PowerShareX

amplifiers

UDP Control Protocol Guide

Revised 2024-06-06

Table of Contents

1 Introduction

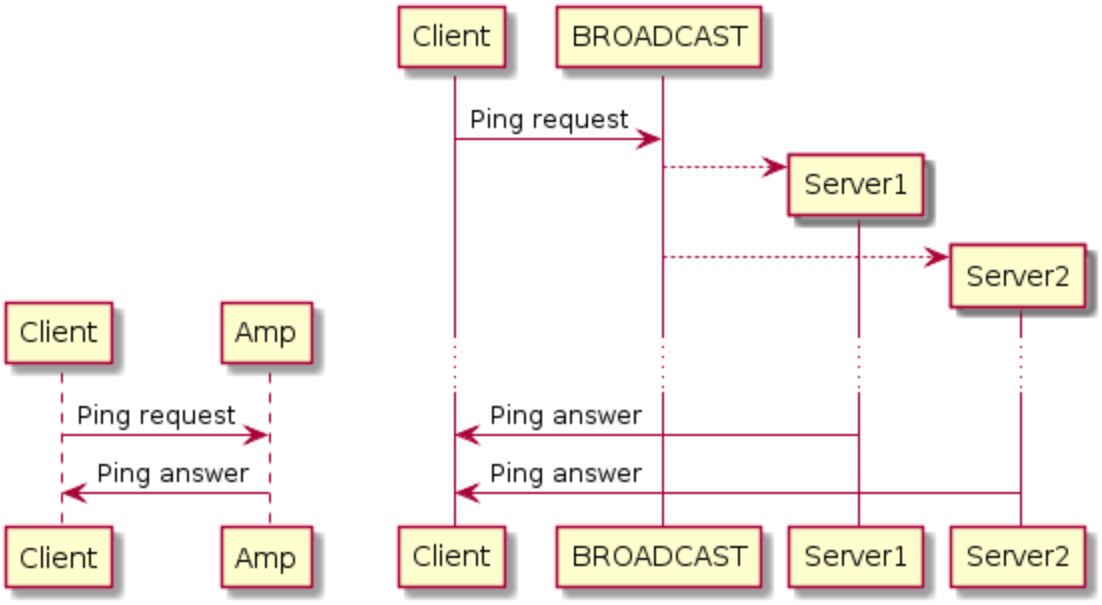
This document describes the protocol required to interact with Bose Professional's PowerShareX amplifiers when a Bose Professional DSP (ControlSpace EX-1280, ControlSpace ESP-880A, etc.) is not present in the design but monitoring or limited control is desired.

2 Protocol

It is possible to interact with any amplifier using a UDP protocol. Each time the amplifier receives a formatted message it replies with a formatted answer to the IP address (single-cast) that originated the request.

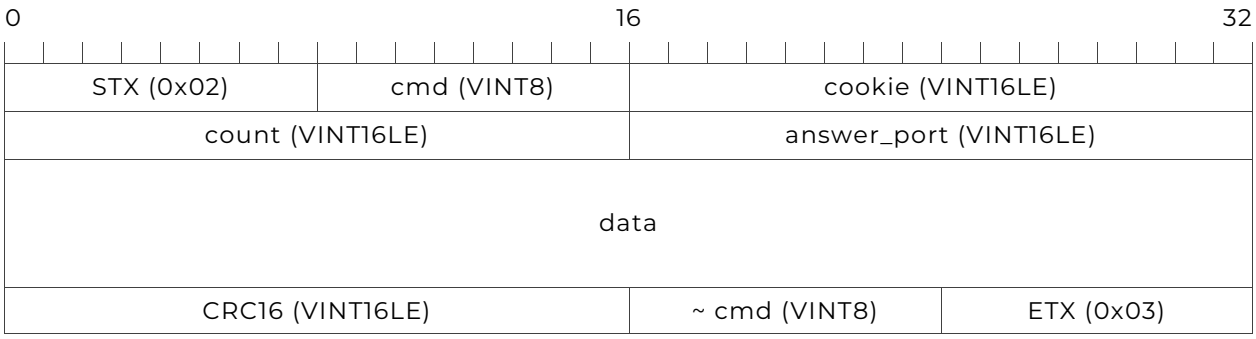
The PowerShareX amplifier manages both broadcast and single-cast requests coming to its own IP address, port 1234. It will answer using the port specified in the request. If the requested port is 0, it will answer using port 1234.

This document details instructions that also apply to 8-channel amplifiers. Bose Professional PowerShareX amplifiers as they are 4-channel amplifiers, so use only 0–3 (**0** is Channel 1, **1** is Channel 2, **2** is Channel 3, and **3** is Channel 4).



The PowerShareX amplifier manages more than one request at the same time. Any incoming request is marked with a cookie that the power amplifier uses to create the response. The device will not check if more than one request with the same cookie is in progress at the same time. It is up to the client to manage the cookie field according to its scope.

Each message (request or answer) is formatted based on this scheme:



STX: 1 byte delimiter (0x02)

cmd: 1 byte (0–127 for the request, 128–255 for the answer)

cookie: any 16-bit value

answer_port: 0 for answers, port at which the device will reply for requests (if left to the default value 0, port 1234 is used)

count: the size in byte of the next data field (its unsigned 16-bit value in little endian format).

data: any data (empty is valid)

crc16: the CRC16 of the data field (0 for empty data). This is the crc16 defined with the following polynomial: x16 + x15 + x2 + 1. Example CRC16 of "123456789" is 0xBB3D.)

~ cmd: the complement a1 of cmd

ETX: 1 byte delimiter (0x03)

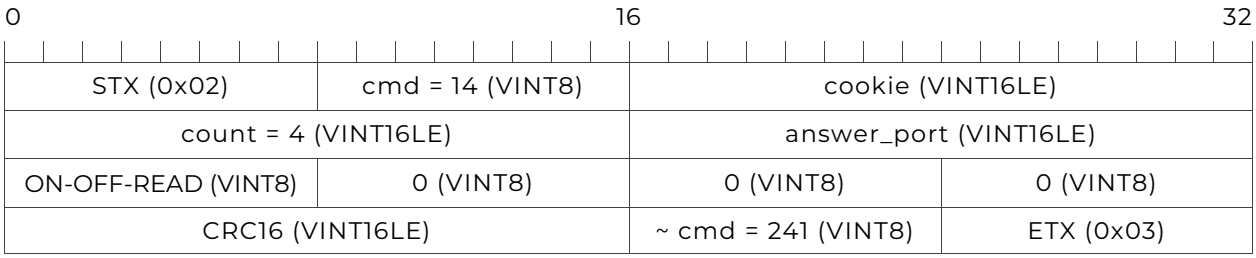
3 Commands

List of accepted cmd is:

Command	cmd	~ cmd	Notes
STANDBY	14	241	
PING	0	255	
READGM	1	254	
WRITEINMUTE	2	253	
WRITEOUTMUTE	3	252	
WRITEINGAIN	4	251	
WRITEOUTGAIN	5	250	
WRITEMULTI	8	247	
READALARMS	13	242	
READALLALARMS	15	240	Deprecated. Use READALLALARMS2.
READPILOTTONEGENERATOR	17	238	
READPILOTTONEDETECTION	18	237	
READLOADMONITOR	19	236	
READLOADDETECT	20	235	
SETPILOTTONEGENERATOR	21	234	
SETPILOTTONEDETECTION	22	233	
SETLOADMONITOR	23	232	
SETLOADDETECT	24	231	
READALLALARMS2	25	230	
SOURCEMETER	27	228	
OUTPUTMETER	28	227	
READLOADSTATUS	29	226	

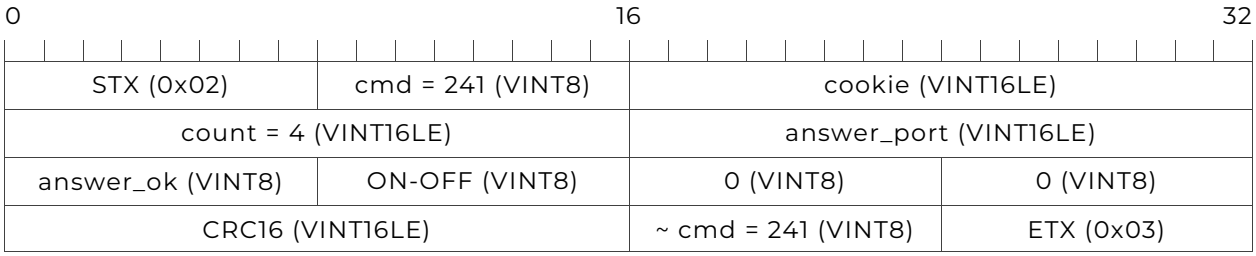
1 STANDBY

This command is used to set or read the standby status. The command is formatted as follows:



ON-OFF-READ: 0 to read the STANDBY state without changing it, 1 to set standby OFF (amplifier operative), 2 to set standby to ON (amplifier in standby, not operative).

The answer is:



answer_ok: 1 indicates a valid answer.

ON-OFF: 2 indicates STANDBY OFF (amplifier operative), 1 indicates STANDBY ON (amplifier not operative)

Example:

In hex code: The diagram below shows the command in hex code, which is used to enter the command.

POWER ON (STANDBY OFF): **02** **0E** 3D 00 **04 00** 88 13 **01 00 00 00** **01 FC** F1 **03**

02 — START (STX)

0E — 14 (COMMAND SET STANDBY)

3D 00 — COOKIE (VALUE NOT SPECIFIC, TAG FOR COMMAND)

04 00 — COUNT (4)(LITTLE ENDIAN)

88 13 — ANSWER PORT (VALUE NOT SPECIFIC, 0000 FOR DEFAULT)

01 00 00 00 — DATA (VALUE 01 = POWER ON/STANDBY OFF)

01 FC — CRC16 (DATA FIELD 01 00 00 00) (LITTLE ENDIAN)

F1 — 241 (COMMAND SET STANDBY REVERSE) (255-14)

03 — END (ETX)

Structuring commands: Despite the first-glance complexity, sending commands to PowerShareX is simple. Most fields are static, except for the command, command reverse, the command data, and the CRC16 checksum of the command data.

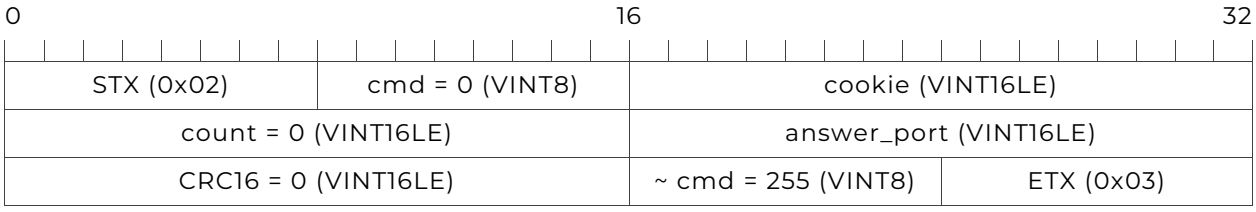
- The command in this example is 14 and the reverse is 241 (255 minus 14).
- Data is 01 00 00 00 for POWER ON, and 02 00 00 00 for POWER OFF.
- The CRC16 checksum is calculated from the data field, and done as CRC16/ARC (remember to use hex code as the input, 01000000 = FC01).

  An online calculator can be used to calculate the checksum, e.g., online CRC-8 CRC-16 CRC-32 calculator (**crccalc.com**).
- The cookie and answer port lines are irrelevant for this command and not used in this example.

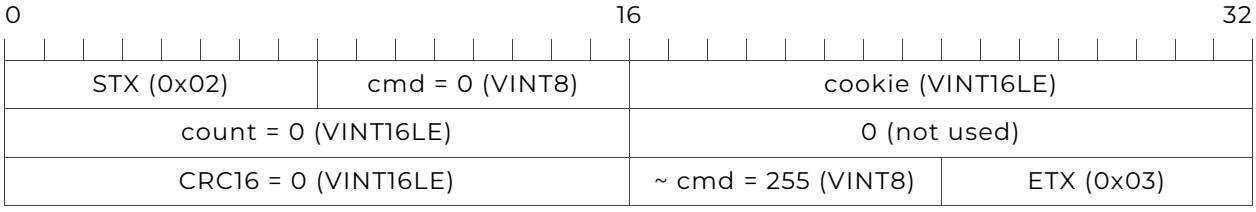
In this case, where the values for COUNT and CRC16 are presented in little-endian, the 2 hex code fields are reversed (FC 01 in little-endian is 01 FC).

2 PING

This command is used only to test if the amplifier is alive. The command is formatted as follows:

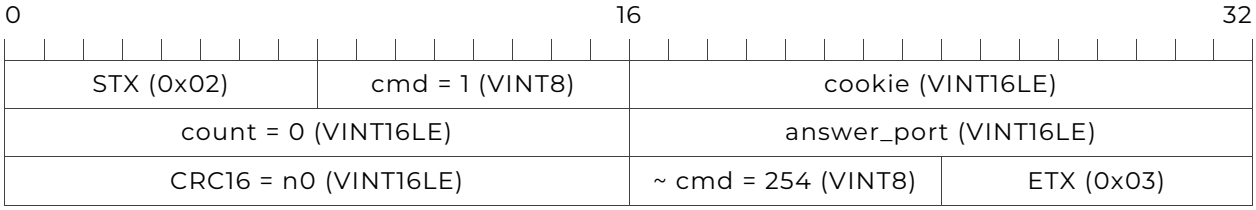


The answer is:



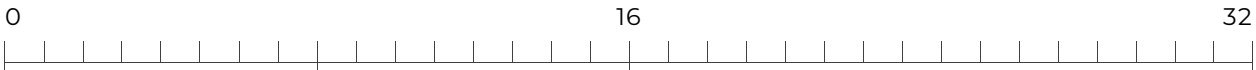
3 READGM

This command is used to read all the Gains and Mutes status inside the amplifier. The command is formatted as follows:



STX (0x02)	cmd = 1 (VINT8)	cookie (VINT16LE)	
count = 0 (VINT16LE)		answer_port (VINT16LE)	
CRC16 = n0 (VINT16LE)		~ cmd = 254 (VINT8)	ETX (0x03)

The answer is:



STX (0x02)	cmd = 254 (VINT8)	cookie (VINT16LE)	
count = 52 (VINT16LE)		0 (not used)	
answer_ok (VINT8)	num_channels (VINT8)	0 (VINT8)	0 (VINT8)
INGAIN0 (INT16LE)		INGAIN1 (INT16LE)	
INGAIN2 (INT16LE)		INGAIN3 (INT16LE)	
INGAIN4 (INT16LE)		INGAIN5 (INT16LE)	
INGAIN6 (INT16LE)		INGAIN7 (INT16LE)	
OUTGAIN0 (INT16LE)		OUTGAIN1 (INT16LE)	
OUTGAIN2 (INT16LE)		OUTGAIN3 (INT16LE)	
OUTGAIN4 (INT16LE)		OUTGAIN5 (INT16LE)	
OUTGAIN6 (INT16LE)		OUTGAIN7 (INT16LE)	
INMUTE0 (VINT8)	INMUTE1 (VINT8)	INMUTE2 (VINT8)	INMUTE3 (VINT8)
INMUTE4 (VINT8)	INMUTE5 (VINT8)	INMUTE6 (VINT8)	INMUTE7 (VINT8)
OUTMUTE0 (VINT8)	OUTMUTE1 (VINT8)	OUTMUTE2 (VINT8)	OUTMUTE3 (VINT8)
OUTMUTE4 (VINT8)	OUTMUTE5 (VINT8)	OUTMUTE6 (VINT8)	OUTMUTE7 (VINT8)
CRC16 (VINT16LE)		~ cmd = 1 (VINT8)	ETX (0x03)

where:

answer_ok: 1 indicates valid answer.

num_channels: is the number of output channels managed by the amplifier. So only the gain/mute status for channel less than num_channels has to be considered valid.

INGAINX: is the input gain of the channel at the output of the matrix X

OUTGAINX: is the output gain of the channel at the output block X in cents of dB (from -6000 to 15000 → -60db to +15db)

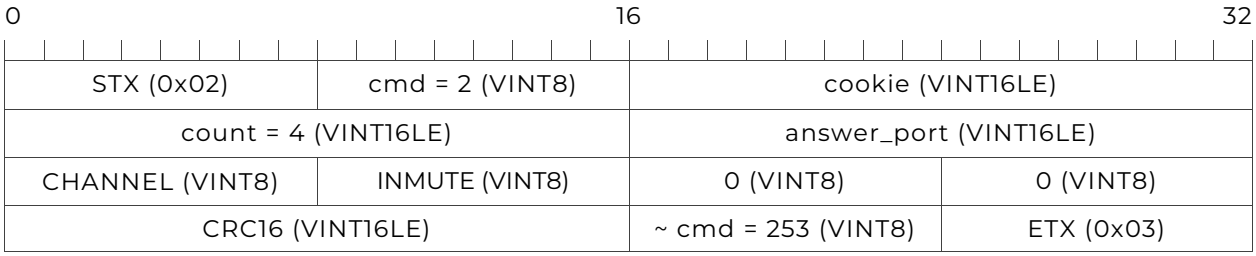
INMUTEX: is the mute status of the channel at the output of the matrix X

OUTMUTEX: is the mute status of output at the output block X in cents of dB (from -6000 to 15000 → -60db to +15db)

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

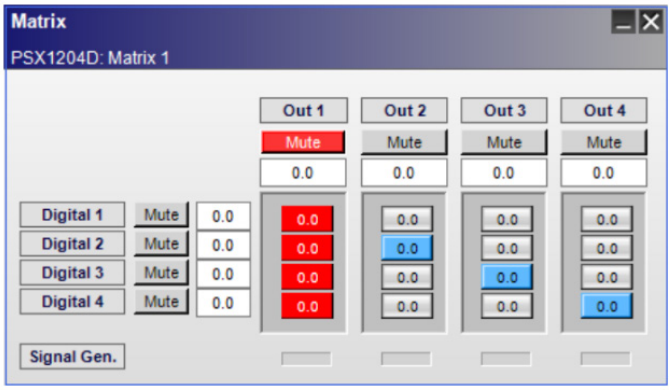
4 WRITEINMUTE

This command is used to set a mute status for one channel at the output of the matrix. The command is formatted as follows:



0		16	32
STX (0x02)	cmd = 2 (VINT8)	cookie (VINT16LE)	
count = 4 (VINT16LE)		answer_port (VINT16LE)	
CHANNEL (VINT8)	INMUTE (VINT8)	0 (VINT8)	0 (VINT8)
CRC16 (VINT16LE)		~ cmd = 253 (VINT8)	ETX (0x03)

Note: A mute command sent to channel 0 (channel 1) would look like this in ControlSpace Designer..



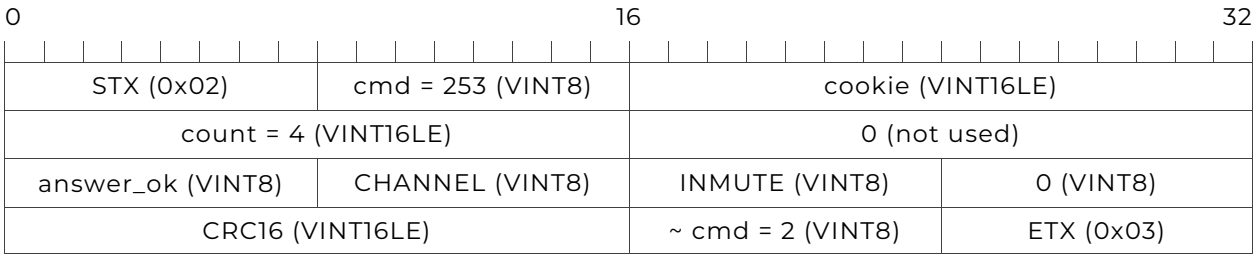
where:

CHANNEL: It ranges from 0 to the number of channels supported, and it is the channel to be muted

INMUTE: 0 to unmute, 1 to mute.

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:



0		16	32
STX (0x02)	cmd = 253 (VINT8)	cookie (VINT16LE)	
count = 4 (VINT16LE)		0 (not used)	
answer_ok (VINT8)	CHANNEL (VINT8)	INMUTE (VINT8)	0 (VINT8)
CRC16 (VINT16LE)		~ cmd = 2 (VINT8)	ETX (0x03)

where:

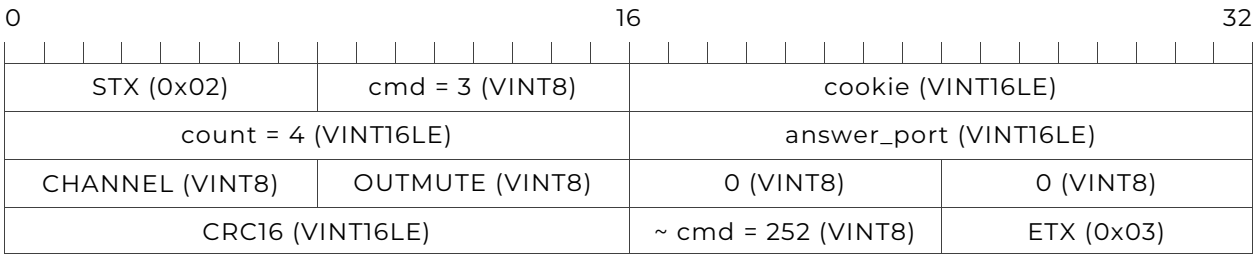
answer_ok: 1 indicates a valid answer.

CHANNEL: has to be the same of the CHANNEL field inside the request

INMUTE: has to be the same of the INMUTE field inside the request.

5 WRITEOUTMUTE

This command is used to set a mute status for one output channel. The command is formatted as follows:



0		16	32
STX (0x02)	cmd = 3 (VINT8)	cookie (VINT16LE)	
count = 4 (VINT16LE)		answer_port (VINT16LE)	
CHANNEL (VINT8)	OUTMUTE (VINT8)	0 (VINT8)	0 (VINT8)
CRC16 (VINT16LE)		~ cmd = 252 (VINT8)	ETX (0x03)

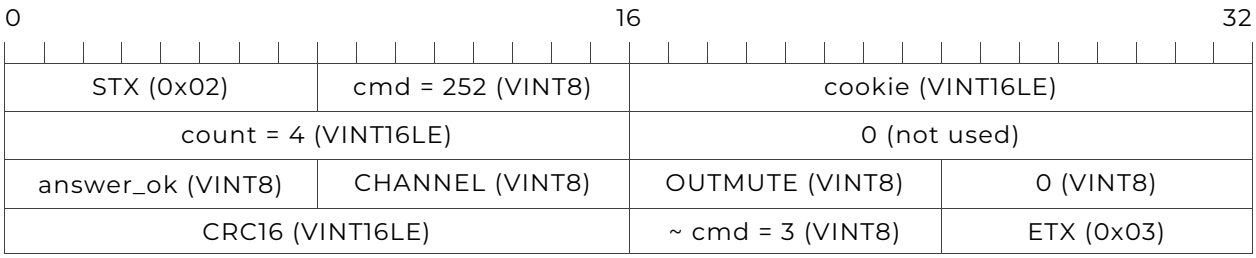
where:

CHANNEL: is from 0 to number of channels supported, and is the channel to mute

OUTMUTE: is 0 to unmute 1 to mute.

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:



0		16	32
STX (0x02)	cmd = 252 (VINT8)	cookie (VINT16LE)	
count = 4 (VINT16LE)		0 (not used)	
answer_ok (VINT8)	CHANNEL (VINT8)	OUTMUTE (VINT8)	0 (VINT8)
CRC16 (VINT16LE)		~ cmd = 3 (VINT8)	ETX (0x03)

where:

answer_ok: 1 indicates valid answer.

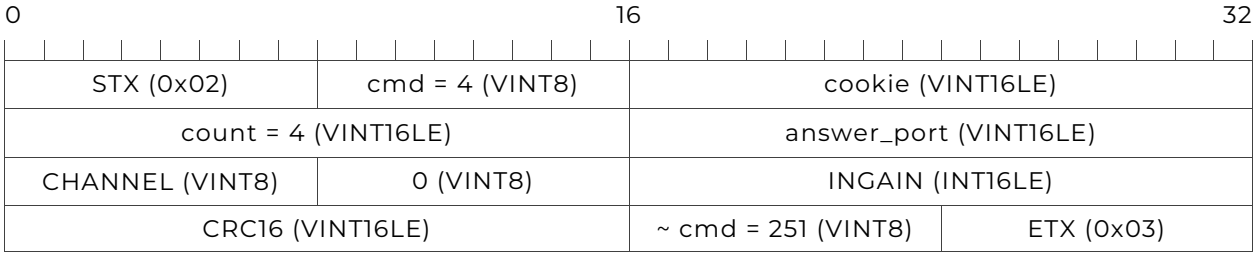
CHANNEL: has to be the same of the CHANNEL field inside the request

OUTMUTE: has to be the same as the OUTMUTE field inside the request

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

6 WRITEINGAIN

This command is used to set a gain status for one channel at the output of the matrix. The command is formatted as follows:



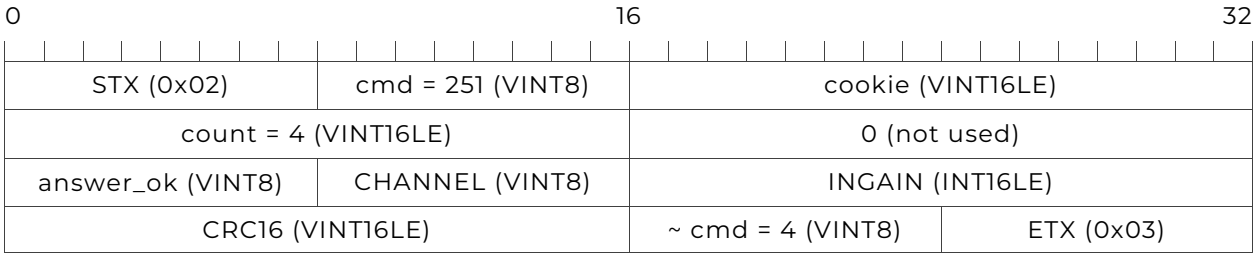
where:

CHANNEL: is from 0 to number of channels supported, and is the channel to control

INGAIN: is number in cents of dB (from -6000 to 15000 → -60db to +15db)

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:



where:

answer_ok: 1 indicates valid answer.

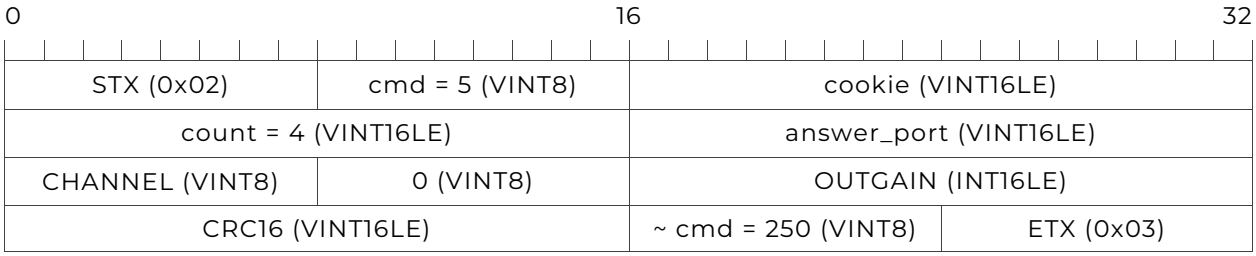
CHANNEL: has to be the same of the CHANNEL field inside the request

INGAIN: has to be the same of the INGAIN field inside the request

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

7 WRITEOUTGAIN

This command is used to set a gain status for one output channel. The command is formatted as follows:



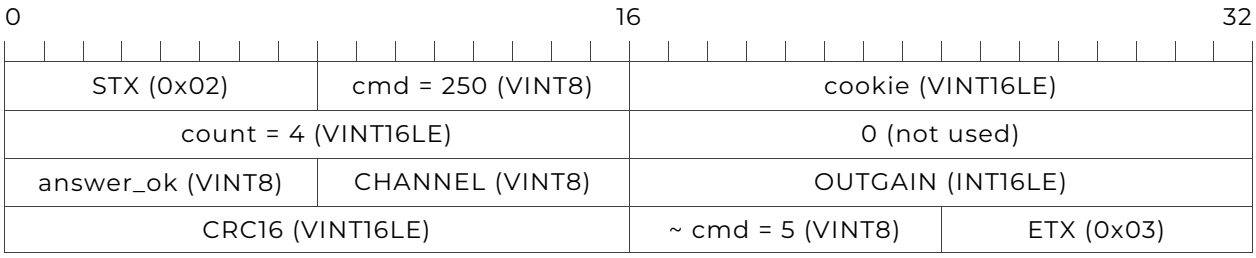
where:

CHANNEL: is from 0 to number of channels supported, and is the channel to control

OUTGAIN: is number in cents of dB (from -6000 to 15000 → -60db to +15db)

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:



where:

answer_ok: 1 indicates valid answer.

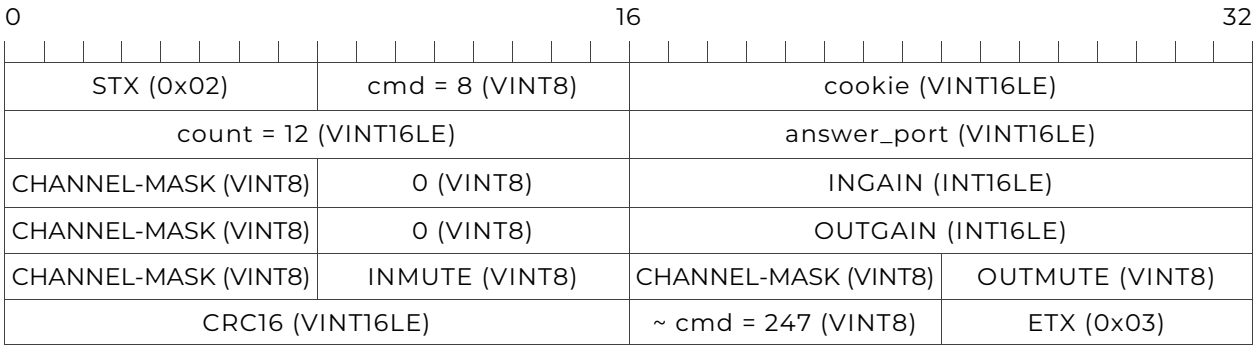
CHANNEL: has to be the same of the CHANNEL field inside the request

OUTGAIN: has to be the same of the OUTGAIN field inside the request

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

8 WRITEMULTI

This command is used to set the same mute/gain value to more than one channel. The command is formatted as follows:



STX (0x02)	cmd = 8 (VINT8)	cookie (VINT16LE)	
count = 12 (VINT16LE)		answer_port (VINT16LE)	
CHANNEL-MASK (VINT8)	0 (VINT8)	INGAIN (INT16LE)	
CHANNEL-MASK (VINT8)	0 (VINT8)	OUTGAIN (INT16LE)	
CHANNEL-MASK (VINT8)	INMUTE (VINT8)	CHANNEL-MASK (VINT8)	OUTMUTE (VINT8)
CRC16 (VINT16LE)		~ cmd = 247 (VINT8)	ETX (0x03)

where:

INGAIN, **OUTGAIN**, **INMUTE**, **OUTMUTE** have the same meaning as the previous write command.

CHANNEL-MASK: is a bit array where bitX=1 indicates that the channel X has to be configured

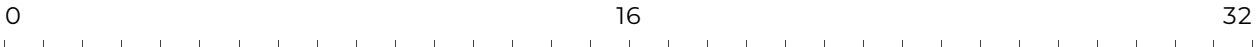
(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:

STX (0x02)	cmd = 247 (VINT8)	cookie (VINT16LE)	
count = 4 (VINT16LE)		0 (not used)	
answer_ok (VINT8)	0 (VINT8)	0 (VINT8)	0 (VINT8)
CRC16 (VINT16LE)		~ cmd = 8 (VINT8)	ETX (0x03)

9 READALARMS

This command is used to read alarms and metering live status. The command is formatted as follows:



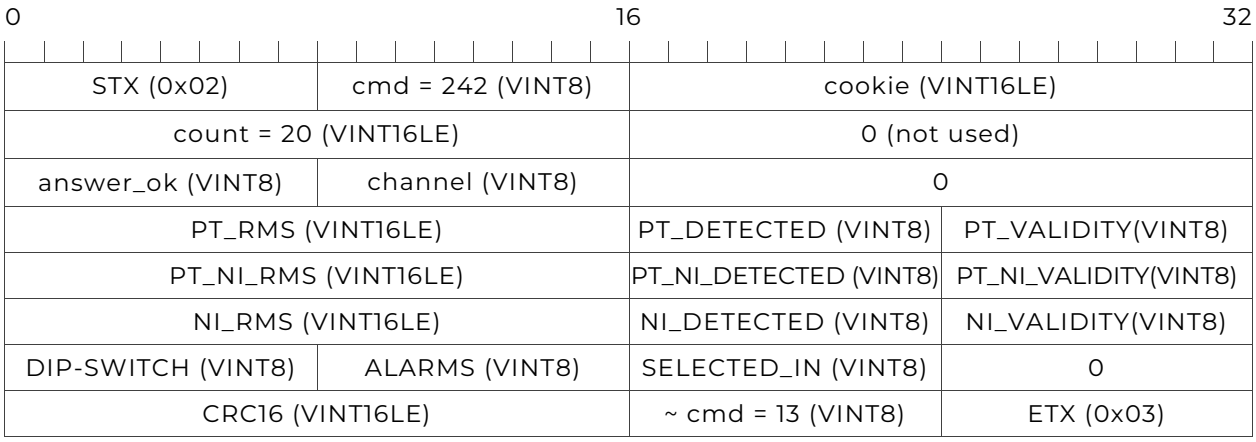
STX (0x02)	cmd = 13 (VINT8)	cookie (VINT16LE)	
count = 4 (VINT16LE)		answer_port (VINT16LE)	
channel (VINT8)			
CRC16 = 0 (VINT16LE)		~ cmd = 242 (VINT8)	ETX (0x03)

where:

CHANNEL: is from 0 to number of channels supported and is the channel to read

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:



where:

answer_ok: 1 indicates valid answer

channel: is the same channel received from request.

x_VALIDITY: 0 indicates that X_RMS and X_DETECTED is not valid.

x_DETECTED: 1 indicates detected.

PT_RMS: is the detected input pilot tone level in tenths of volts.

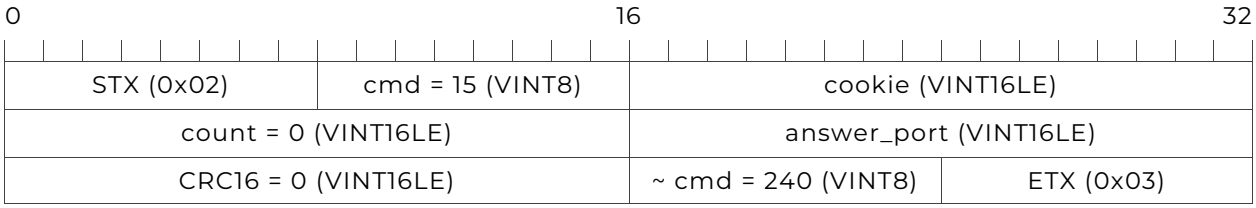
NI_RMS: is the measured broadband nominal impedance of the connected load, in tenths of Ohms.

SELECTED_IN: 0 for ANALOG, 1 for AES3, 2 for DANTE1-8, 3 for DANTE9-16. Any other value indicates that there is no information available.

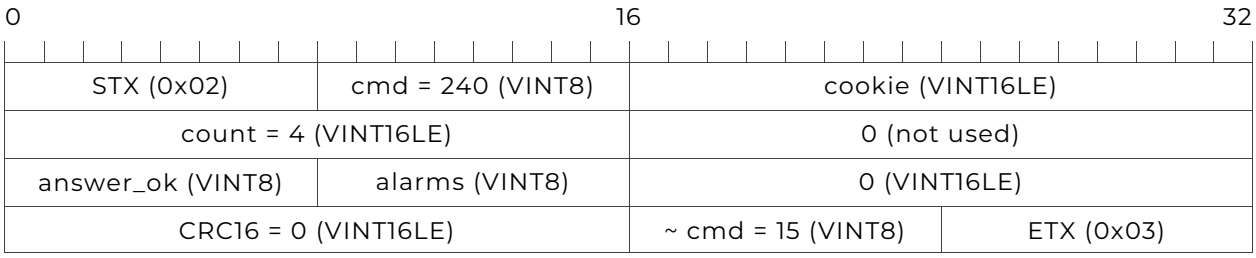
(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

10 READALLALARMS

This command is used to read alarms and metering live status (it is deprecated because it is possible to use READALLALARMS2). The command is formatted as follows:



The answer is:

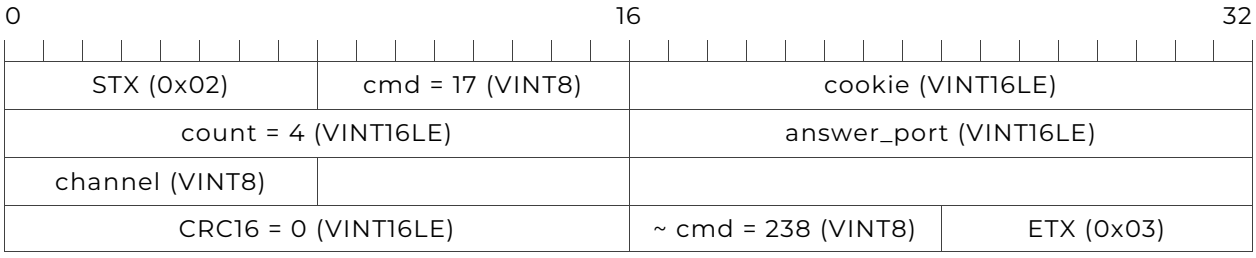


where:

answer_ok: 1 indicates valid answer.

11 READPILOTTONEGENERATOR

This command is used to read the Inner Pilot Tone Generator setting. The command is formatted as follows:

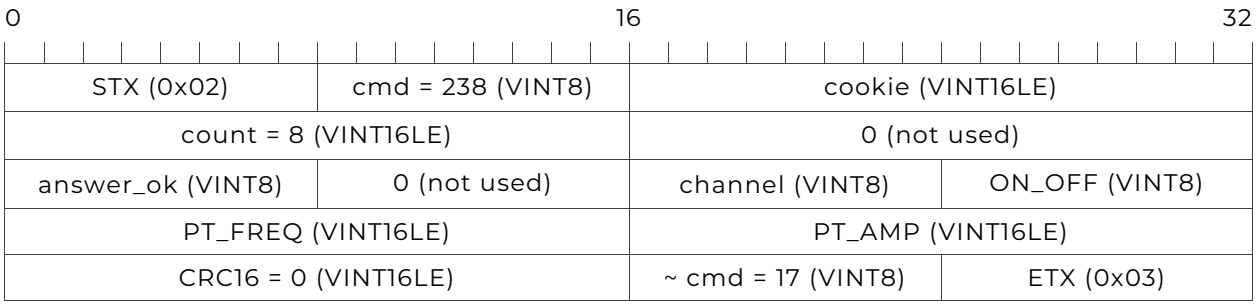


where:

CHANNEL: is from 0 to number of channels supported and is the channel to read

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:



where:

answer_ok: 1 indicates valid answer.

channel: is the same channel received from request.

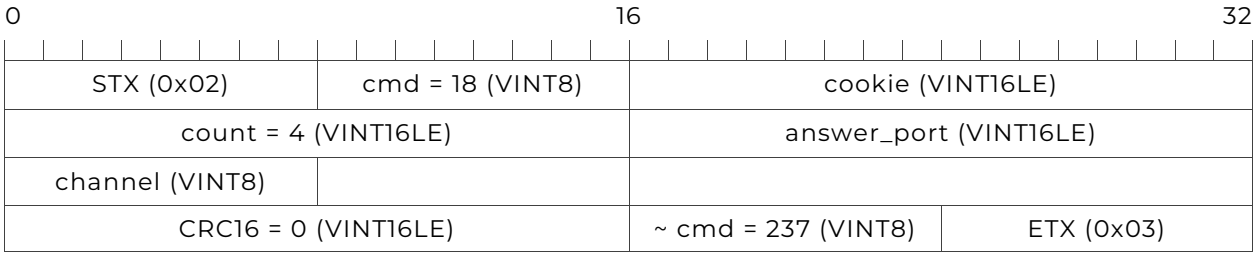
ON_OFF: 0 indicates OFF.

PT_FREQ: is the freq of the generated pilot tone (in Hz)

PT_AMP: is the amplitude pilot tone level in tenths of volts.

12 READPILOTTONEDETECTION

This command is used to read the Output Pilot Tone Detection setting. The command is formatted as follows:

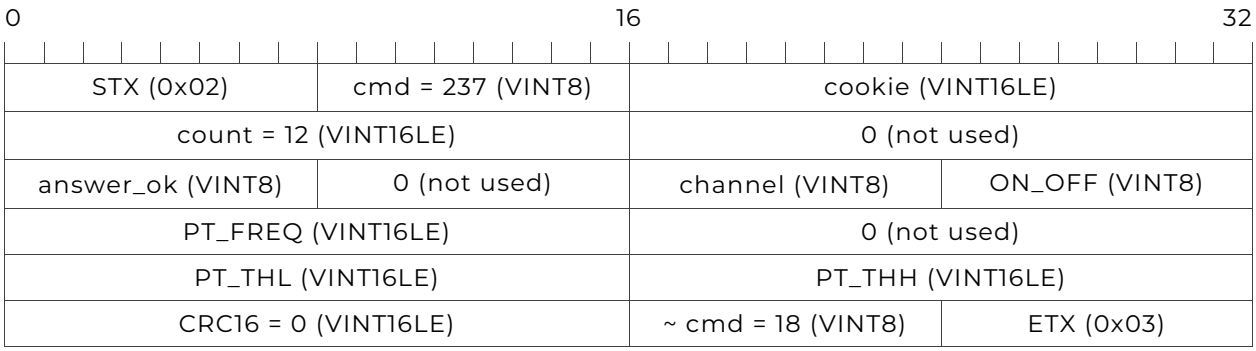


where:

CHANNEL: is from 0 to number of channels supported and is the channel to read

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:



where:

answer_ok: 1 indicates valid answer.

channel: is the same channel received from request.

ON_OFF: 0 indicates OFF.

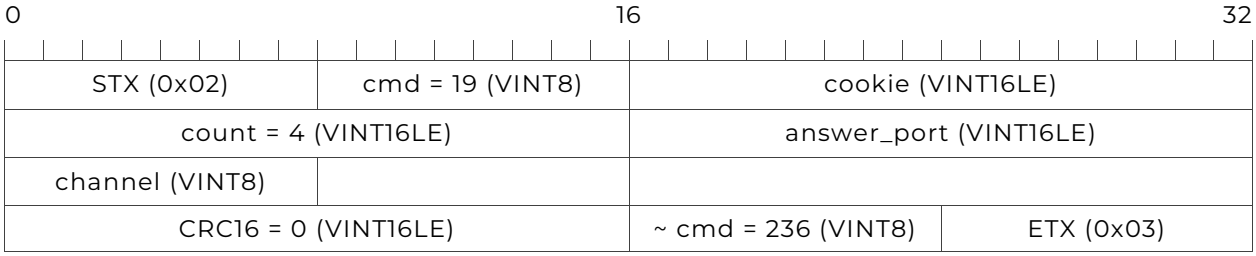
PT_FREQ: is the freq of the generated pilot tone (in Hz)

PT_THL: is low threshold (V_{RMS}) in tenths of volts.

PT_THH: is high threshold (V_{RMS}) in tenths of volts.

13 READLOADMONITOR

This command is used to read the Output Load Monitor setting. The command is formatted as follows:

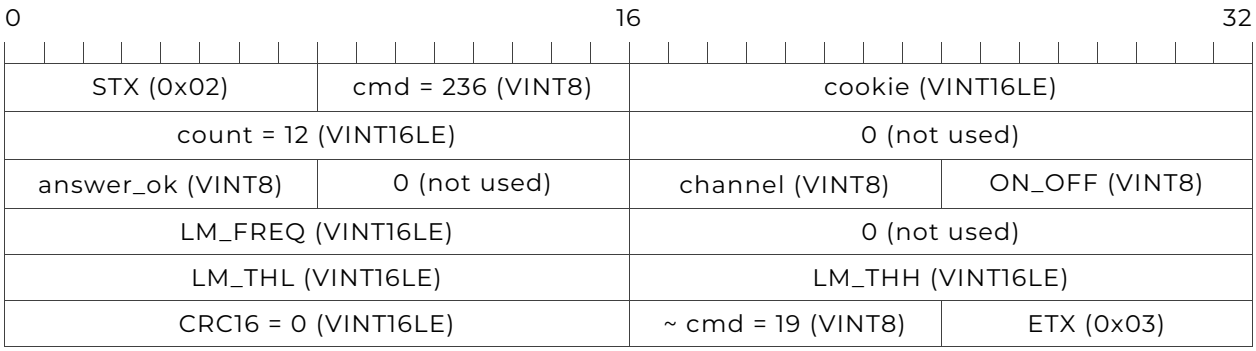


where:

CHANNEL: is from 0 to number of channels supported and is the channel to read

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:



where:

answer_ok: 1 indicates valid answer.

channel: is the same channel received from request.

ON_OFF: 0 indicates OFF.

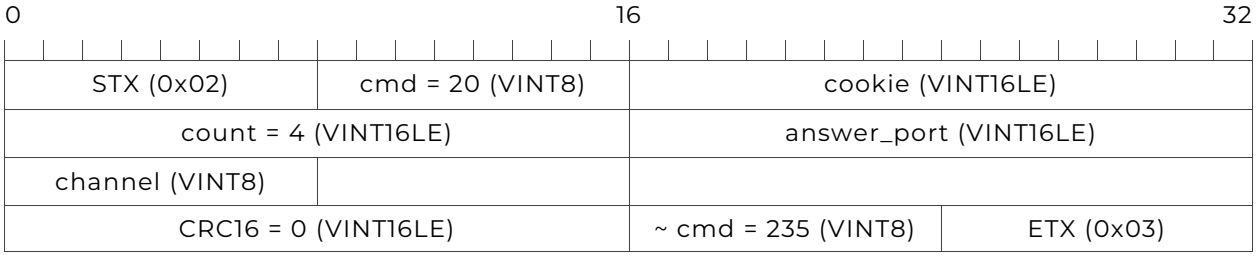
LM_FREQ: is the freq of the generated pilot tone (in Hz)

LM_THL: is low threshold (ohm) in tenths of ohms.

LM_THH: is high threshold (ohm) in tenths of ohms.

14 READLOADDETECT

This command is used to read the Inner Pilot Tone Generator setting. The command is formatted as follows:

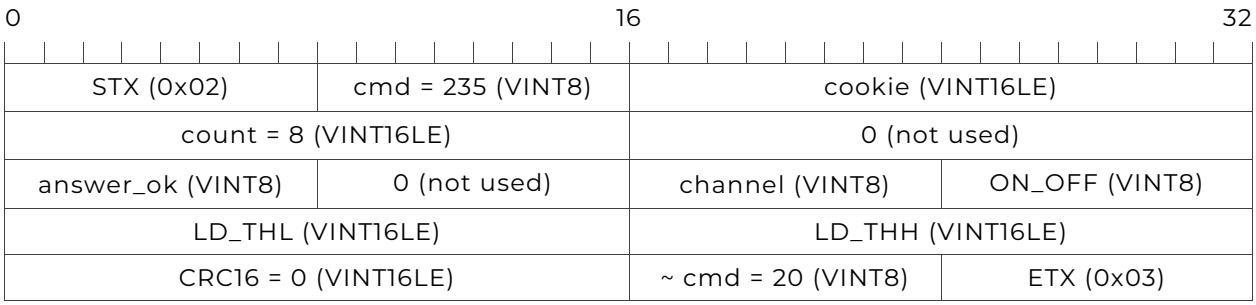


where:

CHANNEL: is from 0 to number of channels supported and is the channel to read

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:



0		16	32
STX (0x02)	cmd = 235 (VINT8)	cookie (VINT16LE)	
count = 8 (VINT16LE)		0 (not used)	
answer_ok (VINT8)	0 (not used)	channel (VINT8)	ON_OFF (VINT8)
LD_THL (VINT16LE)		LD_THH (VINT16LE)	
CRC16 = 0 (VINT16LE)		~ cmd = 20 (VINT8)	ETX (0x03)

where:

answer_ok: 1 indicates valid answer.

channel: is the same channel received from request.

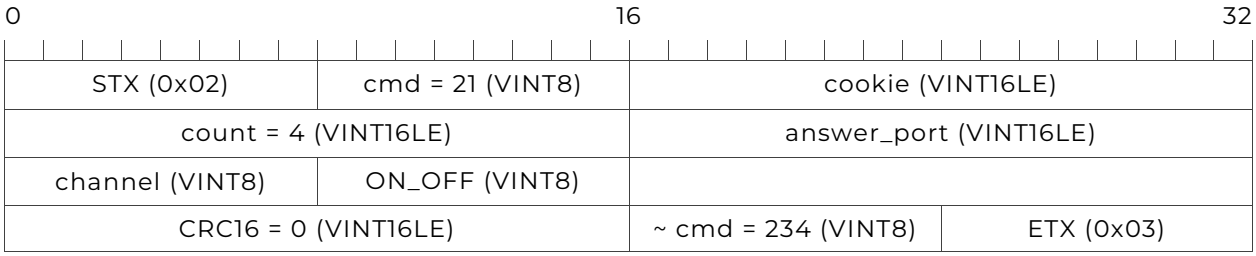
ON_OFF: 0 indicates OFF.

LD_THL: is low threshold (ohm) in tenths of ohms.

LD_THH: is high threshold (ohm) in tenths of ohms.

15 SETPILOTTONEGENERATOR

This command is used to set (on or off) Pilot Tone Generation. The command is formatted as follows:



0		16	32
STX (0x02)	cmd = 21 (VINT8)	cookie (VINT16LE)	
count = 4 (VINT16LE)		answer_port (VINT16LE)	
channel (VINT8)	ON_OFF (VINT8)		
CRC16 = 0 (VINT16LE)		~ cmd = 234 (VINT8)	ETX (0x03)

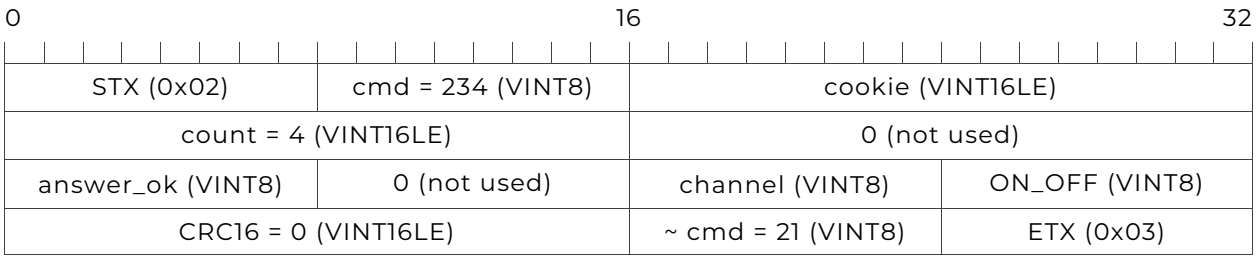
where:

CHANNEL: is from 0 to number of channels supported, and is the channel to read

ON_OFF: 0 indicates OFF.

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:



0		16		32
STX (0x02)		cmd = 234 (VINT8)	cookie (VINT16LE)	
count = 4 (VINT16LE)			0 (not used)	
answer_ok (VINT8)	0 (not used)		channel (VINT8)	ON_OFF (VINT8)
CRC16 = 0 (VINT16LE)			~ cmd = 21 (VINT8)	ETX (0x03)

where:

answer_ok: 1 indicates valid answer.

channel: is the same channel received from request.

ON_OFF: is the same channel received from request.

16 SETPILOTTONEDETECTION

See paragraph SETPILOTTONEGENERATOR but with cmd=22

17 SETLOADMONITOR

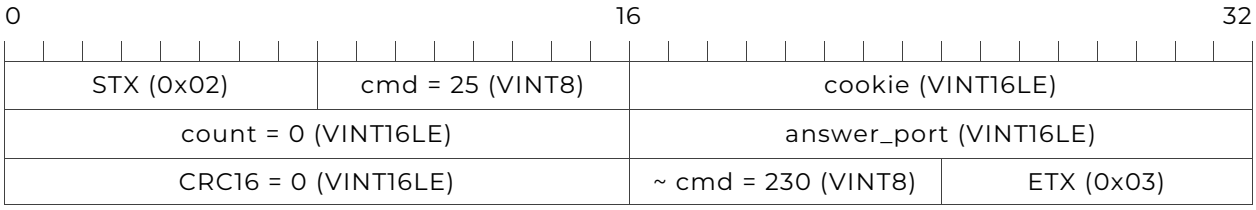
See paragraph SETPILOTTONEGENERATOR but with cmd=23

18 SETLOADDETECT

See paragraph SETPILOTTONEGENERATOR but with cmd=24

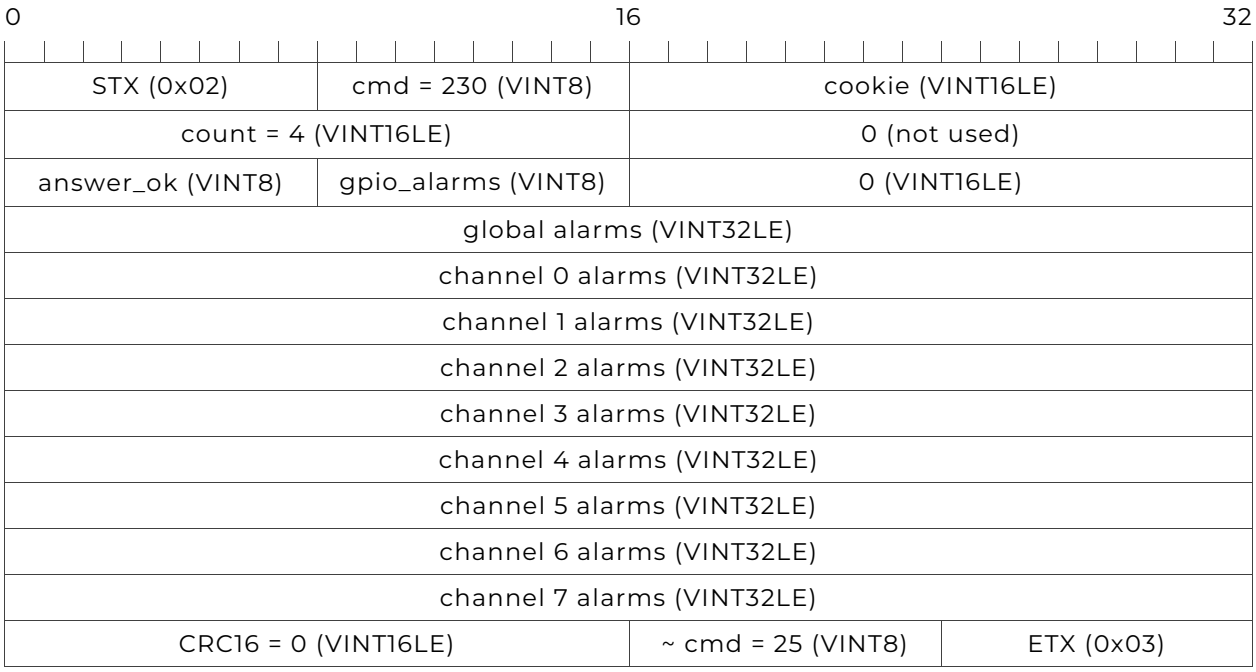
19 READALLALARMS2

This command is used to read alarms' status. The command is formatted as follows:



0		16		32
STX (0x02)		cmd = 25 (VINT8)	cookie (VINT16LE)	
count = 0 (VINT16LE)			answer_port (VINT16LE)	
CRC16 = 0 (VINT16LE)			~ cmd = 230 (VINT8)	ETX (0x03)

The answer is:



where:

answer_ok: 1 indicates valid answer.

global_alarms: (little endian)

bit 0 (LSB): mains phases detect error: set if triphase with missing phase, DC, or not available (only X)

bit 1: AD converter configuration fault

bit 2: DA converter configuration fault

bit 3: AUX voltage fault (only X)

bit 4: Digi board over-temperature → Implicit machine shutdown

bit 5: Power supply over-temperature (only X) → Implicit machine shutdown

bit 6: Fan fault → Implicit machine shutdown

bit 7: moderate over temperature (only X)

bit 8: high over temperature (only X)

bit 9-31: not used

channel X alarms:

bit 0 (LSB): input clip

bit 1: active thermal SOA (only X)

bit 3: over-temperature

bit 4: rail voltage fault

bit 5: AUX current fault (only X)

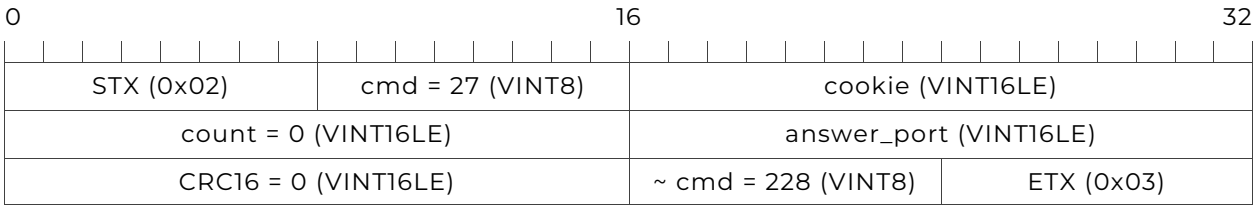
bit 6: other fault

bit 7: low load protection

bit 7-31: not used

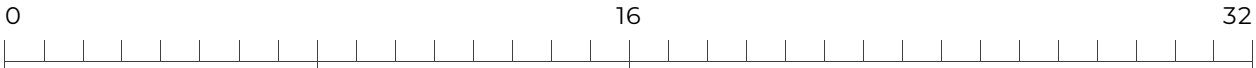
20 SOURCEMETER

This command is used to read source meter information. The command is formatted as follows:



STX (0x02)	cmd = 27 (VINT8)	cookie (VINT16LE)	
count = 0 (VINT16LE)		answer_port (VINT16LE)	
CRC16 = 0 (VINT16LE)		~ cmd = 228 (VINT8)	ETX (0x03)

The answer is:



STX (0x02)	cmd = 228 (VINT8)	cookie (VINT16LE)	
count = 8 (VINT16LE)		not used (VINT16LE)	
PRESENCE_CLIP_0 (VINT8LE)	PRESENCE_CLIP_1 (VINT8LE)	PRESENCE_CLIP_2 (VINT8LE)	PRESENCE_CLIP_3 (VINT8LE)
PRESENCE_CLIP_4 (VINT8LE)	PRESENCE_CLIP_5 (VINT8LE)	PRESENCE_CLIP_6 (VINT8LE)	PRESENCE_CLIP_7 (VINT8LE)
CRC16 (VINT16LE)		~ cmd = 27 (VINT8)	ETX (0x03)

where:

PRESENCE_CLIP_<x>: x is the input channels

bit 0 (LSB): is presence for source slot 0

bit 1: is clip for source slot 0

bit 2: is presence for source slot 1

bit 3: is clip for source slot 1

bit 4: is presence for source slot 2

bit 5: is clip for source slot 2

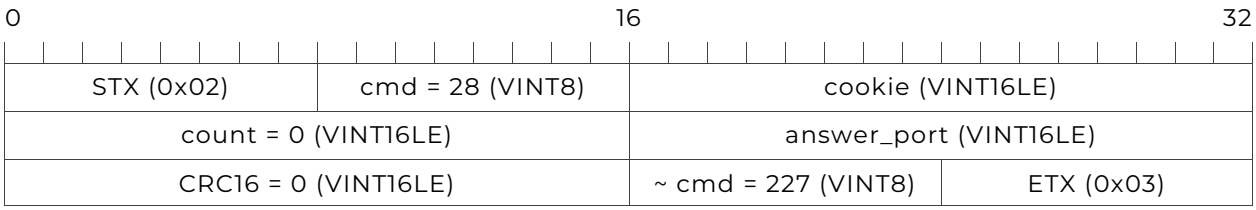
bit 6: is presence for source slot 3

bit 7: is clip for source slot 3

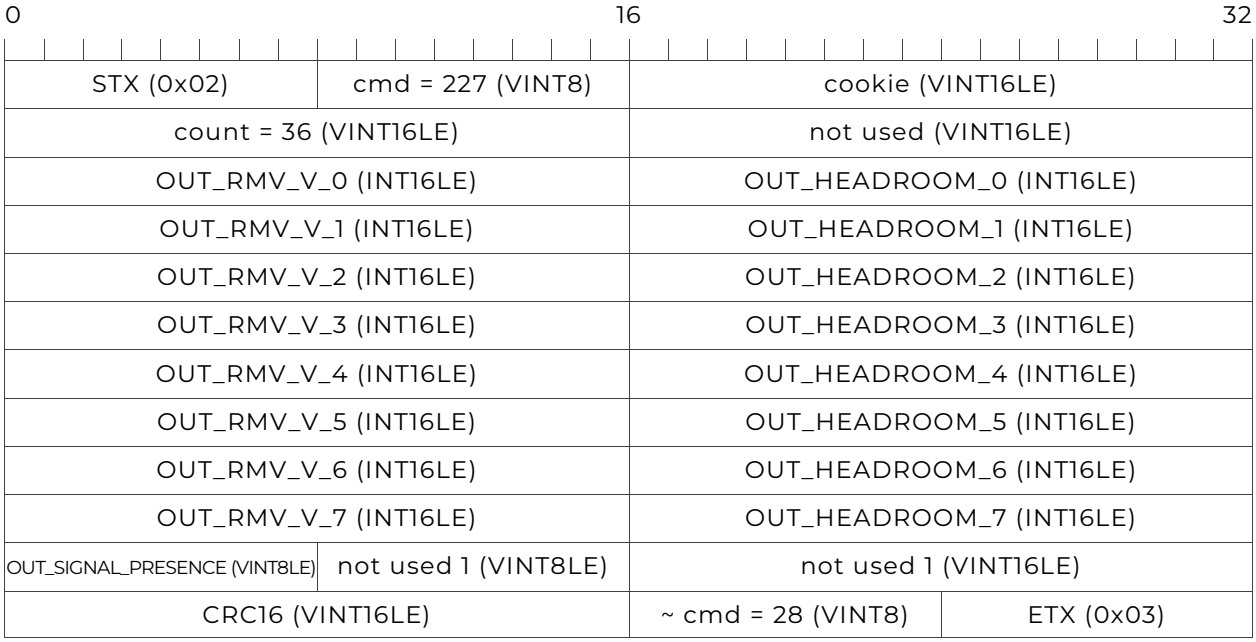
(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

21 OUTPUTMETER

This command is used to read output meter information. The command is formatted as follows:



The answer is:



where:

OUT_RMS_V<x>: is the output Voltage RMS meter of channel X in tenths of volt.

OUT_HEADROOM_<x>: is the output headroom of channel X in cents of db.

OUT_SIGNAL_PRESENCE:

bit 0 (LSB): is presence for out channel 0

bit 1: is presence for out channel 1

bit 2: is presence for out channel 2

bit 3: is presence for out channel 3

bit 4: is presence for out channel 4

bit 5: is presence for out channel 5

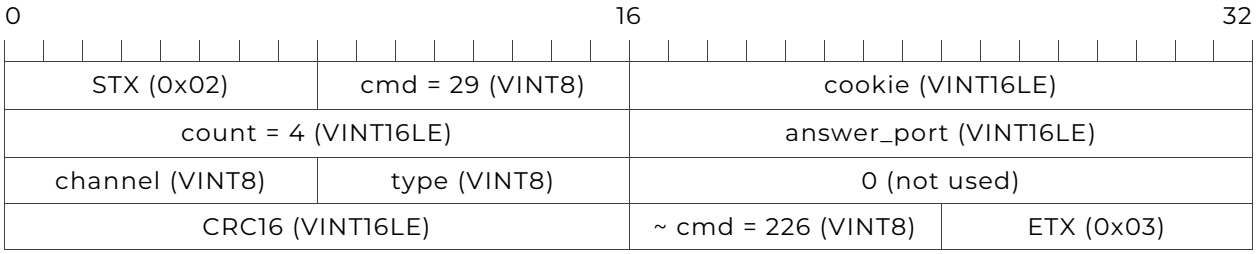
bit 6: is presence for out channel 6

bit 7: is presence for out channel 7

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

22 READLOADSTATUS

This command is used to read the status of the load monitor for a single channel. The command is formatted as follows:



where:

CHANNEL: is from 0 to number of channels supported, and is the channel to read

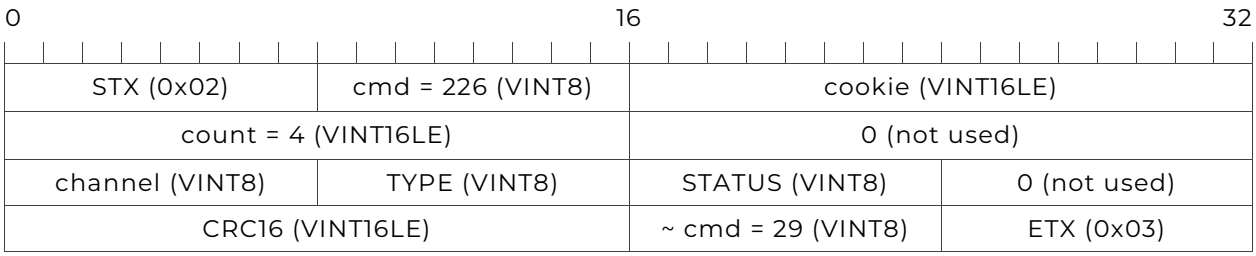
TYPE: indicates the type of information to get:

0 for the nominal impedance

1 for the load monitor

(**0** = Channel 1, **1** = Channel 2, **2** = Channel 3, **3** = Channel 4.)

The answer is:



STX (0x02)	cmd = 226 (VINT8)	cookie (VINT16LE)	
count = 4 (VINT16LE)		0 (not used)	
channel (VINT8)	TYPE (VINT8)	STATUS (VINT8)	0 (not used)
CRC16 (VINT16LE)		~ cmd = 29 (VINT8)	ETX (0x03)

where:

channel: is the same channel received from the request.

TYPE: indicates the type of status:

0 for nominal impedance

1 for load monitor

STATUS: the status (based on the type it could represent nominal impedance or load monitor) and can have the following values

0 if the nominal impedance or the load monitor (based on the type) is in the threshold and no short circuit is detected.

1 if the channel has a low short circuit (< 1 Ohm)

2 if the nominal impedance or the load monitor (based on the type) is below the specified threshold.

3 if the nominal impedance or the load monitor (based on the type) is above the specified threshold.

4 if the nominal impedance or the load monitor (based on the type) is unknown (i.e. if the channel is mute)

If request is not valid status will be zero.

Alarm Outs over Network

This will help you find the function to receive an alarm status over the network for the alarms you see set up in ControlSpace Designer that can be sent out the rear-panel GPO ALARM outputs.

Thermal Stress — over temp per channel from READALLALARMS2

Amplifier Standby — not a fault/alarm - can be read from STANDBY

Output Ch1-4: short circuit — READLOADSTATUS

Output Ch1-4: pilot tone voltage — detected value and state can be read from READALARMS (per channel)

Output Ch1-4: pilot tone load — READLOADSTATUS

Output Ch1-4: nominal impedance — READLOADSTATUS

Backup strategy Source 1-4 pilot tone lost: GPO has triggered out of source x (1-4), across any outputs that are assigned to play that source (1-4, or all) based on the matrix.

GPO/Relays can be read from READALARMS, READALLALARMS and READALLALARMS2 — whether backup strategy will trigger the relay depends on whether alarm option is set – but you can only derive if lost input caused it by ruling out all other causes.