
- The fader positions are returned in bytes 4 through 11, as described below.

FR-8/16:
- The option length is 21 bytes.
- The first eight fader positions are returned in bytes 4 through 11, with the second eight faders positions in bytes 12 through 19.
- The buttons on the FR-8/16 are treated as a bit array.

Present: (aa)
- 0 = Not Present
- 1-0xFF = Present (always present for FR-8/16)

Level: (bb)
- Fader Attenuation Level in dB
- 0 to124= 0 to 62dB in 0.5dB increments.
- Channels 9 through 16 are 0 in the case of a FR-8

“Master” button: (cc)
- 0 = Disengaged
- 1 = Engaged

Channel buttons: (dd, ee)
- The individual bits represent the state of the channel's button (0=Disengaged, 1=Engaged):
<table>
<tr><td></td><td>bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>dd:</td><td>channel</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>ee:</td><td>channel</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td></tr>
</table></td></tr>
<tr><td>Implemented Products</td><td>RD8C, FR-8/16</td></tr>
<tr><td>Attributes</td><td>Read Only</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x50</td><td>Option User Info Identifier</td></tr>
<tr><td>2</td><td>9 or 21</td><td>Length of data field</td></tr>
<tr><td>3</td><td>aa</td><td>RD8C or FR-8/16 Present</td></tr>
<tr><td>4</td><td>bb</td><td>RD8C or FR-8/16 Level 1</td></tr>
<tr><td>5</td><td>bb</td><td>RD8C or FR-8/16 Level 2</td></tr>
<tr><td>6</td><td>bb</td><td>RD8C or FR-8/16 Level 3</td></tr>
<tr><td>7</td><td>bb</td><td>RD8C or FR-8/16 Level 4</td></tr>
<tr><td>8</td><td>bb</td><td>RD8C or FR-8/16 Level 5</td></tr>
<tr><td>9</td><td>bb</td><td>RD8C or FR-8/16 Level 6</td></tr>
<tr><td>10</td><td>bb</td><td>RD8C or FR-8/16 Level 7</td></tr>
<tr><td>11</td><td>bb</td><td>RD8C or FR-8/16 Level 8</td></tr>
<tr><td>12</td><td>bb</td><td>FR-16 Level 9</td></tr>
<tr><td>13</td><td>bb</td><td>FR-16 Level 10</td></tr>
<tr><td>14</td><td>bb</td><td>FR-16 Level 11</td></tr>
<tr><td>15</td><td>bb</td><td>FR-16 Level 12</td></tr>
<tr><td>16</td><td>bb</td><td>FR-16 Level 13</td></tr>
<tr><td>17</td><td>bb</td><td>FR-16 Level 14</td></tr>
<tr><td>18</td><td>bb</td><td>FR-16 Level 15</td></tr>
<tr><td>19</td><td>bb</td><td>FR-16 Level 16</td></tr>
<tr><td>20</td><td>bb</td><td>FR-8/16 “Master” Level</td></tr>
<tr><td>21</td><td>cc</td><td>FR-8/16 “Master” Button</td></tr>
<tr><td>22</td><td>dd</td><td>FR-8/16 Buttons 1 through 8</td></tr>
<tr><td>23</td><td>ee</td><td>FR-16 Buttons 9 through 16</td></tr>
</table>
</td></tr>
</table>

Name	OPT_REMOTE_LEV
Option #	0x53
Description	Option to access the current Rear Panel Attenuators Rear Level: (bbbb) - Fader Attenuation Level in dB - 0 to102 = 0 to 51dB in 0.5dB increments. - Anything over 50 is considered Mute
Implemented Products	NE Rackmount
Attributes	Read Only
Byte Description	Byte # / Value / Description 1 / 0x50 / Option User Info Identifier 2 / 9 / Length of data field 3 / bb / Rear Level 1 4 / bb / Rear Level 2 5 / bb / Rear Level 3 6 / bb / Rear Level 4 7 / bb / Rear Level 5 8 / bb / Rear Level 6 9 / bb / Rear Level 7 10 / bb / Rear Level 8

Name	OPT_DEVICE_NAME
Option #	0x6E
Description	Option to access/Modify the Name given to a particular Device. This name is user definable so that any device in the system is easily identifiable. Valid characters are 0x00 and 0x20 to 0x7A. The Name is 20 characters Long. Unused Characters should be filled with 0x00.
Implemented Products	All Protea NE Products
Attributes	Read/Write
Byte Description	Byte # / Value / Description 1 / 0x6E / Option Device Name 2 / 20 / Length of data field 3-23 / zz / Device Name

Name	OPT_DEVICE_GROUP
Option #	0x6F
Description	Option to access/Modify the Group Name given to a particular Device. This name is used in PneS to group Devices into folders. This is purely for display purposes only. Valid characters are 0x00 and 0x20 to 0x7A. The Name is 20 characters Long. Unused Characters should be filled with 0x00.
Implemented Products	All Protea NE Products
Attributes	Read/Write
Byte Description	Byte # / Value / Description 1 / 0x6F / Option Device Group 2 / 20 / Length of data field 3-23 / zz / Device Group (valid range 0x00, 0x20-0x7A)

<table>
<tr><td>Name</td><td>OPT_ILL_FRONT</td></tr>
<tr><td>Option #</td><td>0x70</td></tr>
<tr><td>Description</td><td>Special Option to allow a NE Device to be identified in a rack. When this option is sent to a device the COM Led will turn on and remain lit for a short amount of time, a few seconds. This option can not be read.

Illuminate Front Panel: (zz)
- 0x00 – Do nothing
- 0x01 to 0xFF – Turn on COM Led</td></tr>
<tr><td>Implemented Products</td><td>All PE Series Amp Inputs</td></tr>
<tr><td>Attributes</td><td>Write Only</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x70</td><td>Option Illuminate front panel (COM)</td></tr>
<tr><td>2</td><td>1</td><td>Length of data field</td></tr>
<tr><td>3</td><td>zz</td><td>Illuminate Front Panel</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_DEVICE_INFO</td></tr>
<tr><td>Option #</td><td>0x71</td></tr>
<tr><td>Description</td><td>Option to read the device information for a particular device. This information includes the Device Model Type and also the Firmware Revision:

Device Type aa
- Model Type of the Product
- 0x06 = ne24.24M
- 0x08 = Standard PE Series Amp Input
- 0x09 = DSPPE Series Amp Input
- 0x0B = CobraNet PE Series Amp Input
- 0x0C = WR5 Active Wall Remote
- 0x0E = NE Rackmount*
- 0x10 = NE Multichannel Amplifiers (4250,4070,8250,8070)*

Firmware Revision yz
- b.c if b = 1, c = 2 then firmware Revision = 1.2

Hardware Options*
- The Products with an * have Hardware Options that may be installed.
- These are bitwise flags indicating if the option is installed. 1 = Installed.

Model (ee) **

NE MultiChannel Amplifier Hardware Options
0x01 – 8 Channel Amplifier (0 = 4 Channel Amplifier)
0x02 – DSP1 Installed (Channels 1-4)
0x04 – DSP2 Installed (Channels 5-8)
0x08 – CobraNet Installed
0x10 – AES Inputs Installed

ne24.24M Hardware Options
0x01 – Inputs 5-8 Installed
0x02 – Inputs 9-12 Installed
0x04 – Inputs 13-16 Installed
0x08 – Inputs 17-20 Installed
0x10 – Outputs 5-8 Installed
0x20 – Outputs 9-12 Installed
0x40 – Outputs 13-16 Installed
0x80 – Outputs 17-20 Installed.
0x0100 – Output Logic Card Installed.

NE Rackmount Hardware Options
0x08 – CobraNet Installed
0x10 – AES Inputs Installed
0x20 – AES Outputs Installed

NE Rackmount Models
0x00 – Model 4x4
0x01 – Model 4x8
0x02 – Model 8x8
0xFF – Unknown Model</td></tr>
<tr><td>Implemented Products</td><td>All Protea NE Products</td></tr>
<tr><td>Attributes</td><td>Read Only</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x71</td><td>Option User Info Identifier</td></tr>
<tr><td>2</td><td>2</td><td>Length of data field</td></tr>
<tr><td>3</td><td>xx</td><td>Device Type</td></tr>
<tr><td>4</td><td>bc</td><td>Firmware Revision</td></tr>
</table>

* Products w/ Hardware Options (ME Multi)</td></tr>
</table>

Byte #	Value	Description
1	0x71	Option User Info Identifier
2	0x03	Length of data field
3	aa	Device Type
4	bc	Firmware Revision
5	dd	Hardware Options Installed.

* Products w/ Hardware Options & Model (NE Rackmount and newer)

Byte #	Value	Description
1	0x71	Option User Info Identifier
2	3	Length of data field
3	aa	Device Type
4	bc	Firmware Revision
5	dd	Hardware Options Installed.
6	ee	Model

Name	OPT_COBRANET_INFO
Option #	0x78
Description	Option to access information about the CM2 Card or CobraNet Chip Installed. This includes firmware information, as well as the Mac address and Description. All of which is read only. Firmware *: - Contains Firmware Number as reported by CobraNet Chip Mac: - Contains 6 byte, Globally Unique Identifier for CobraNet Chip. - Note there is only 1 per CobraNet Card, the two ports are redundant. Description: - Firmware Description, as reported by the CobraNet Chip.
Implemented Products	PE Amp CobraNet Input Card
Attributes	Read Only
Byte Description	

Byte #	Value	Description
1	0x78	Option CobraNet Info
2	??	Length of data field (length of desc. + 9)
3	xx	Firmware Major
4	xx	Firmware Minor
5	xx	Firmware Protocol
6-11	yy	MAC
12-??	zz	Description

Name	OPT_COBRANET_STATUS
Option #	0x79
Description	Option to access the current status of the CobraNet Interface. This includes its conductor Status, Each Ethernet Jack Status as well as the Current Audio Mode and any Error indicators or codes that need to be displayed. Conductor Status: - 0 = Not Conductor - 1 = Conductor - All other values undefined
Implemented Products	PE Amp CobraNet Input Card
Attributes	Read Only
Byte Description	Byte # / Value / Description 1 / 0x79 / Option CobraNet Info 2 / 14 / Length of data field 3 / xx / Conductor Status 4 / xx / Current Interface 5 / xx / Interface 1 Status 6 / xx / Interface 2 Status 7-8 / xx / Mode Status 9-12 / xx / Error Status 13-16 / yy / Error Code

Name	OPT_COBRANET_RX_STATUS
Option #	0x7A
Description	Option to access the status of a receiver in a CobraNet Module. Rx Status = 'rxStatus' SubBundleFormat = 'rxSubFormat' Refer to CobraNet Programmers manual for more information.
Implemented Products	PE Amp CobraNet Input Card
Attributes	Read Only
Byte Description	Byte # / Value / Description 1 / 0x7A / Option CobraNet Info 2 / 3 / Length of data field 3 / xx / Receiver Number (0-3 = 1-4) 4 / xx / Rx Status 5 / xx / Sub-Bundle Format

<table>
<tr><td>Name</td><td>OPT_DSP_FUNC_PARAMS</td></tr>
<tr><td>Option #</td><td>0x81</td></tr>
<tr><td>Description</td><td>Option to allow access to DSP Function Parameters. This one option is used to access all DSP Function parameters (see Appendix B for a list of valid DSP Options). For each DSP Function Parameter, the first 5 bytes are the same, while the rest vary depending on the DSP Function.

DSP Channel Type
- 0 =Output Channel
- 1 = Input Channel
- 2 – 0xFF are Undefined and should not be used

DSP Channel
- Specific Channel Number (0-23 = 1to 24)

DSP Function
- See Appendix B.</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP Input Card</td></tr>
<tr><td>Attributes</td><td>Read/Write</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>xx</td><td>Length of data field (dependent on function type)</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-23 = 1-24)</td></tr>
<tr><td>5</td><td>yy</td><td>DSP Function (see section on DSP Functions)</td></tr>
<tr><td>XX</td><td>XX</td><td>Parameters Defined by DSP Functions</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>OPT_DSP_FUNC_METERS</td></tr>
<tr><td>Option #</td><td>0x82</td></tr>
<tr><td>Description</td><td>Option to allow access to DSP Meter Parameters. This one option is used to access all DSP Function Meters. See Appendix B. for a list of valid DSP Meter Options. Note this only defines the first 5 bytes, the rest are defined in appendix B. Not all DSP Function have Meters!

DSP Channel Type
- 0 =Output Channel
- 1 = Input Channel
- 2 – 0xFF are Undefined and should not be used

DSP Channel
- Specific Channel Number (0-23 = 1to 24)

DSP Function
- See Appendix B.</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP Input Card</td></tr>
<tr><td>Attributes</td><td>Read Only</td></tr>
<tr><td>Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x82</td><td>Option DSP Function Meters</td></tr>
<tr><td>2</td><td>xx</td><td>Length of data field (dependent on function type)</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-23 = 1-24)</td></tr>
<tr><td>5</td><td>yy</td><td>DSP Function (see section on DSP Functions)</td></tr>
<tr><td>XX</td><td>XX</td><td>Meters Defined by DSP Functions</td></tr>
</table>
</td></tr>
</table>

Name	OPT_DSP_CHANNEL_METER
Option #	0x88
Description	Option to access the Meters for DSP Channels. For Output channels this meter is after the last DSP Block before the final Mute. For Input channels this meter is before the input mute. DSP Channel Type - 0 =Output Channel - 1 = Input Channel - 2 – 0xFF are Undefined and should not be used DSP Channel - Specific Channel Number (0-23 = 1to 24) Meter Format: - Should Be set to 0 = IEEE Format - All other values are invalid Clip: - Indicates if the signal is clipping - 1-0xFF = Clipping - 0 = not clipping Meter - 32 bit IEEE Floating point value where 1.0 = 20dBu - 0 = -Inf
Implemented Products	PE Series DSP Input Card
Attributes	Read/Write
Byte Description	

Byte #	Value	Description
1	0x88	Option DSP Channel Meters
2	8	Length of data field (dependent on function type)
3	x	DSP Channel Type
4	x	DSP Channel
5	x	Meter Format
6	x	Clip
7-10	x	Meter

Name	OPT_END*
Option #	0xFF
Description	Required Option at the end of all Messages. **This special option has no length.**
Implemented Products	All Protea NE Products
Attributes	N/A
Byte Description	

Byte #	Value	Description
1	0xFF	Option User Info Identifier (no length or data)

Appendix B:

DSP FUNCTIONS

Overview:

The DSP Functions are grouped under a single set of options to handle parameters, meters, link group assignment and block assignment. This section defines the different parameters/ meters available for the different functions as well as what the function Identifier is for a particular function

0x00 FUNC_NONE
0x01 FUNC_GEQ28
0x02 FUNC_SIG_GEN
0x03 FUNC_AUTOLEV
0x04 FUNC_DUCKER
0x05 FUNC_LIMITER
0x06 FUNC_GATE
0x07 FUNC_PEQ2
0x08 FUNC_PEQ4
0x09 FUNC_PEQ6
0x0A FUNC_PEQ10
0x0C FUNC_PEQ15
0x0F FUNC_HPF
0x10 FUNC_LPF
0x11 FUNC_DELAY_BASE
0x12 FUNC_DELAY_EXTRA
0x13 FUNC_MIXER_X_IN
0x17 FUNC_GAIN
0x1A FUNC_METER
0x1F FUNC_CLIP_LIMITER
0x20 FUNC_GEQ31
0x21 FUNC_WR5_LEVEL
0x22 FUNC_REMOTE_LEVEL
0x25 FUNC_PREAMP
0x26 FUNC_FBS_CTRL
0x28 FUNC_ANC
0x64 FUNC_GAIN_VCA_ASSIGNMENT

<table>
<tr><td>Name</td><td>FUNC_GEQ28</td></tr>
<tr><td>Option #</td><td>0x01</td></tr>
<tr><td>Description</td><td>Graphic EQ w/ 28 Bands DSP Function. This DSP function has a lot of data, as it must control 28 filters. The request only requires the first 5 bits.

Bypass: (aa)
- Will Set the GEQ Bypass Status
- 0 = Active (Not Bypassed)
- 1-255 = Bypass

GEQ Type: (bb)
- Selects the Type of GEQ Filter to use
- 0 = Constant Q
- 1 = Proportional Q
- All other values unsupported.

GEQ Q: (cc)
- Q of each GEQ Band
- 59 = ¼ Octave to 71 = ½ Oct, step = 1/48 Oct.
- 0-58 are unsupported,
- 72-255 are unsupported

Band x Level: (xxxx)
- Boost/Cut for each of the 28 Bands
- 8042 to 8342 = -15dB to +15dB in .1dB steps</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+), PE4/8, NE Rackmount</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>62</td><td>Length of data field</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x01</td><td>Function: GEQ</td></tr>
<tr><td>6</td><td>ss</td><td>Bypass</td></tr>
<tr><td>7</td><td>tt</td><td>GEQ Type</td></tr>
<tr><td>8</td><td>xx</td><td>GEQ Q</td></tr>
<tr><td>9-10</td><td>xxxx</td><td>Band 1 Level (31Hz)</td></tr>
<tr><td>11-12</td><td>xxxx</td><td>Band 2 Level</td></tr>
<tr><td>13-14</td><td>xxxx</td><td>Band 3 Level</td></tr>
<tr><td>15-16</td><td>xxxx</td><td>Band 4 Level</td></tr>
<tr><td>17-18</td><td>xxxx</td><td>Band 5 Level</td></tr>
<tr><td>19-20</td><td>xxxx</td><td>Band 6 Level</td></tr>
<tr><td>21-22</td><td>xxxx</td><td>Band 7 Level</td></tr>
<tr><td>23-24</td><td>xxxx</td><td>Band 8 Level</td></tr>
<tr><td>25-26</td><td>xxxx</td><td>Band 9 Level</td></tr>
<tr><td>27-28</td><td>xxxx</td><td>Band 10 Level</td></tr>
<tr><td>29-30</td><td>xxxx</td><td>Band 11 Level</td></tr>
<tr><td>31-32</td><td>xxxx</td><td>Band 12 Level</td></tr>
<tr><td>33-34</td><td>xxxx</td><td>Band 13 Level</td></tr>
<tr><td>35-36</td><td>xxxx</td><td>Band 14 Level</td></tr>
<tr><td>37-38</td><td>xxxx</td><td>Band 15 Level</td></tr>
<tr><td>39-40</td><td>xxxx</td><td>Band 16 Level</td></tr>
<tr><td>41-42</td><td>xxxx</td><td>Band 17 Level</td></tr>
<tr><td>43-44</td><td>xxxx</td><td>Band 18 Level</td></tr>
<tr><td>45-46</td><td>xxxx</td><td>Band 19 Level</td></tr>
<tr><td>47-48</td><td>xxxx</td><td>Band 20 Level</td></tr>
<tr><td>49-50</td><td>xxxx</td><td>Band 21 Level</td></tr>
<tr><td>51-52</td><td>xxxx</td><td>Band 22 Level</td></tr>
<tr><td>53-54</td><td>xxxx</td><td>Band 23 Level</td></tr>
<tr><td>55-56</td><td>xxxx</td><td>Band 24 Level</td></tr>
<tr><td>57-58</td><td>xxxx</td><td>Band 25 Level</td></tr>
<tr><td>59-60</td><td>xxxx</td><td>Band 26 Level</td></tr>
<tr><td>61-62</td><td>xxxx</td><td>Band 27 Level</td></tr>
<tr><td>63-64</td><td>xxxx</td><td>Band 28 Level (16Khz)</td></tr>
</table>
</td></tr>
<tr><td>Function Meters
Byte Description</td><td>None</td></tr>
</table>

Name	FUNC_SIG_GEN		
Option #	0x02		
Description			
Implemented Products	PE Series DSP (v1.0+)		
Function Parameters Byte Description	Byte #	Value	Description
	1	0x81	Option DSP Function Parameters
	2	9	Length of data field
	3	xx	Channel Type (0 = Output, 1 = Input)
	4	xx	Channel (0-24= 1-25)
	5	0x02	Function Signal Generator
	6	ss	Bypass: 0 = active; 0x1-FF = bypass
	7	xx	Signal Type (0 = Sine, 1= White Noise, 2 = Pink Noise)
	8-9	xxxx	Signal Freq (20 – 20,000)
	10-11	xxxx	Signal Level (7692 – 8392 = -50dBu to + 20dBu) 0 = OFF
Function Meters Byte Description	None		

<table>
<tr><td>Name</td><td>FUNC_AUTO_LEV</td></tr>
<tr><td>Option #</td><td>0x03</td></tr>
<tr><td>Description</td><td>AutoLeveler Function. This attempts to maintain the output level at a given Target Level

Target Level:
- 60 to 120 = -40dBu to +20dBu
- Level that AutoLeveler will attempt to achieve.

Threshold below target:
- 70 to 100 = -30 to 0dB below target level
- Target – Threshold below Target is the level at which the AutoLeveler will begin to apply gain.

Ratio:
- Compression ratio.
- The Higher the ratio, the more aggressive the AutoLeveler
- The Higher the ratio the higher the maximum gain that will be applied.
- 0 to 6 = 1.2, 1.5,2,3,4,6, 10:to 1

Gain Increase Rate:
- Rate at which the AutoLeveler increases Gain
- 0 to 7 = 5,10,20,50,100,200,500, 100s ms/dB

Gain Decrease Rate:
- Rate at which the AutoLeveler decreases Gain
- 0 to 7 = 5,10,20,50,100,200,500, 100s ms/dB

Hold Time:
- Time after the level falls below threshold that gain is maintained.
- 0 to 6 = 0 to 6 seconds.

Meter Format
- Currently Only IEEE Floating Point Meter format is supported
- Leave at 0

Input Meter
- Input Signal Level in floating point, 1.0 = +20dBu

Gain/Attenuation
- Gain/ Attenuation value 1.0 = 0dBu Gain/Attenuation</td></tr>
<tr><td>Implemented Products</td><td>NE Rackmount</td></tr>
<tr><td>Function Parameters Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>10</td><td>Length of data field</td></tr>
<tr><td>3</td><td>yy</td><td>Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>Channel (0-24= 1-25)</td></tr>
<tr><td>5</td><td>0x03</td><td>Function Auto Leveler</td></tr>
<tr><td>6</td><td>aa</td><td>Bypass (0 = active; 0x1-FF = bypass)</td></tr>
<tr><td>7</td><td>bb</td><td>Target Level</td></tr>
<tr><td>8</td><td>cc</td><td>Threshold</td></tr>
<tr><td>9</td><td>dd</td><td>Ratio</td></tr>
<tr><td>10</td><td>ee</td><td>Gain Increase Rate</td></tr>
<tr><td>11</td><td>ff</td><td>Gain Decrease Rate</td></tr>
<tr><td>12</td><td>gg</td><td>Hold time</td></tr>
</table>
</td></tr>
<tr><td>Function Meters Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x82</td><td>Option DSP Function Meters</td></tr>
<tr><td>2</td><td>12</td><td>Length of data field (dependent on function type)</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x05</td><td>Function: Limiter</td></tr>
<tr><td>6</td><td>xx</td><td>Meter Format</td></tr>
<tr><td>7-10</td><td>zzzzzzzz</td><td>Input Meter</td></tr>
</table>
</td></tr>
</table>

	11-14	zzzzzzzz	Gain/Attenuation	

Name	FUNC_DUCKER
Option #	0x04
Description	Option to modify the Ducker Function in the DSP. This does not set the Block location however it will adjust all parameters. All parameters must be set. Bypass: (aa) - 0 – Function is Active (not Bypassed) - 1 to 0xFF = Function is Bypassed (will not duck or be ducked) Threshold: (bb) - Point at which channel Begins Ducking other channels - 20 to 120 = -80dBu to +20dBu - 1dB Increments - All other values are invalid and will cause the value to be saturated. Depth: (cc) - Amount of Attenuation applied when a channel is ducked - 0 to 31 = 0dB to –30dB, and Infinity (off) Release rate: (dd) - The rate at which attenuation is decreased. - 0 to 7 = 5, 10, 20, 50, 100, 200, 500 and 1000ms/dB Ducker Type: (ee) - The Type of Ducker - 0 – High Priority - 1 – Low Priority - 2 – Filibuster - 3 – Program All other values will saturate to 1000ms/dB and are considered invalid
Implemented Products	NE Rackmount, PE Multi Channel Amp

Function Parameters Byte Description

Byte #	Value	Description
1	0x81	Option DSP Function Parameters
2	8	Length of data field
3	yy	DSP Channel Type (0 = Output, 1 = Input)
4	yy	DSP Channel (0-255 = channels 1-256)
5	0x04	Function: Ducker
6	aa	Bypass
7	bb	Threshold
8	cc	Depth
9	dd	Release rate
10	ee	Ducker Type

Function Meters Byte Description

Byte #	Value	Description
1	0x82	Option DSP Function Meters
2	9	Length of data field (dependent on function type)
3	yy	DSP Channel Type (0 = Output, 1 = Input)
4	yy	DSP Channel (0-255 = channels 1-256)
5	0x04	Function: Ducker
6	xx	Meter Format
7	zz	Ducking Status (0 = Normal, 1= Ducked, 2 = Ducking, 3-FF = undefined)
8-11	zzzzzzzz	Input Meter

<table>
<tr><td>Name</td><td>FUNC_LIMITER</td></tr>
<tr><td>DSP Function #</td><td>0x05</td></tr>
<tr><td>Description</td><td>Option to modify the Limiter Function in the DSP. This does not set the Block location however it will adjust all parameters. All parameters must be set.

Bypass: (aa)
- 0 – Function is Active (not Bypassed)
- 1 to 0xFF = Function is Bypassed

Threshold: (bb)
- Point at which Limiting Begins
- 80 to 120 = -20dBu to +20dBu
- 1dB Increments
- All other values are invalid and will cause the value to be saturated.

Ratio: (cc)
- Ratio of Input Gain to Output Gain
- 0 to 8 = 1.2,1.5,2,3,4,6,10,20,Inf to 1
- All other values will saturate to Inf:1

Attack rate: (dd)
- Rate at which attenuation is increased
- 0 to 7 = 0.2,0.5,1,2,5,10,20,50 ms/dB
- All other values will saturate to 50ms/dB

Release rate: (ee)
- The rate at which attenuation is decreased.
- 0 to 7 = 5, 10, 20, 50, 100, 200, 500 and 1000ms/dB
- All other values will saturate to 1000ms/dB and are considered invalid

Attenuation Bus: (ff)
- The attenuation of multiple comp/limiters may be linked together with this option. The highest attenuation of all limiters on a bus is applied to all limiters.
- 0 = None (no attenuation bus)
- 1 = Attenuation Bus 1
- 2 = Attenuation Bus 2

Type: (gg)
- Compressor/Limiter Type determines how the attenuation is calculated
- 0 = Peak Detector used to calculate attenuation.
- 1 = Average Detector used to calculate attenuation.

Meter Format:
- Currently only 1 Meter format is supported. That format is IEEE floating point.</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+), NE Rackmount, PE Multi Channel Amp</td></tr>
<tr><td>Function Parameters Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>10</td><td>Length of data field</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-23 = 1-24)</td></tr>
<tr><td>5</td><td>0x05</td><td>Function: Limiter</td></tr>
<tr><td>6</td><td>aa</td><td>Bypass</td></tr>
<tr><td>7</td><td>bb</td><td>Threshold</td></tr>
<tr><td>8</td><td>cc</td><td>Ratio</td></tr>
<tr><td>9</td><td>dd</td><td>Attack rate</td></tr>
<tr><td>10</td><td>ee</td><td>Release rate</td></tr>
<tr><td>11</td><td>ff</td><td>Attenuation Bus</td></tr>
<tr><td>12</td><td>gg</td><td>Type</td></tr>
</table>
</td></tr>
<tr><td>Function Meters Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x82</td><td>Option DSP Function Meters</td></tr>
<tr><td>2</td><td>6</td><td>Length of data field (dependent on function type)</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
</table>
</td></tr>
</table>

	5	0x05	Function: Limiter	
	6	xx	Meter Format	
	7-10	zzzzzzzz	Input Meter	
	11-14	zzzzzzzz	Attenuation	

<table>
<tr><td>Name</td><td>FUNC_GATE</td></tr>
<tr><td>Option #</td><td>0x06</td></tr>
<tr><td>Description</td><td>Option to modify the Gate Function in the DSP. This does not set the Block location however it will adjust all parameters. All parameters must be set.

Bypass: (aa)
- 0 – Function is Active (not Bypassed)
- 1 to 0xFF = Function is Bypassed

Threshold: (bb)
- Point at which Limiting Begins
- 20 to 120 = -80dBu to +20dBu
- 1dB Increments
- All other values are invalid and will cause the value to be saturated.

Range: (cc)
- Amount of Attenuation applied when level is below threshold.
- 0 to 161 = 0dB to –160dB, and Infinity (off)
- The Maximum range allowed is 80-Threshold, if the range is below this value, Infinity is automatically used.

Attack rate: (dd)
- Rate at which attenuation is increased
- 0 to 7 = 0.2,0.5,1,2,5,10,20,50 ms/dB
- All other values will saturate to 50ms/dB

Release rate: (ee)
- The rate at which attenuation is decreased.
- 0 to 7 = 5, 10, 20, 50, 100, 200, 500 and 1000ms/dB

All other values will saturate to 1000ms/dB and are considered invalid</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+), NE Rackmount, PE Multi Channel Amp</td></tr>
<tr><td>Function Parameters Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>13 (8)</td><td>Length of data field (8 if not using Key & Lockout)</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x06</td><td>Function: Gate</td></tr>
<tr><td>6</td><td>aa</td><td>Bypass</td></tr>
<tr><td>7</td><td>bb</td><td>Threshold</td></tr>
<tr><td>8</td><td>bb</td><td>Range</td></tr>
<tr><td>9</td><td>cc</td><td>Attack rate</td></tr>
<tr><td>10</td><td>dd</td><td>Release rate</td></tr>
<tr><td>11</td><td>ee</td><td>Key Engaged*</td></tr>
<tr><td>12-13</td><td>ffff</td><td>Key Freq*</td></tr>
<tr><td>14</td><td>gg</td><td>Key Q*</td></tr>
<tr><td>15</td><td>hh</td><td>Lockout Engaged*</td></tr>
</table>
* Not Supported by PE DSP v1.x & 2.x
</td></tr>
<tr><td>Function Meters Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x82</td><td>Option DSP Function Meters</td></tr>
<tr><td>2</td><td>9</td><td>Length of data field (dependent on function type)</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x06</td><td>Function: Gate</td></tr>
<tr><td>6</td><td>xx</td><td>Meter Format</td></tr>
<tr><td>6</td><td>zz</td><td>Gate Status (0 = Closed, 1= Open,
2-FF = undefined)</td></tr>
<tr><td>7-10</td><td>zzzzzzzz</td><td>Input Meter</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Name</td><td>FUNC_PEQ2</td></tr>
<tr><td>Option #</td><td>0x07</td></tr>
<tr><td>Description</td><td>Note:
Option 1 – VariQ supported (replaces Normal LP & HP)
Option 2 – Band Pass Filter
Option 3 - Full Range HPF/LPF</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+)</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>18</td><td>Length of data field</td></tr>
<tr><td>3</td><td>xx</td><td>Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>xx</td><td>Channel</td></tr>
<tr><td>5</td><td>0x07</td><td>Function PEQ 2</td></tr>
<tr><td>6</td><td>ss</td><td>Bypass (0 = active; 0x1-FF = bypass)</td></tr>
<tr><td>7</td><td>ss</td><td>Filter 1 Bypass: 0 = active, 0x1-FF = bypass</td></tr>
<tr><td>8</td><td>xx</td><td>Filter 1 Type (0-8=PEQ, LS1, LS2, HS1, HS2, ALLPASS, LP, HP, NOTCH)</td></tr>
<tr><td>9-10</td><td>xxxx</td><td>Filter 1 Freq 20Hz – 20Khz</td></tr>
<tr><td>11-12</td><td>xxxx</td><td>Filter 1 Level 20-120 = -80 to +20dBu</td></tr>
<tr><td>13</td><td>xx</td><td>Filter 1 Q</td></tr>
<tr><td>14</td><td>ss</td><td>Filter 2 Bypass</td></tr>
<tr><td>14</td><td>xx</td><td>Filter 2 Type</td></tr>
<tr><td>15-16</td><td>xxxx</td><td>Filter 2 Freq</td></tr>
<tr><td>17-18</td><td>xxxx</td><td>Filter 2 Level</td></tr>
<tr><td>19</td><td>xx</td><td>Filter 2 Q</td></tr>
</table>
</td></tr>
<tr><td>Function Meters
Byte Description</td><td>None</td></tr>
</table>

<table>
<tr><td>Name</td><td>FUNC_PEQ4</td></tr>
<tr><td>Option #</td><td>0x08</td></tr>
<tr><td>Description</td><td>Note:
Option 1 – VariQ supported (replaces Normal LP & HP)
Option 2 – Band Pass Filter
Option 3 - Full Range HPF/LPF</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+)</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><th>Byte #</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>32</td><td>Length of data field</td></tr>
<tr><td>3</td><td>xx</td><td>Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>xx</td><td>Channel</td></tr>
<tr><td>5</td><td>0x08</td><td>Function PEQ 4</td></tr>
<tr><td>6</td><td>ss</td><td>Bypass (0 = active, 0x1-FF = bypass)</td></tr>
<tr><td>7</td><td>xx</td><td>Filter 1 Bypass</td></tr>
<tr><td>8</td><td>xx</td><td>Filter 1 Type</td></tr>
<tr><td>9-10</td><td>xxxx</td><td>Filter 1 Freq</td></tr>
<tr><td>11-12</td><td>xxxx</td><td>Filter 1 Level</td></tr>
<tr><td>13</td><td>xx</td><td>Filter 1 Q</td></tr>
<tr><td>14</td><td>xx</td><td>Filter 2 Bypass</td></tr>
<tr><td>15</td><td>xx</td><td>Filter 2 Type</td></tr>
<tr><td>16-17</td><td>xxxx</td><td>Filter 2 Freq</td></tr>
<tr><td>18-19</td><td>xxxx</td><td>Filter 2 Level</td></tr>
<tr><td>20</td><td>xx</td><td>Filter 2 Q</td></tr>
<tr><td>21</td><td>xx</td><td>Filter 3 Bypass</td></tr>
<tr><td>22</td><td>xx</td><td>Filter 3 Type</td></tr>
<tr><td>23-24</td><td>xxxx</td><td>Filter 3 Freq</td></tr>
<tr><td>25-26</td><td>xxxx</td><td>Filter 3 Level</td></tr>
<tr><td>27</td><td>xx</td><td>Filter 3 Q</td></tr>
<tr><td>28</td><td>xx</td><td>Filter 4 Bypass</td></tr>
<tr><td>29</td><td>xx</td><td>Filter 4 Type</td></tr>
<tr><td>30-31</td><td>xxxx</td><td>Filter 4 Freq</td></tr>
<tr><td>32-33</td><td>xxxx</td><td>Filter 4 Level</td></tr>
<tr><td>34</td><td>xx</td><td>Filter 4 Q</td></tr>
</table>
</td></tr>
<tr><td>Function Meters
Byte Description</td><td>Metering
None – see clipping message</td></tr>
</table>

Name	FUNC_PEQ6
Option #	0x09
Description	Note: Option 1 – VariQ supported (replaces Normal LP & HP) Option 2 – Band Pass Filter Option 3 - Full Range HPF/LPF
Implemented Products	PE Series DSP (v1.0+)

Function Parameters Byte Description

Byte #	Value	Description
1	0x81	Option DSP Function Parameters
2	46	Length of data field
3	xx	Channel Type (0 = Output, 1 = Input)
4	xx	Channel
5	0x09	Function PEQ6
6	ss	Bypass (0 = active, 0x1-7F = bypass)
7	xx	Filter 1 Bypass
8	xx	Filter 1 Type
9-10	xxxx	Filter 1 Freq
11-12	xxxx	Filter 1 Level
13	xx	Filter 1 Q
14	xx	Filter 2 Bypass
15	xx	Filter 2 Type
16-17	xxxx	Filter 2 Freq
18-19	xxxx	Filter 2 Level
20	xx	Filter 2 Q
21	xx	Filter 3 Bypass
22	xx	Filter 3 Type
23-24	xxxx	Filter 3 Freq
25-26	xxxx	Filter 3 Level
27	xx	Filter 3 Q
28	xx	Filter 4 Bypass
29	xx	Filter 4 Type
30-31	xxxx	Filter 4 Freq
32-33	xxxx	Filter 4 Level
34	xx	Filter 4 Q
35	xx	Filter 5 Bypass
36	xx	Filter 5 Type
37-38	xxxx	Filter 5 Freq
39-40	xxxx	Filter 5 Level
41	xx	Filter 5 Q
42	xx	Filter 6 Bypass
43	xx	Filter 6 Type
44-45	xxxx	Filter 6 Freq
46-47	xxxx	Filter 6 Level
48	xx	Filter 6 Q

Function Meters Byte Description	Metering None – See clipping message

Name	FUNC_PEQ10
Option #	0x0A
Description	Note: Option 1 – VariQ supported (replaces Normal LP & HP) Option 2 – Band Pass Filter Option 3 - Full Range HPF/LPF
Implemented Products	PE Series DSP (v1.0+)

Function Parameters Byte Description

Byte #	Value	Description
1	0x81	Option DSP Function Parameters
2	74	Length of data field
3	xx	Channel Type (0 = Output, 1 = Input)
4	xx	Channel
5	0x0a	Function PEQ 10
6	ss	Bypass (0 = active, 0x1-7F = bypass)
7	xx	Filter 1 Bypass
8	xx	Filter 1 Type
9-10	xxxx	Filter 1 Freq
11-12	xxxx	Filter 1 Level
13	xx	Filter 1 Q
14	xx	Filter 2 Bypass
15	xx	Filter 2 Type
16-17	xxxx	Filter 2 Freq
18-19	xxxx	Filter 2 Level
20	xx	Filter 2 Q
21	xx	Filter 3 Bypass
22	xx	Filter 3 Type
23-24	xxxx	Filter 3 Freq
25-26	xxxx	Filter 3 Level
27	xx	Filter 3 Q
28	xx	Filter 4 Bypass
29	xx	Filter 4 Type
30-31	xxxx	Filter 4 Freq
32-33	xxxx	Filter 4 Level
34	xx	Filter 4 Q
35	xx	Filter 5 Bypass
36	xx	Filter 5 Type
37-38	xxxx	Filter 5 Freq
39-40	xxxx	Filter 5 Level
41	xx	Filter 5 Q
42	xx	Filter 6 Bypass
43	xx	Filter 6 Type
44-45	xxxx	Filter 6 Freq
46-47	xxxx	Filter 6 Level
48	xx	Filter 6 Q
49	xx	Filter 7 Bypass
50	xx	Filter 7 Type
51-52	xxxx	Filter 7 Freq
53-54	xxxx	Filter 7 Level
55	xx	Filter 7 Q
56	xx	Filter 8 Bypass
57	xx	Filter 8 Type
58-59	xxxx	Filter 8 Freq
60-61	xxxx	Filter 8 Level
62	xx	Filter 8 Q
63	xx	Filter 9 Bypass
64	xx	Filter 9 Type
65-66	xxxx	Filter 9 Freq
67-68	xxxx	Filter 9 Level
69	xx	Filter 9 Q
70	xx	Filter 10 Bypass
71	xx	Filter 10 Type
72-73	xxxx	Filter 10 Freq
74-75	xxxx	Filter 10 Level
76	xx	Filter 10 Q

Function Meters Byte Description	None

Name	FUNC_PEQ15

Option #	0x0B
Description	Note: Option 1 – VariQ supported (replaces Normal LP & HP) Option 2 – Band Pass Filter Option 3 - Full Range HPF/LPF
Implemented Products	PE Series DSP (v1.0+)

Function Parameters Byte Description

Byte #	Value	Description
1	0x81	Option DSP Function Parameters
2	109	Length of data field
3	xx	Channel Type (0 = Output, 1 = Input)
4	xx	Channel
5	0x0a	Function PEQ 10
6	ss	Bypass: 0 = active; 0x1-7F = bypass
7	xx	Filter 1 Bypass
8	xx	Filter 1 Type
9-10	xxxx	Filter 1 Freq
11-12	xxxx	Filter 1 Level
13	xx	Filter 1 Q
14	xx	Filter 2 Bypass
15	xx	Filter 2 Type
16-17	xxxx	Filter 2 Freq
18-19	xxxx	Filter 2 Level
20	xx	Filter 2 Q
21	xx	Filter 3 Bypass
22	xx	Filter 3 Type
23-24	xxxx	Filter 3 Freq
25-26	xxxx	Filter 3 Level
27	xx	Filter 3 Q
28	xx	Filter 4 Bypass
29	xx	Filter 4 Type
30-31	xxxx	Filter 4 Freq
32-33	xxxx	Filter 4 Level
34	xx	Filter 4 Q
35	xx	Filter 5 Bypass
36	xx	Filter 5 Type
37-38	xxxx	Filter 5 Freq
39-40	xxxx	Filter 5 Level
41	xx	Filter 5 Q
42	xx	Filter 6 Bypass
43	xx	Filter 6 Type
44-45	xxxx	Filter 6 Freq
46-47	xxxx	Filter 6 Level
48	xx	Filter 6 Q
49	xx	Filter 7 Bypass
50	xx	Filter 7 Type
51-52	xxxx	Filter 7 Freq
53-54	xxxx	Filter 7 Level
55	xx	Filter 7 Q
56	xx	Filter 8 Bypass
57	xx	Filter 8 Type
58-59	xxxx	Filter 8 Freq
60-61	xxxx	Filter 8 Level
62	xx	Filter 8 Q
63	xx	Filter 9 Bypass
64	xx	Filter 9 Type
65-66	xxxx	Filter 9 Freq
67-68	xxxx	Filter 9 Level
69	xx	Filter 9 Q
70	xx	Filter 10 Bypass
71	xx	Filter 10 Type
72-73	xxxx	Filter 10 Freq
74-75	xxxx	Filter 10 Level
76	xx	Filter 10 Q
77	xx	Filter 11 Bypass
78	xx	Filter 11 Type
79-80	xxxx	Filter 11 Freq
81-82	xxxx	Filter 11 Level
83	xx	Filter 11 Q

	84	xx	Filter 12 Bypass	
	85	xx	Filter 12 Type	
	86-87	xxxx	Filter 12 Freq	
	88-89	xxxx	Filter 12 Level	
	90	xx	Filter 12 Q	
	91	xx	Filter 13 Bypass	
	92	xx	Filter 13 Type	
	93-94	xxxx	Filter 13 Freq	
	95-96	xxxx	Filter 13 Level	
	97	xx	Filter 13 Q	
	98	xx	Filter 14 Bypass	
	99	xx	Filter 14 Type	
	100-101	xxxx	Filter 14 Freq	
	102-103	xxxx	Filter 14 Level	
	104	xx	Filter 14 Q	
	105	xx	Filter 15 Bypass	
	106	xx	Filter 15 Type	
	107-108	xxxx	Filter 15 Freq	
	109-110	xxxx	Filter 15 Level	
	111	xx	Filter 15 Q	
Function Meters Byte Description	None			

Name	FUNC_HPF
Option #	0x0F
Description	High Pass Filter Function. This option modifies the HPF for a particular channel. HPF Type: (aa) - Type of High Pass Filter to use - 0 – Butterworth (12dB/Octave) - 1 – Bessel (12dB/Octave) - 2 – Linkwitz/Riley (12dB/Octave) - 3 – Butterworth/Linkwitz/Riley (18dB/Octave) - 4 – Bessel (18dB/Octave) - 5 – Butterworth (24dB/Octave) - 6 – Bessel (24dB/Octave) - 7 – Linkwitz/Riley (24dB/Octave) - 8 – Butterworth (48dB/Octave) - 9 – Bessel (48dB/Octave) - 10 – Linkwitz/Riley (48dB/Octave) HPF Freq: (bb) - HPF Frequency - Valid Range 20 to 20,000 - 19 = Off
Implemented Products	PE Series DSP (v1.0+)

Function Parameters Byte Description

Byte #	Value	Description
1	0x81	Option DSP Function Parameters
2	6	Length of data field
3	yy	DSP Channel Type (0 = Output, 1 = Input)
4	yy	DSP Channel (0-255 = channels 1-256)
5	0x0F	Function: HPF
6	aa	HPF Type
7-8	bbbb	HPF Freq

Function Meters Byte Description	None

<table>
<tr><td>Name</td><td>FUNC_LPF</td></tr>
<tr><td>Option #</td><td>0x10</td></tr>
<tr><td>Description</td><td>Low pass filter

LPF Type: (aa)
- Type of Low Pass Filter to use
- 0 – Butterworth (12dB/Octave)
- 1 – Bessel (12dB/Octave)
- 2 – Linkwitz/Riley (12dB/Octave)
- 3 – Butterworth/Linkwitz/Riley (18dB/Octave)
- 4 – Bessel (18dB/Octave)
- 5 – Butterworth (24dB/Octave)
- 6 – Bessel (24dB/Octave)
- 7 – Linkwitz/Riley (24dB/Octave)
- 8 – Butterworth (48dB/Octave)
- 9 – Bessel (48dB/Octave)
- 10 – Linkwitz/Riley (48dB/Octave)

LPF Freq: (bb)
- Low Pass Filter Frequency
- Valid Range 20 to 20,000
- 20,001 = OFF
- all other values are invalid</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+)</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>6</td><td>Length of data field</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x10</td><td>Function: HPF</td></tr>
<tr><td>6</td><td>aa</td><td>LPF Type</td></tr>
<tr><td>7-9</td><td>bbbb</td><td>LPF Freq (20-20000)</td></tr>
</table>
</td></tr>
<tr><td>Function Meters
Byte Description</td><td>None</td></tr>
</table>

<table>
<tr><td>Name</td><td>FUNC_DELAY_BASE</td></tr>
<tr><td>Option #</td><td>0x11</td></tr>
<tr><td>Description</td><td></td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+)</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>6</td><td>Length of data field</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x11</td><td>Function: Delay Base</td></tr>
<tr><td>7</td><td>zz</td><td>Bypass: 0 = active; 0x1-7F = bypass</td></tr>
<tr><td>7-8</td><td>zzzz</td><td>Delay in samples
Input (0 - 24575 samples = 511.979ms @ 48KHz, 255.990ms @ 96KHz)
Output (0 – 1023 samples = 21.333ms @ 48KHz, 10.667ms @ 96KHz)</td></tr>
</table>
</td></tr>
<tr><td>Function Meters
Byte Description</td><td>None</td></tr>
</table>

<table>
<tr><td>Name</td><td>FUNC_DELAY_EXTRA</td></tr>
<tr><td>Option #</td><td>0x12</td></tr>
<tr><td>Description</td><td></td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+)</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>6</td><td>Length of data field</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x12</td><td>Function: Delay Extra</td></tr>
<tr><td>6</td><td>zz</td><td>Bypass: 0 = active; 0x1-7F = bypass</td></tr>
<tr><td>7-8</td><td>zzzz</td><td>Delay in samples - Input/Output (0 - 21844 samples = 455.083ms @ 48KHz, 227.542ms @ 96KHz)</td></tr>
</table>
</td></tr>
<tr><td>Function Meters
Byte Description</td><td>None</td></tr>
</table>

<table>
<tr><td>Name</td><td>FUNC_MIXER_X_IN</td></tr>
<tr><td>Option #</td><td>0x13</td></tr>
<tr><td>Description</td><td>DSP Function to handle Mixer Settings. The PE Series DSP does not Support the Enable Flag, Only Mute.

The number of channels specified by this message is the maximum number of input channels in the product's family, not the physical number of inputs in the target device. The unused channels can be padded out with zeroes.

<table>
<tr><th>Product Family</th><th>Maximum Channels</th><th>Message Length</th></tr>
<tr><td>All 2-channel amps</td><td>2</td><td>2*3+3 = 9</td></tr>
<tr><td>All multichannel amps
All PEMA
All neXX00 signal processors</td><td>8</td><td>8*3+3 = 27</td></tr>
<tr><td>Any ne24.24M signal processor</td><td>20</td><td>20*3+3 = 63</td></tr>
</table>

Gain: aaaa
- Level/Gain for a given Input
- Range = -50 to + 12dB (7692 to 8312)
- 0 = Off

Mute/Enable: (bb)
- Bitwise Flags
- Bit 0 = Mute (0 = Unmuted, 1 = mute)
- Bit 1 = Enabled (0 = Enabled, 1 = Not Enabled) *

* Not Supported By PE Series DSP</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+), NE Rackmount, PE Multichannel Amplifier.</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>xx</td><td>Length of data = 9, 27, or 63</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>zz</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x13</td><td>Function: Mixer</td></tr>
<tr><td>6-7</td><td>aaaa</td><td>In 1 Gain (-50 to +12 = 7692 to 8312, Off = 0)</td></tr>
<tr><td>8</td><td>bb</td><td>In 1 Mute / Enable</td></tr>
<tr><td>9-10</td><td>aaaa</td><td>In 2 Gain (-50 to +12 = 7692 to 8312, Off = 0)</td></tr>
<tr><td>11</td><td>bb</td><td>In 2 Mute / Enable</td></tr>
</table>

This may be extended for more than 2 channels.</td></tr>
<tr><td>Function Meters
Byte Description</td><td>(none) see clipping message</td></tr>
</table>

<table>
<tr><td>Name</td><td>FUNC_GAIN</td></tr>
<tr><td>Option #</td><td>0x17</td></tr>
<tr><td>Description</td><td>Option to modify the Gain Function in the DSP. This does not set the Block location however it will adjust all parameters. All parameters must be set.

Gain: (aaaa)
- The amount of Gain/Attenuation to apply.
- 7692 to 8312 = -50dB to +12dB
- 0 = Off (Mute)
- .1dB increments

Polarity: (bb)
- Sets the polarity of the gain block
- 0 = Normal polarity.
- 1 = Inverted polarity.</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+), NE Rackmount, PE Multi Channel Amp</td></tr>
<tr><td>Function Parameters Byte Description</td><td><table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>6</td><td>Length of data field</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x17</td><td>Function: Gain</td></tr>
<tr><td>6-7</td><td>aaaa</td><td>Gain</td></tr>
<tr><td>8</td><td>bb</td><td>Polarity</td></tr>
</table></td></tr>
<tr><td>Function Meters Byte Description</td><td><u>None – see clipping message</u></td></tr>
</table>

<table>
<tr><td>Name</td><td>FUNC_METER</td></tr>
<tr><td>Option #</td><td>0x1A</td></tr>
<tr><td>Description</td><td>Option to modify the Meter Block. The meter block has No parameters and thus not Function Parameters Message. However there is a meter message to receive the Meter level

Meter Format: (aa)
- 0 = IEEE Floating Point
- Currently only IEEE Floating point is supported
- This parameter may be used for the request to request metering in a specific format.

Input Meter: (bbbbbbbb)
- 4 byte input meter in specified format from Meter Format</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+)</td></tr>
<tr><td>Function Parameters Byte Description</td><td>(NONE)</td></tr>
<tr><td>Function Meters Byte Description</td><td><table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x82</td><td>Option DSP Function Meters</td></tr>
<tr><td>2</td><td>5</td><td>Length of data field (dependent on function type)</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x1A</td><td>Function: Meter</td></tr>
<tr><td>6</td><td>aa</td><td>Meter Format</td></tr>
<tr><td>7-10</td><td>bbbbbbbb</td><td>Input Meter</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Name</td><td>FUNC_CLIP_LIMITER</td></tr>
<tr><td>Option #</td><td>0x1F</td></tr>
<tr><td>Description</td><td>Option to modify the Clip Limiter Function in the DSP. This does not set the Block location however it will adjust all parameters. All parameters must be set. The Clip limiter is a more limited version of the compressor limiter. It does not allow for changes in the type and also ratio.

Bypass: (aa)
- 0 – Function is Active (not Bypassed)
- 1 to 0xFF = Function is Bypassed

Threshold: (bb)
- Point at which Limiting Begins
- 80 to 120 = -20dBu to +20dBu
- 1dB Increments
- All other values are invalid and will cause the value to be saturated.

Attack rate: (dd)
- Rate at which attenuation is increased
- 0 to 7 = 0.2,0.5,1,2,5,10,20,50 ms/dB
- All other values will saturate to 50ms/dB

Release rate: (ee)
- The rate at which attenuation is decreased.
- 0 to 7 = 5, 10, 20, 50, 100, 200, 500 and 1000ms/dB
- All other values will saturate to 1000ms/dB and are considered invalid

Meter Format:
- Currently only 1 Meter format is supported. That format is IEEE floating point.</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+), NE Rackmount, PE Multi Channel Amp</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>7</td><td>Length of data field</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-23 = 1-24)</td></tr>
<tr><td>5</td><td>0x1F</td><td>Function: Clip Limiter</td></tr>
<tr><td>6</td><td>aa</td><td>Bypass</td></tr>
<tr><td>7</td><td>bb</td><td>Threshold</td></tr>
<tr><td>8</td><td>dd</td><td>Attack rate</td></tr>
<tr><td>9</td><td>ee</td><td>Release rate</td></tr>
</table>
</td></tr>
<tr><td>Function Meters
Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x82</td><td>Option DSP Function Meters</td></tr>
<tr><td>2</td><td>6</td><td>Length of data field (dependent on function type)</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x1F</td><td>Function: Limiter</td></tr>
<tr><td>6</td><td>xx</td><td>Meter Format</td></tr>
<tr><td>7-10</td><td>zzzzzzzz</td><td>Input Meter</td></tr>
<tr><td>11-14</td><td>zzzzzzzz</td><td>Attenuation</td></tr>
</table>
</td></tr>
</table>

Name	FUNC_GEQ31
Option #	0x20
Description	Graphic EQ w/ 31 Bands DSP Function. This DSP function has a lot of data, as it must control 31 filters. The request only requires the first 5 bits. Bypass: (aa) - Will Set the GEQ Bypass Status - 0 = Active (Not Bypassed) - 1-255 = Bypass GEQ Type: (bb) - Selects the Type of GEQ Filter to use - 0 = Constant Q - 1 = Proportional Q - All other values unsupported. GEQ Q: (cc) - Q of each GEQ Band - 59 = ¼ Octave to 71 = ½ Oct, step = 1/48 Oct. - 0-58 are unsupported, - 72-255 are unsupported Band x Level: (xxxx) - Boost/Cut for each of the 28 Bands - 8042 to 8342 = -15dB to +15dB in .1dB steps
Implemented Products	PE Series DSP (v1.0+), PE4/8, NE Rackmount

Function Parameters Byte Description

Byte #	Value	Description
1	0x81	Option DSP Function Parameters
2	68	Length of data field
3	yy	DSP Channel Type (0 = Output, 1 = Input)
4	yy	DSP Channel (0-255 = channels 1-256)
5	0x20	Function: GEQ31
6	ss	Bypass
7	tt	GEQ Type
8	xx	GEQ Q
9-10	xxxx	Band 1 Level (20Hz)
11-12	xxxx	Band 2 Level
13-14	xxxx	Band 3 Level
15-16	xxxx	Band 4 Level
17-18	xxxx	Band 5 Level
19-20	xxxx	Band 6 Level
21-22	xxxx	Band 7 Level
23-24	xxxx	Band 8 Level
25-26	xxxx	Band 9 Level
27-28	xxxx	Band 10 Level
29-30	xxxx	Band 11 Level
31-32	xxxx	Band 12 Level
33-34	xxxx	Band 13 Level
35-36	xxxx	Band 14 Level
37-38	xxxx	Band 15 Level
39-40	xxxx	Band 16 Level
41-42	xxxx	Band 17 Level
43-44	xxxx	Band 18 Level
45-46	xxxx	Band 19 Level
47-48	xxxx	Band 20 Level
49-50	xxxx	Band 21 Level
51-52	xxxx	Band 22 Level
53-54	xxxx	Band 23 Level
55-56	xxxx	Band 24 Level
57-58	xxxx	Band 25 Level
59-60	xxxx	Band 26 Level
61-62	xxxx	Band 27 Level
63-64	xxxx	Band 28 Level
65-66	xxxx	Band 29 Level
67-68	xxxx	Band 30 Level
69-70	xxxx	Band 31 Level

Function Meters Byte Description	None

Name	FUNC_WR5_LEVEL
Option #	0x21
Description	Option to modify the WR5 Level DSP Block. The WR5 Level block applies attenuation to the signal based on WR5's that are present in the System. This is also user adjustable and will be tracked by assigned WR5s. WR5_Atten (bb) - WR5 Attenuation Value between 0 & 99 - 0 = Full Attenuation (mute) - 99 = No Attenuation - each step is ½ dB
Implemented Products	NE Rackmount, PE Multi Channel Amp
Function Parameters Byte Description	Byte # / Value / Description 1 / 0x81 / Option DSP Function Parameters 2 / 5 / Length of data field 3 / yy / DSP Channel Type (0 = Output, 1 = Input) 4 / yy / DSP Channel (0-23 = 1-24) 5 / 0x21 / Function: WR5 Atten 6 / 00 / Reserved 7 / bb / WR5 Attenuation
Function Meters Byte Description	(none)

Name	FUNC_REMOTE_LEVEL
Option #	0x22
Description	Option to modify the Remote Level DSP Block. The Remote Level block applies attenuation to the signal based on various types of remotes such as RD8C And Rear panel Pots that are present in the System. Bypass: (aa) - 0 – Function is Active (not Bypassed) - 1 to 0xFF = Function is Bypassed RD8C Assignments (cc) and Rear Panel Assignments (bb) - Bitwise flag indicating if each fader is assigned - Bit 0 = fader 1, Bit 1 = fader 2, etc. - 0 = Not Assigned, 1 = Assigned.
Implemented Products	NE Rackmount, PE Multi Channel Amp
Function Parameters Byte Description	Byte # / Value / Description 1 / 0x81 / Option DSP Function Parameters 2 / 6 / Length of data field 3 / yy / DSP Channel Type (0 = Output, 1 = Input) 4 / yy / DSP Channel (0-23 = 1-24) 5 / 0x22 / Function: Clip Limiter 6 / aa / Bypass 7 / bb / Rear Panel Assignments 8 / cc / RD8C Assignments
Function Meters Byte Description	(none)

<table>
<tr><td>Name</td><td>FUNC_PREAMP</td></tr>
<tr><td>Option #</td><td>0x25</td></tr>
<tr><td>Description</td><td>Option to modify the Preamp Function in the DSP. This does not set the Block location however it will adjust all parameters. All parameters must be set.

Preamp Gain: (aa)
- Preamp Gain level in dB.
- 0 = 0dB
- 20 = 20dB
- 40 = 40dB
- 60 = 60dB

Phantom Power: (bb)
- 0 = Phantom Power Off
- 1-FF = Phantom Power On

Touch To Talk Enabled (cc)
- 0 = Touch To Talk Off
- 1-FF = Touch To Talk On.

Touch To Talk Logic Input
- Logic input number to use for touch to talk.</td></tr>
<tr><td>Implemented Products</td><td>NE24.24M</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>7</td><td>Bytes to follow</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x25</td><td>Function: Preamp</td></tr>
<tr><td>6</td><td>aa</td><td>Preamp Gain</td></tr>
<tr><td>7</td><td>bb</td><td>Phantom Power</td></tr>
<tr><td>8</td><td>cc</td><td>Touch To Talk Enabled</td></tr>
<tr><td>9</td><td>dd</td><td>Touch To Talk Input</td></tr>
</table>
</td></tr>
<tr><td>Function Meters
Byte Description</td><td>None – see clipping message</td></tr>
</table>

<table>
<tr><td>Name</td><td>FUNC_FBS_CTRL</td></tr>
<tr><td>Option #</td><td>0x26</td></tr>
<tr><td>Description</td><td>Option to modify the FBS Function in the DSP. This does not set the Block location however it will adjust all parameters. All parameters must be set.
Note: Only Supported on Inputs!</td></tr>
<tr><td>Implemented Products</td><td>NE Rackmount</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>9</td><td>Bytes to follow</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x26</td><td>Function: FBS</td></tr>
<tr><td>6</td><td>aa</td><td>Bypass</td></tr>
<tr><td>7</td><td>bb</td><td>Sensitivity</td></tr>
<tr><td>8-11</td><td>cc</td><td>Filters Timeout</td></tr>
</table>
</td></tr>
<tr><td>Function Meters
Byte Description</td><td>None – see clipping message</td></tr>
</table>

Name	FUNC_ANC
Option #	0x28
Description	Option to modify the Ambient noise compensation block
Implemented Products	NE Rackmount

Function Parameters Byte Description

Byte #	Value	Description
1	0x81	Option DSP Function Parameters
2	25	Bytes to follow
3	yy	DSP Channel Type (0 = Output, 1 = Input)
4	yy	DSP Channel (0-255 = channels 1-256)
5	0x28	Function: ANC
6	Aa	Bypass
7-10	Bb	Max Gain
11-14	bb	Min Gain
15-18	Bb	Gain Change Rate : 1
19	Cc	Mixer Input Channel
20-23	Bb	Noise Threshold (dBFS)
24-27	Bb	Prog/ Amb. Ratio : 1

Function Meters Byte Description

Byte #	Value	Description
1	0x82	Option DSP Function Meters
2	6	Length of data field (dependent on function type)
3	yy	DSP Channel Type (0 = Output, 1 = Input)
4	yy	DSP Channel (0-255 = channels 1-256)
5	0x28	Function: ANC
6	xx	Meter Format
7-10	zzzzzzzz	Channel Meter
11-14	zzzzzzzz	Gain/ Attenuation
15-18	Zzzzzzzz	Ambient Input Meter

<table>
<tr><td>Name</td><td>FUNC_GAIN_VCA_ASSIGNMENT</td></tr>
<tr><td>Option #</td><td>0x64</td></tr>
<tr><td>Description</td><td>Option to modify the VCA Gain Assignment for a gain block in the DSP. This does not set the Block location however it will adjust all parameters. All parameters must be set. This function does not have a specific block associated with it; rather it is part of the Gain block.

VCAs Enabled: (aa)
- 0x01 to 0xFF =True indicates that VCA's should be used with the Gain block on this channel.
- 0x00 = False indicates that VCA's should NOT be used with the gain block on this channel.
- The gain block as defined in this document will function in either case.
- When VCA's are enabled then the total gain applied by the block is the sum of all assigned VCA's + the Gain blocks Gain.

VCA Assignments (bb)
- Bitwise assignment Variable, if a bit is set then the corresponding VCA is assigned to the gain block on this channel (if VCA's Enabled)
- Bit 0 = VCA 1 (Mask: 0x01)
- Bit 1 = VCA 2 (Mask: 0x02)
- Bit 2 = VCA 3 (Mask: 0x04)
- Bit 3 = VCA 4 (Mask: 0x08)</td></tr>
<tr><td>Implemented Products</td><td>PE Series DSP (v1.0+), NE Rackmount, PE Multi Channel Amp</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>5</td><td>Length of data field</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>0x64</td><td>Function: Gain VCA Assignments</td></tr>
<tr><td>6</td><td>aa</td><td>VCAs Enabled</td></tr>
<tr><td>7</td><td>bb</td><td>VCA Assignments (bitwise)</td></tr>
</table>
</td></tr>
<tr><td>Function Meters
Byte Description</td><td>(NONE)</td></tr>
</table>

<table>
<tr><td>Name</td><td>FUNC_FBS_FILTER</td></tr>
<tr><td>Option #</td><td></td></tr>
<tr><td>Description</td><td></td></tr>
<tr><td>Implemented Products</td><td>NE Rackmount</td></tr>
<tr><td>Function Parameters
Byte Description</td><td>
<table>
<tr><td>Byte #</td><td>Value</td><td>Description</td></tr>
<tr><td>1</td><td>0x81</td><td>Option DSP Function Parameters</td></tr>
<tr><td>2</td><td>22</td><td>Bytes to follow</td></tr>
<tr><td>3</td><td>yy</td><td>DSP Channel Type (0 = Output, 1 = Input)</td></tr>
<tr><td>4</td><td>yy</td><td>DSP Channel (0-255 = channels 1-256)</td></tr>
<tr><td>5</td><td>xx</td><td></td></tr>
<tr><td>6</td><td>aa</td><td>Filter Number</td></tr>
<tr><td>7</td><td>bb</td><td>Filter Bypass</td></tr>
<tr><td>8</td><td>cc</td><td>Filter Type</td></tr>
<tr><td>9</td><td>dd</td><td>Filter Mode.</td></tr>
<tr><td>10</td><td>ee</td><td>Filter Lockout</td></tr>
<tr><td>11-14</td><td>Ff</td><td>Filter Freq (IEEE Floating Point Hz)</td></tr>
<tr><td>15-18</td><td>Gg</td><td>Filter Level (IEEE Floating Point dB)</td></tr>
<tr><td>19-22</td><td>Hh</td><td>Filter Q (IEEE Floating Point)</td></tr>
</table>
</td></tr>
<tr><td>Function Meters
Byte Description</td><td>None – see clipping message</td></tr>
</table>

Appendix C:

SAMPLE MESSAGES

Sample Mute Message:

These messages will Get the Mute Status as well as Mute/Unmute a channel on a particular Ashly Device. Since All Ashly Network products use the same protocol this will work on All Networked products.

Getting the Current Mute Status:

To Request the settings from a Device we will use the "Get Message" Protocol as well as the Mute Option. This option will be completely specified in HEX (Base 16).

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Value	0x8F	0x8F	0x8F	0x8F	0x00	0x14	0xAA	0x00	0x00	0x01	0x00	0x00	0x00	0x00	0x02	0x02	0x00	0x00	0xFF

Bytes 1-4 are the Header Information required to talk to the Get Parameter Server.
Bytes 5-10 are the MAC Address of the device you wish to access. These are important since you are not required to send the UDP Message to a particular address; instead you may broadcast the message. This Mac will be used to determine which device will change.
Bytes 11-14 are basically reserved again by the Get Parameter Header.
Bytes 15-18 are where the Mute Option is specified:

- Byte 15 - 0x02 is the value for the Mute Option.
- Byte 16 - is part of the mute option and specifies the number of bytes to follow that will also be part of the mute option. This length is important and must be adjusted when a length of a message is changed.
- Byte 17 – is also part of the mute option and specifies if an Input or Output Channel is to be accessed.
- Byte 18 – is the Channel number to be accessed (Either Input or Output). This message has specified to access Output Number 1. (Or Amplifier Channel number 1).

Byte 19 – is the End Option Byte. This byte must always be placed after the last option.

In this message we only requested 1 option, the mute status for Output 1. Also since we are requesting a parameter we can truncate the mute message and not specify a dummy value. This is why the message length is only two bytes long instead of three. The Message should be sent to the device via UDP and on port 3100. The device will then reply to the port from which it received the request with the following Message:

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Value	0x8F	0x8F	0x8F	0x8F	0x00	0x14	0xAA	0x00	0x00	0x01	0x01	0x00	0x00	0x00	0x02	0x03	0x00	0x00	0x01	0xFF

Note: The Reply Message is very similar to the request (only a few bytes have changed)
Byte 11 – Now is 0x01 indicating this is a reply Message.
Byte 16 – Has been changed to 0x03 as the Mute option is now has 3 bytes following the length.
Byte 19 – Has been added to the mute message. This indicates the current Mute status. 0x01 indicates the Output is currently muted.
Byte 20 – This is the End Option Byte. It has been moved by 1, due to the Mute option containing an extra byte.

Multiple Payload Message:

It is also possible to request multiple options in a single message. The following message will request the mute status for outputs 1 & 2.

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Value	0x8F	0x8F	0x8F	0x8F	0x00	0x14	0xAA	0x00	0x00	0x01	0x00	0x00	0x00	0x00	0x02	0x02	0x00	0x00
Byte	**19**	**20**	**21**	**22**	**23**													
Value	0x02	0x02	0x00	0x01	0xFF													

Changing the Current Mute Status:

To Change a setting on an Ashly Network device is very similar to requesting a parameter. The header changes; however, the general Structure of the options is similar. Also, the options must be fully specified (i.e. the new parameter values specified).

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Value	0xAA	0xAA	0xAA	0xAA	0x00	0x14	0xAA	0x00	0x00	0x01	0x64	0x65	0x66	0x61	0x75	0x6C
Byte	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
Value	0x74	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x01	0x00	0x00	0x02	0x03
Byte	**33**	**34**	**35**	**36**												
Value	0x00	0x00	0x01	0xFF												

Bytes 1-4 are the Header, which is used to talk to the Set Parameter Server.
Bytes 5-10 are the MAC Address of the device you wish to access. These are important since you are not required to send the UDP Message to a particular address. Instead, you may broadcast the message. This MAC will be used to determine which device will change.
Bytes 11-18 are the user name to use for security. Here, it is entered as “default “
Bytes 19-26 is the password for the user, the default user has no password so it is left as all 0x00s. If another user is specified the password for that user should be entered here.
Bytes 27-28 are the message number, this is used for the ACK from the device however it can be ignored and set to 0x00s
Byte 29 is the ACK status of the message. This should be set to 0x00 indicating original message.
Byte 30 is reserved, set to 0x00
Bytes 31- 35 are the Mute Option:
- Byte 31 – 0x02 = Mute Option Number
- Byte 32 -- the number of bytes to follow in the option
- Byte 33 – the channel type 0 = Output Channel
- Byte 34 – the Channel number (0 = channel 1)
- Byte 35 – the new mute status, (1 = Muted)

Byte 36 – is the required end option.

Mixer Mute/Gain Messages
Supplement to *Ethernet Communications v5.1*

Revision 1.1, January 4, 2012

Purpose

As described in our *Ethernet Communications v5.1* document, the FUNC_MIXER_X_IN message allows changing the mixer's parameters (level, mute, and routing status) for each channel. Unfortunately, this message doesn't allow changing these parameters *independently* on *individual* mixer channels. To use FUNC_MIXER_X_IN, you have to fully specify *all* parameters for *all* channels of the mixer. This works if the controlling application knows all of the mixer's parameters and can resend all of them. But usually, this is not the case and you want to be able to vary individual mixer parameters independently of others.

To address this, two additional DSP messages were added to allow independent control over the level and mute parameters on individual channels. These messages work with all Ashly NE (Network Enabled) equipment with DSPs.

Notation

As in our other documentation, we notate hexadecimal numbers with a 0x prefix and do not use any prefix for decimal numbers. Our choice of using hexadecimal or decimal numbers depends on which is more natural for a particular kind of value. But numbers are numbers; the decimal value 255 is exactly the same as the hexadecimal value 0xFF. It's just two different notations to represent the exact same value; the value itself isn't hexadecimal or decimal. It's just a number.

Likewise, when you take the numbers in our documentation and use them with other systems, you're free to choose whatever representation is most natural or required by that system. The *values* are all the same, you're just changing *notation*.

If you need help with converting between hexadecimal and decimal numbers, there are many tools and techniques for this. The built-in calculator that comes with Windows allows converting between hexadecimal and decimal (if you put it in “Scientific” or “Programmer” mode).

Messages

Name	FUNC_MIXER_MUTE
Option #	0x74
Description	Allows independently muting and unmuting on one or more individual mixer channels. Level and routing parameters are not affected by this message.
Attributes	Write Only
Function Parameters Byte Description	

Byte #	*Value*	*Description*
1	0x81 (129)	OPT_DSP_FUNC_PARAMS
2	0xb (11)	Number of bytes to follow
3	0	DSP channel type (must be always be 0 for this message)
4	0 to 23	DSP channel number with mixer you want to control
5	0x74 (116)	FUNC_MIXER_MUTE
6	bits 31 to 24	A 32-bit value describing the set of channels to mute. If a bit is 1, the corresponding channel is muted. If a bit is 0, the corresponding channel is unchanged. All bits may be 0, which means that no channels are being muted.
7	bits 23 to 16	
8	bits 15 to 8	
9	bits 7 to 0	
10	bits 31 to 24	A 32-bit value describing the set of channels to unmute. If a bit is 1, the corresponding channel is unmuted. If a bit is 0, the corresponding channel is unchanged. All bits may be 0, which means that no channels are being unmuted.
11	bits 23 to 16	
12	bits 15 to 8	
13	bits 7 to 0	

Name	FUNC_MIXER_GAIN
Option #	0x75
Description	Allows independently controlling level on one or more individual mixer channels. Level and routing parameters are not affected by this message.
Attributes	Write Only
Function Parameters Byte Description	

Byte #	*Value*	*Description*
1	0x81 (129)	OPT_DSP_FUNC_PARAMS
2	0x9 (9)	Number of bytes to follow
3	0	DSP channel type (must be always be 0 for this message)
4	0 to 23	DSP channel number with mixer you want to control
5	0x75 (117)	FUNC_MIXER_GAIN
6	bits 31 to 24	A 32-bit value describing the set of channels to receive a change in level. If a bit is 1, the corresponding channel is set to the level specified in the following bytes. If a bit is 0, the corresponding channel's level is left alone. All bits may be 0, in which case this message does nothing.
7	bits 23 to 16	
8	bits 15 to 8	
9	bits 7 to 0	
10	bits 15 to 8	A 16-bit value with the encoded level. Note that the value 0 or a level below -50dB is *effectively* a mute, although the mute status is not affected.
11	bits 7 to 0	

Specifying Sets of Channels as 32-bit Numbers

Both of these messages use 32-bit numbers to encode the set of channels the messages operate on. The mapping between set of channels and bits is to take the channel number and ensure the corresponding bit is 1. Keep in mind that as with other messages, channel numbers start at zero (so channel 1 is represented in the message as 0, channel 2 is represented as 1, etc.).

For example, if you wanted to represent the set of channels 1, 3, 7, 16, and 22 the bits would look like this:

byte #	1								2								3								4							
bit #	31	30	29	28	27	26	25	24	23	22	**21**	20	19	18	17	16	**15**	14	13	12	11	10	9	8	7	**6**	5	4	3	**2**	1	**0**
	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	**1**	0	0	0	**1**	0	**1**

Converting these bits into bytes is easiest using hexadecimal numbers. Each digit in a hexadecimal number represents four bits according to this table:

Bits	Hexadecimal Value
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7

Bits	Hexadecimal Value
1 0 0 0	8
1 0 0 1	9
1 0 1 0	A
1 0 1 1	B
1 1 0 0	C
1 1 0 1	D
1 1 1 0	E
1 1 1 1	F

For example, if you had the bits 0101, that would be the hexadecimal value 5.

The process of converting these 32-bit numbers to 4 bytes is then to collect bits left to right, four at a time, find the hexadecimal value, and append these together. So taking the above example, the bits in groups of four are 0000, 0000, 0010, 0000, 1000, 0000, 0100, 0101. Looking up the hexadecimal values, that is 00208045. Then, take pairs of the hexadecimal digits and you have the four bytes: 0x00, 0x20, 0x80, 0x45.

This is only one way of converting the 32-bit numbers to bytes. Use whatever method makes sense to you.

Specifying Level

Ashly encodes level as a 16-bit (two byte) value. This value's range is from -50.0dB to +12.0dB with tenth-dB precision. The process for converting a value in dB to the encoded form is to take the dB value, multiply by 10, and add 8192. For example, the encoded form of -6.3dB would be -6.3dB * 10 + 8192 = 8129. You can then take that encoded value and convert it to the two bytes by converting to hexadecimal (0x1FC1). So the first byte is 0x1F and the second byte is 0xC1.

Note that any dB value below -50dB doesn't pass any audio. This is effectively a mute, although the channel's mute status does not change. An encoded value of 0 also passes no audio.

Document Revision History

Revision	Date	Changes
1.0	December 21, 2011	Initial release
1.1	January 4, 2012	Added “attributes” to point out these are write-only messages.