



Protocol for X Series Amplifiers

Changelog:

Rev:	Note	Author
0.0	First edition	Michele Dionisio
0.1	Added EnergySafe	Michele Dionisio
0.2	Modified Gain resolution. (readgm, writeingain, writeingain, writemulti commands are changed too)	Michele Dionisio
0.3	Added read alarms	Michele Dionisio
0.4	Text Review	Luigi Chelli
0.5	Added command to read all gpio	Michele Dionisio
0.6	Added SaveAs	Michele Dionisio
0.7	Added GPIO commands	Michele Dionisio
0.8	Added get all alarms	Michele Dionisio
0.9	Add command to read more preset	Michele Dionisio
0.10	FIX documentation for STANDBY command	Michele Dionisio
0.11	Add loading snapshot without removing group	Michele Dionisio
0.12	Fix bitmap for READRESET	Michele Dionisio
0.13	Fix missing bitmap for INFO response	Michele Dionisio
0.14	Add SOURCEMETER, OUTPUTMETER command	Michele Dionisio
0.15	Change reference to paragraph in cmd=22,23,24 because paragraph number is not well generated	Michele Dionisio
0.16	Add READLOADSTATUS command	Filippo Digiugno

1 Introduction

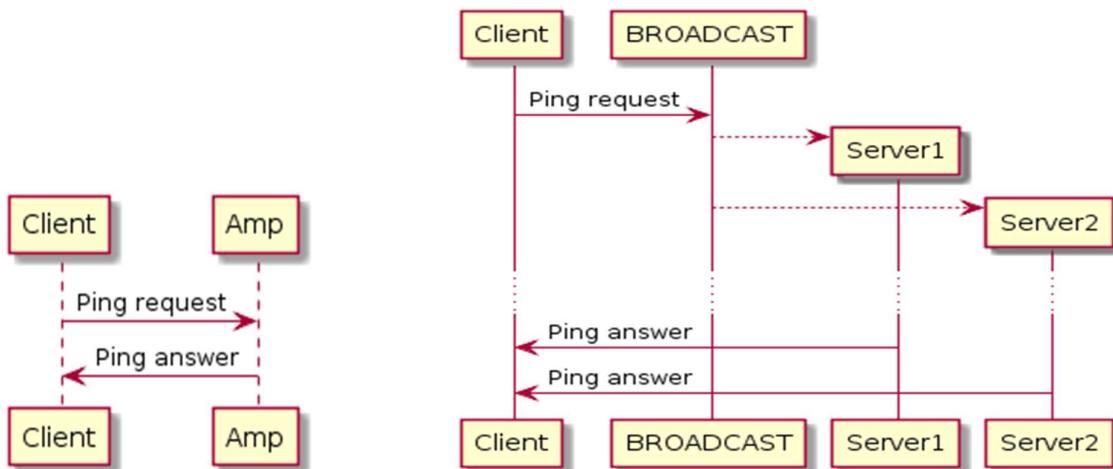
This document describes the protocol needed to interact with X Series amplifiers.

Usage of this protocol in conjunction with Armonia is discouraged, otherwise you might experience de-synchronizations between on-board settings and software settings.

2 Protocol

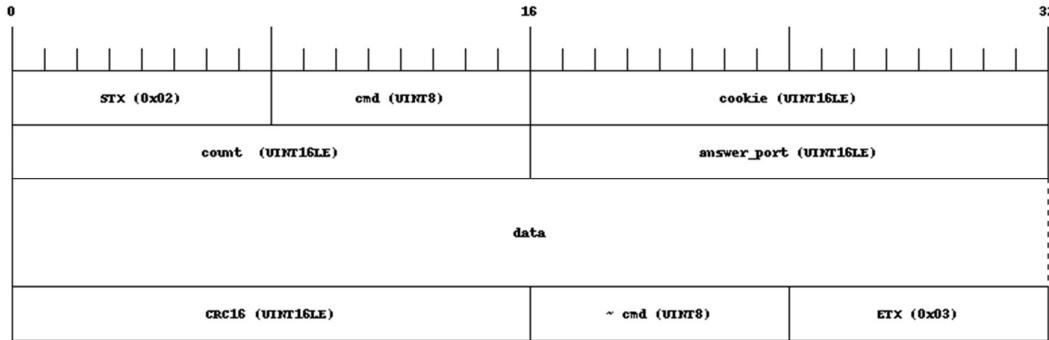
It's possible to interact with any amplifier using an UDP protocol. Each time that the amplifier receives a well-formatted message it replies with a well-formatted answer to the IP (single-cast) that originated the request.

The amplifier manages both broadcast and single-cast requests coming to its own IP address, port: 1234. It will answer to the port specified in the request, or to port 1234 if the requested port is zero.



The amplifier manages more than one request at the same time. Any incoming request is “marked” with a cookie that the 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 according to the following schema:



where:

STX: is one byte delimiter (0x02)
cmd: is one byte (0 - 127: for request, 128 – 255: for answer)
cookie: can be any 16bit 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: is the size in byte of the next data field. (it's unsigned 16 bit value in little endian format).
data: any data (empty is valid)
crc16¹: is the crc16 of the data field (0 for empty data).
~cmd: is the complement a1 of cmd
ETX: is one byte delimiter (0x03)

List of accepted cmd is:

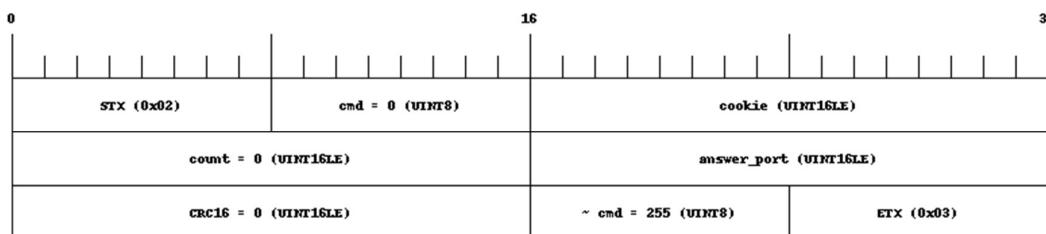
1. PING (cmd = 0, ~cmd = 255)
2. READGM (cmd = 1, ~cmd = 254)
3. WRITEINMUTE (cmd = 2, ~cmd = 253)
4. WRITEOUTMUTE (cmd = 3, ~cmd = 252)
5. WRITEINGAIN (cmd = 4, ~cmd = 251)
6. WRITEOUTGAIN (cmd = 5, ~cmd = 250)
7. READPRESET (cmd = 6, ~cmd = 249)
8. LOADPRESET (cmd = 7, ~cmd = 248)
9. WRITEMULTI (cmd = 8, ~cmd = 247)
10. REMOVEPRESET (cmd = 9, ~cmd = 246)
11. SAVEPRESET (cmd = 10, ~cmd = 245)
12. SAVEASPRESET (cmd = 16, ~cmd = 239)
13. INFO (cmd = 11, ~cmd = 244)
14. READPRESETINFO (cmd = 12, ~cmd = 243)
15. READALARMS (cmd = 13, ~cmd = 242)
16. STANDBY (cmd = 14, ~cmd = 241)
17. READALLALARMS (cmd = 15, ~cmd = 240) (deprecated, use READALLALARMS2)
18. READPILOTTONEGENERATOR (cmd = 17, ~cmd = 238)
19. READPILOTTONEDETECTION (cmd = 18, ~cmd = 237)
20. READLOADMONITOR (cmd = 19, ~cmd = 236)
21. READLOADDETECT (cmd = 20, ~cmd = 235)

¹It is the crc16 defined with the following polynomial: $x^{16} + x^{15} + x^2 + 1$. Example CRC16 of "123456789" is **0xBB3D**

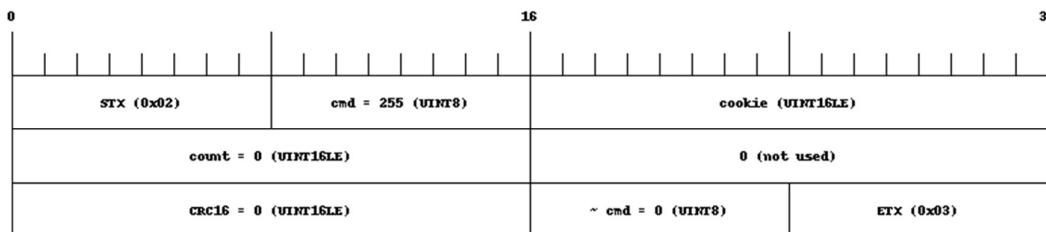
-
22. SETPILOTTONEGENERATOR (cmd =21, ~cmd = 234)
 23. SETPILOTTONEDETECTION (cmd =22, ~cmd = 233)
 24. SETLOADMONITOR (cmd =23, ~cmd = 232)
 25. SETLOADDETECT (cmd =24, ~cmd = 231)
 26. READALLALARMS2 (cmd =25, ~cmd = 230)
 27. LOADPRESET2 (cmd = 26, ~cmd = 229)
 28. SOURCEMETER (cmd = 27, ~cmd = 228)
 29. OUTPUTMETER (cmd = 28, ~cmd = 227)
 30. READLOADSTATUS (cmd=29, ~cmd = 226)

1 Ping

This command is used only to test if the amplifier is alive. The command is formatted according to the next schema:

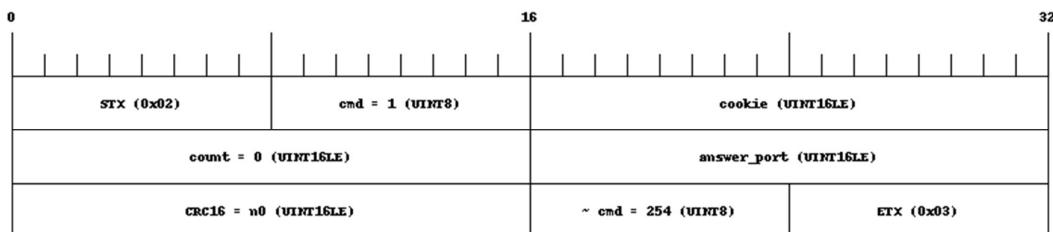


The answer is:



2 READGM

It is used to read all the Gains and Mutes status inside the amplifier. The command is formatted according to the next schema:



The answer is:

0		16	32
STX (0x02)		cmd = 254 (UINT8)	cookie (UINT16LE)
count = 52 (UINT16LE)		0 (not used)	
answer_ok (UINT8)	num_channels (UINT8)	0 (UINT8)	0 (UINT8)
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 (UINT8)	INMUTE1 (UINT8)	INMUTE2 (UINT8)	INMUTE3 (UINT8)
INMUTE4 (UINT8)	INMUTE5 (UINT8)	INMUTE6 (UINT8)	INMUTE7 (UINT8)
OUTMUTE0 (UINT8)	OUTMUTE1 (UINT8)	OUTMUTE2 (UINT8)	OUTMUTE3 (UINT8)
OUTMUTE4 (UINT8)	OUTMUTE5 (UINT8)	OUTMUTE6 (UINT8)	OUTMUTE7 (UINT8)
CRC16 (UINT16LE)		~ cmd = 1 (UINT8)	ETX (0x03)

Where:

- answer_ok:** 1 means 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 consider valid.
- INGAINX:** is the input gain of speaker X
- OUTGAINX:** is the output gain of channel X in cents of db (from -6000 to 15000 → -60db to +15db)
- INMUTEX:** is the mute status of speaker X
- OUTMUTEX:** is the mute status of output X in cents of db (from -6000 to 15000 → -60db to +15db)

3 WRITEINMUTE

It is used to set a mute status for one speaker (Adv EQ section in Armonia). The command is formatted according to the next schema:

0		16		32
	STX (0x02)	cmd = 2 (UINT8)	cookie (UINT16LE)	
	count = 4 (UINT16LE)		answer_port (UINT16LE)	
	CHANNEL (UINT8)	INMUTE (UINT8)	0 (UINT8)	0 (UINT8)
	CRC16 (UINT16LE)		~ cmd = 253 (UINT8)	ETX (0x03)

where:

- CHANNEL:** ranges from 0 to the number of speakers supported, and it's the speaker to be muted
- INMUTE:** 0 to unmute, 1 to mute

The answer is:

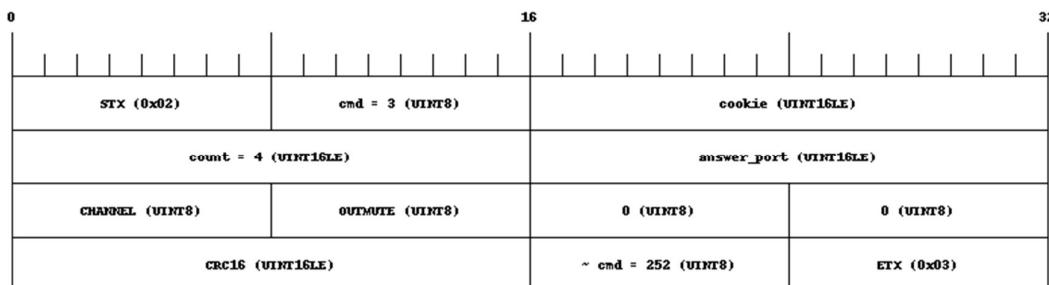
0		16		32
	STX (0x02)	cmd = 253 (UINT8)	cookie (UINT16LE)	
	count = 4 (UINT16LE)		0 (not used)	
	answer_ok (UINT8)	CHANNEL (UINT8)	INMUTE (UINT8)	0 (UINT8)
	CRC16 (UINT16LE)		~ cmd = 2 (UINT8)	ETX (0x03)

where:

- answer_ok:** 1 means 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

4 WRITEOUTMUTE

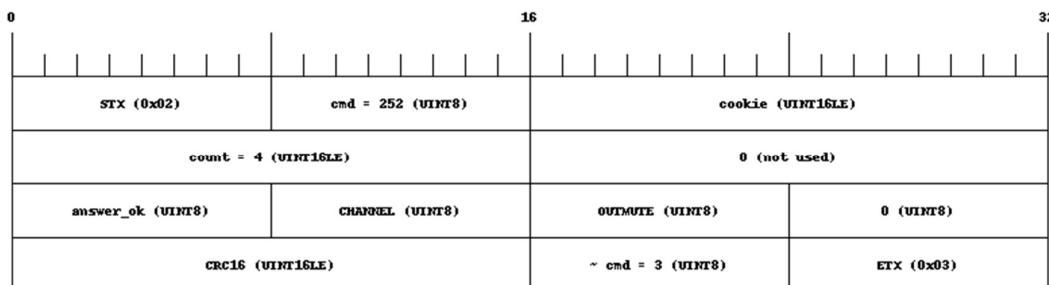
It's used to set a mute status for one output channel ("Way EQ" in Armonia). The command is formatted according to the next schema:



where:

CHANNEL: is from 0 to number of channel supported, and is the channel to mute
OUTMUTE: is 0 to unmute 1 to mute

The answer is:



where:

answer_ok: 1 means valid answer.
CHANNEL: has to be the same of the CHANNEL field inside the request
OUTMUTE: has to be the same of the OUTMUTE field inside the request

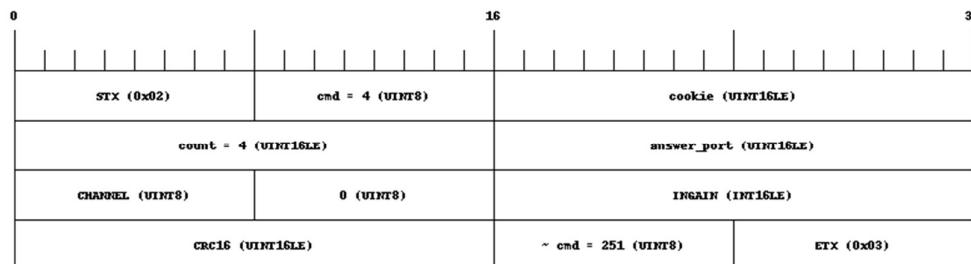
5 WRITEINGAIN

It's used to set a gain status for one speaker (Adv EQ section in Armonia). The command is formatted according to the next schema:

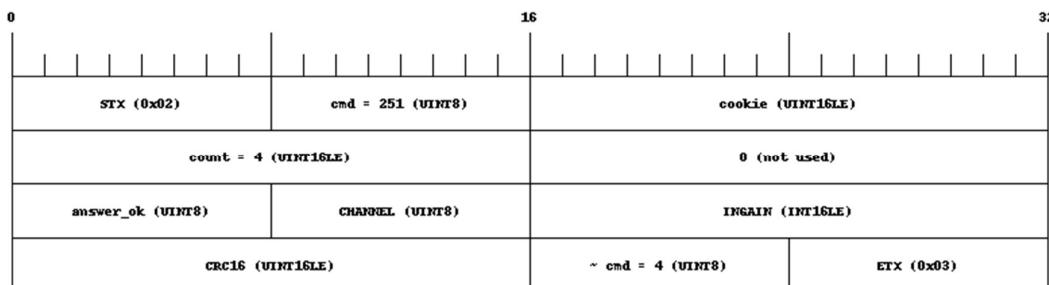
where:

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

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



The answer is:



where:

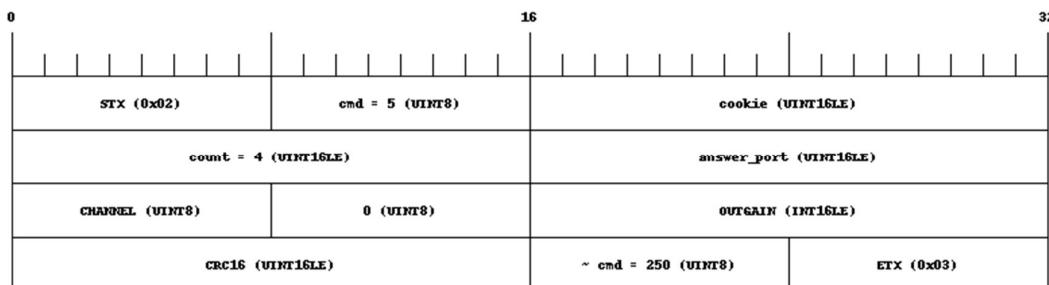
answer_ok: 1 means valid answer.

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

6 WRITEOUTGAIN

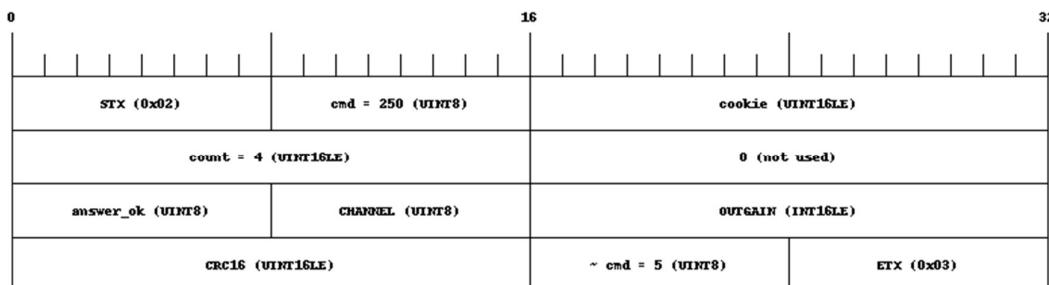
It is used to set a gain status for one output channel ("Way EQ" in Armonia). The command is formatted according to the next schema:



where:

CHANNEL: is from 0 to number of channel supported, and is the channel to control
OUTGAIN: is number in cents of db (from -6000 to 15000 → -60db to +15db)

The answer is:



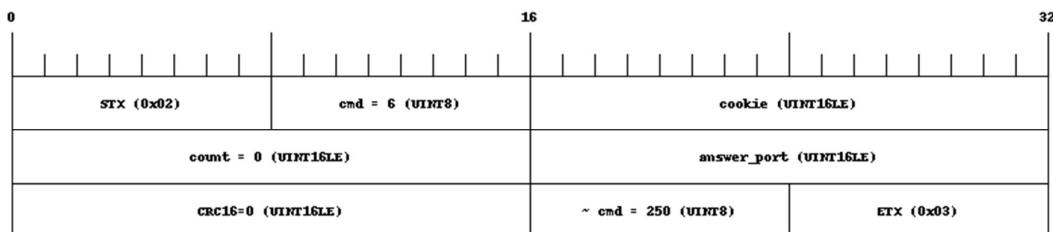
where:

answer_ok: 1 means 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

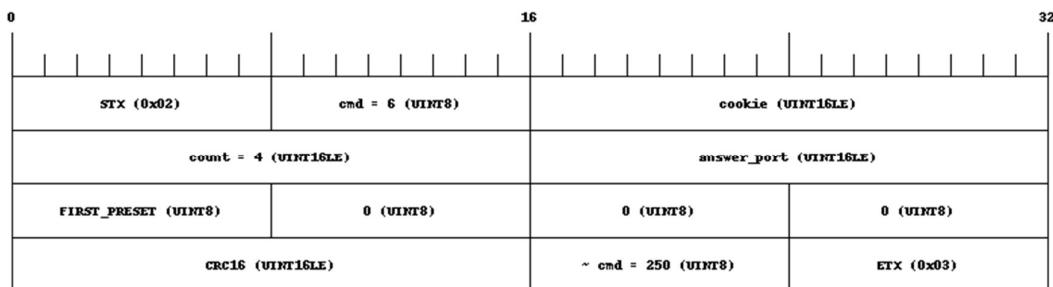


7 READPRESET

It is used to request the list of available presets ("Snapshots" of the entire amplifier). The command is formatted according to the next 2 possible schemas:



or



where:

FIRST_PRESET: is the index of the first consecutive preset requested. The first schema is equivalent to the second with PAGE=0

If FIRST_PRESET=0 (or the request is done with schema 1) the answer is:

0	16	32
STX (0x02)	cmd = 250 (UINT8)	cookie (UINT16LE)
count = 8 (UINT16LE)		not_used (0)
answer_ok	PRESET (01-02-03-04-05-06-07-08) PRESET (09-10-11-12-13-14-15-16) PRESET (17-18-19-20-21-22-23-24)	
PRESET (25-26-27-28-29-30-31-32) PRESET (33-34-35-36-37-38-39-40) PRESET (41-42-43-44-45-46-47-48) PRESET (49-50-51-52-53-54-55-56)		
CRC16 (UINT16LE)	~ cmd = 6 (UINT8)	ETX (0x03)

otherwise

0	16	32
STX (0x02)	cmd = 250 (UINT8)	cookie (UINT16LE)
count = 8 (UINT16LE)		not_used (0)
answer_ok (UINT8)	FIRST_PRESET (UINT8)	not_used (0) not_used (0)
not_used (0)	PRESET (01-02-03-04-05-06-07-08) PRESET (09-10-11-12-13-14-15-16) PRESET (17-18-19-20-21-22-23-24)	
PRESET (25-26-27-28-29-30-31-32) PRESET (33-34-35-36-37-38-39-40) PRESET (41-42-43-44-45-46-47-48) PRESET (49-50-51-52-53-54-55-56)		
CRC16 (UINT16LE)	~ cmd = 6 (UINT8)	ETX (0x03)

where:

FIRST_PRESET: it the number that has to be added to PRESET.

answer_ok: 1 means valid answer.

PRESET (01-02-03-04-05-06-07-08): is a bit field array, i.e. bit 0 set to 1 means that the preset 1 is available, bit 7 set to 1 means that preset 7 is available.

PRESET (09-10-11-12-13-14-15-16): is a bit field array with the same meaning of the previous field but related to preset 09 to 16

....

8 LOADPRESET

It's used to load one preset (a "Snapshot" of the entire amplifier). The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 7 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
PRESET (UINT8)	0 (UINT8)	0 (UINT8) 0 (UINT8)
CRC16 (UINT16LE)	~ cmd = 248 (UINT8)	ETX (0x03)

Where:

PRESET: is a number from 0 to 200 representing the preset to be loaded

The answer is:

0	16	32
STX (0x02)	cmd = 248 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		0 (not used)
answer_ok (UINT8)	PRESET (UINT8)	0 (UINT8) 0 (UINT8)
CRC16 (UINT16LE)	~ cmd = 7 (UINT8)	ETX (0x03)

where:

answer_ok: 1 means valid answer

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

9 WRITEMULTI

It is used to set same mute/gain value to more than one channels. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 8 (UINT8)	cookie (UINT16LE)
count = 12 (UINT16LE)		answer_port (UINT16LE)
CHANNEL-MASK (UINT8)	0 (UINT8)	INGAIN (INT16LE)
CHANNEL-MASK (UINT8)	0 (UINT8)	OUTGAIN (INT16LE)
CHANNEL-MASK (UINT8)	INMUTE (UINT8)	CHANNEL-MASK (UINT8)
CRC16 (UINT16LE)	~ cmd = 247 (UINT8)	OUTMUTE (UINT8)
		ETX (0x03)

Where:

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

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

The answer is:

0	16	32
STX (0x02)	cmd = 247 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		0 (not used)
answer_ok (UINT8)	0 (UINT8)	0 (UINT8)
CRC16 (UINT16LE)	~ cmd = 8 (UINT8)	ETX (0x03)

10 REMOVEPRESET

It is used to remove a preset from the list of the available ones. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 9 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
PRESET (UINT8)	0 (UINT8)	0 (UINT8) 0 (UINT8)
CRC16 (UINT16LE)	~ cmd = 246 (UINT8)	ETX (0x03)

Where:

PRESET: is a number from 0 to 200 representing the preset to be removed

The answer is:

0	16	32
STX (0x02)	cmd = 246 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		0 (not used)
answer_ok (UINT8)	PRESET (UINT8)	0 (UINT8) 0 (UINT8)
CRC16 (UINT16LE)	~ cmd = 9 (UINT8)	ETX (0x03)

where:

answer_ok: 1 means valid answer

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

11 SAVEPRESET

It is used to save the running situation as a new available preset. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 10 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
PRESET (UINT8)	0 (UINT8)	0 (UINT8) 0 (UINT8)
CRC16 (UINT16LE)	~ cmd = 245 (UINT8)	ETX (0x03)

Where:

PRESET: is a number from 0 to 200 representing the preset to be saved. If another preset occupies the destination slot it should be removed first.

The answer is:

0	16	32
STX (0x02)	cmd = 245 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		0 (not used)
answer_ok (UINT8)	PRESET (UINT8)	0 (UINT8) 0 (UINT8)
CRC16 (UINT16LE)	~ cmd = 10 (UINT8)	ETX (0x03)

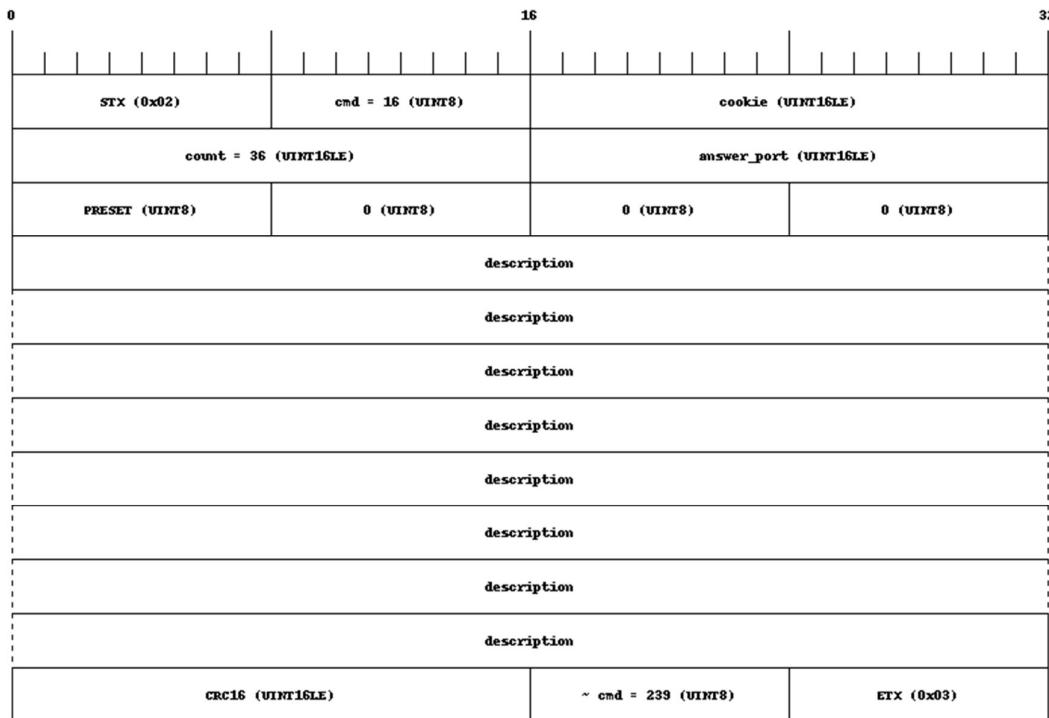
Where:

answer_ok: 1 means a valid answer

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

12 SAVEASPRESET

It is used to save the running situation as a new available preset. The command is formatted according to the next schema:

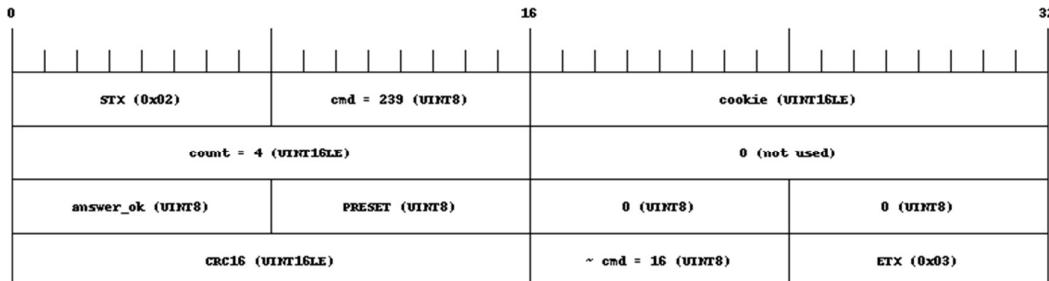


Where:

PRESET: is a number from 0 to 200 representing the preset to be saved. If another preset occupies the destination slot it should be removed first.

description: is a string filed null terminated (max 31 charapters + '\0')

The answer is:



Where:

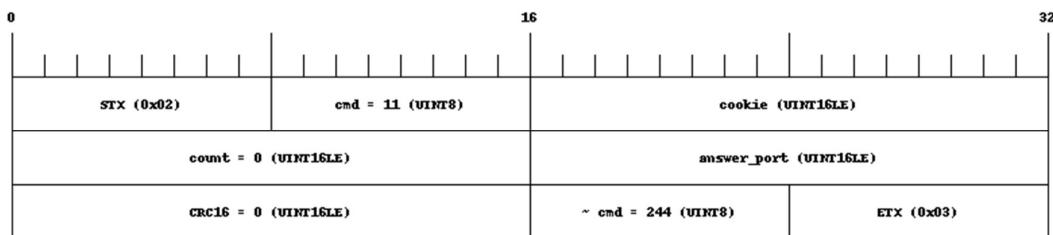


answer_ok: 1 means a valid answer

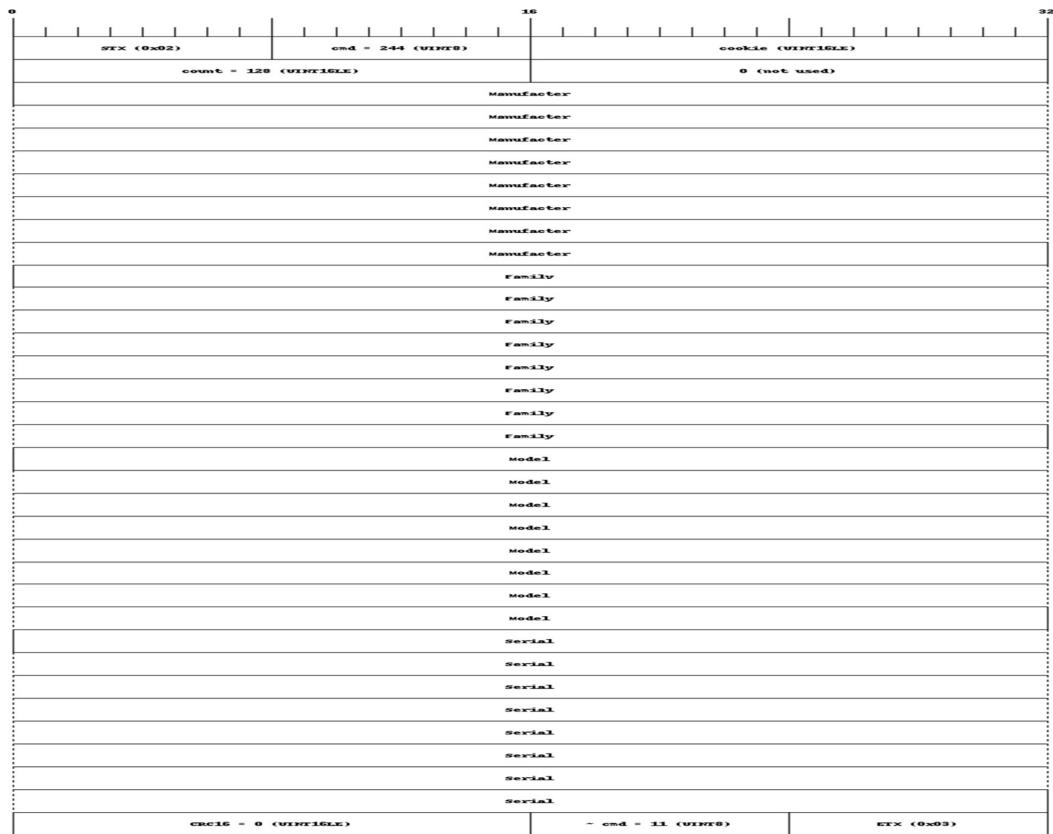
PRESET: has to be the same of the PRESET field inside the request

13 INFO

It is used to read static information from an amplifier. The command is formatted according to the next schema:



The answer has 128 byte of data with the following meaning:





Manufacturer: is a string null terminated (max 31 chapters + '\0')

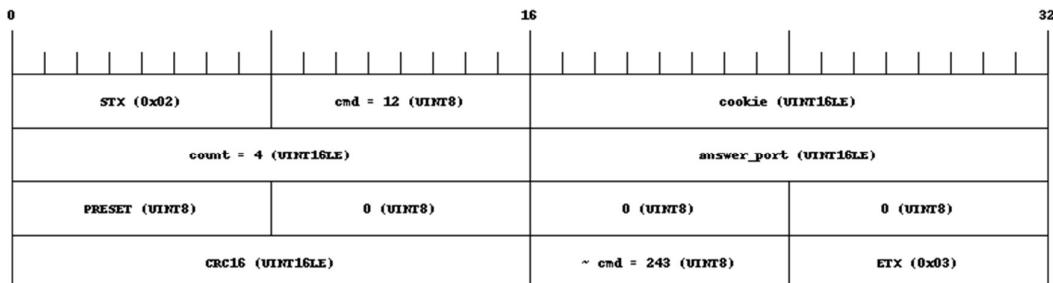
Family: is a string null terminated (max 31 chapters + '\0')

Model: is a string null terminated (max 31 chapters + '\0')

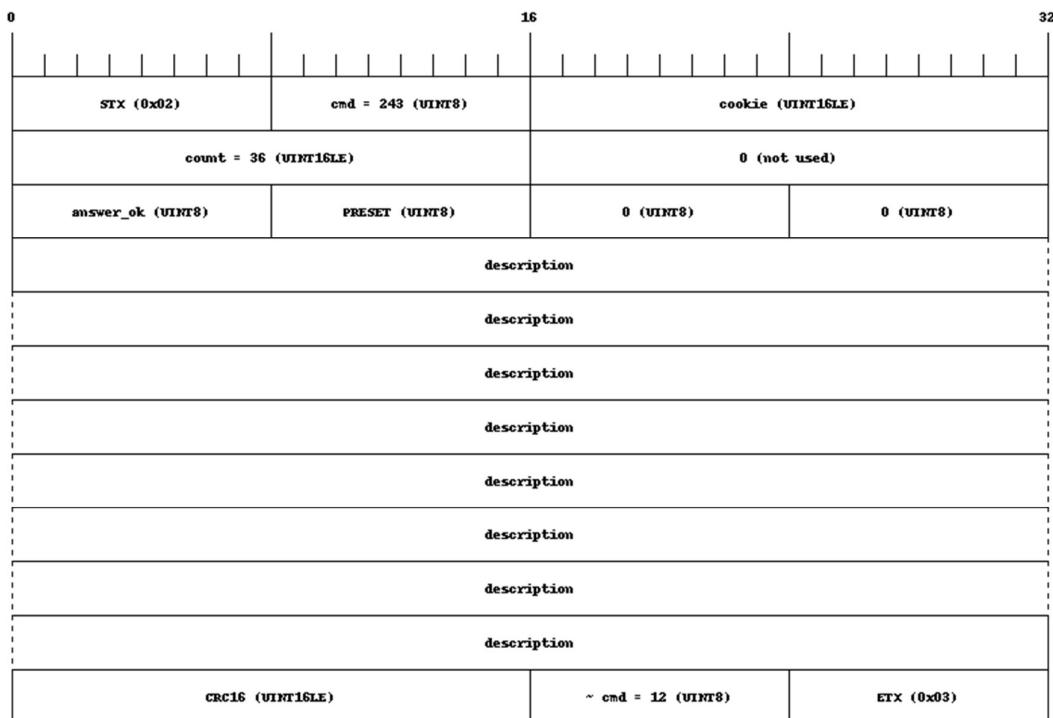
Serial: is a string null terminated (max 31 chapters + '\0')

14 READPRESETINFO

It is used to read preset description. The command is formatted according to the next schema:



The answer is:



where:

description: is a string filed null terminated (max 31 chapters + '\0')

15 READALARMS

It is used to read alarms and metering live status. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 13 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
channel (UINT8)		
CRC16 = 0 (UINT16LE)	~ cmd = 242 (UINT8)	ETX (0x03)

where:

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

The answer is:

0	16	32
STX (0x02)	cmd = 242 (UINT8)	cookie (UINT16LE)
count = 20 (UINT16LE)		0 (not used)
answer_ok (UINT8)	channel (UINT8)	0
PT_RMS (UINT16LE)	PT_DETECTED (UINT8)	PT_VALIDITY (UINT8)
PT_NI_RMS (UINT16LE)	PT_NI_DETECTED (UINT8)	PT_NI_VALIDITY (UINT8)
NI_RMS (UINT16LE)	NI_DETECTED (UINT8)	NI_VALIDITY (UINT8)
DIP-SWITCH (UINT8)	ALARMS (UINT8)	SELECTED_IN (UINT8)
CRC16 = 0 (UINT16LE)	~ cmd = 13 (UINT8)	ETX (0x03)



where:

answer_ok: 1 means valid answer

channel: is the same channel received from request

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

x_DETECTED: 1 means detected

PT_RMS: is the detected input pilot tone level in s of volts (Source Selection window in Armonia)

PT_NI_RMS: is the measured impedance at a specific frequency (set via Armonia) of the connected load, in tenths of Ohm

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

DIP-SWITCH (for Ottocanali DSP+D only): 0 for LOW_Z, 1 for HIGH_Z_70, 2 for HIGH_Z_100, 3 for out of range

ALARMS: status of output relays (for Ottocanali DSP+D)

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

16 STANDBY

It is used to set/read the standby status. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 14 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
ON-OFF-READ (UINT8)	0 (UINT8)	0 (UINT8) 0 (UINT8)
CRC16 (UINT16LE)	~ cmd = 241 (UINT8)	ETX (0x03)

Where:

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:

0	16	32
STX (0x02)	cmd = 241 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
answer_ok (UINT8)	ON-OFF (UINT8)	0 (UINT8) 0 (UINT8)
CRC16 (UINT16LE)	~ cmd = 14 (UINT8)	ETX (0x03)

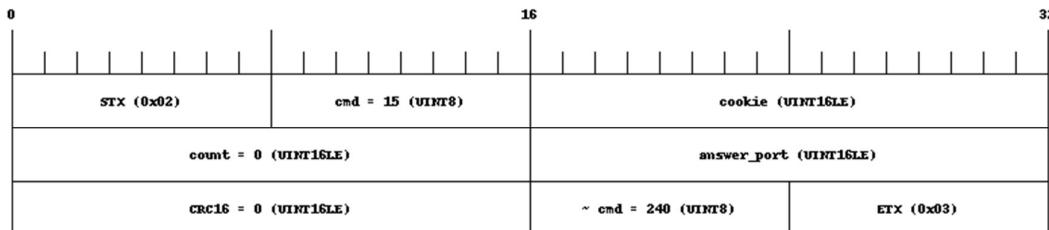
Where:

answer_ok: 1 means a valid answer

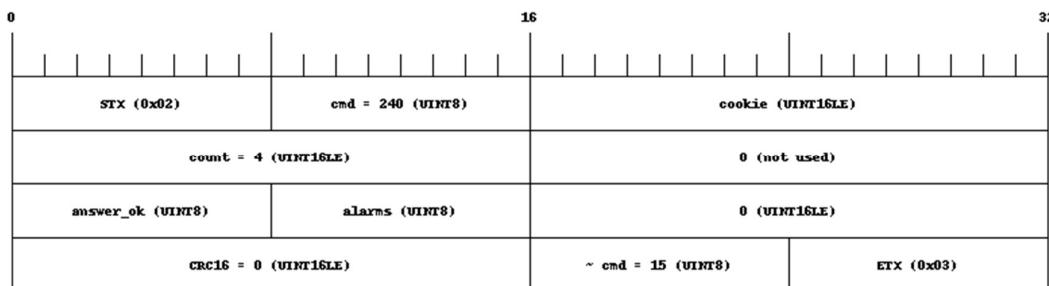
ON-OFF: 2 means STANDBY OFF (amplifier operative), 1 means ON (amplifier not operative)

17 READALLALARMS

It is used to read alarms and metering live status (it is deprecated because it is possible to use READALLALARMS2).
The command is formatted according to the next schema:



The answer is:



where:

answer_ok: 1 means valid answer

ALARMS: status of output relays (*for Ottocanali DSP+D*)

18 READPILOTTONEGENERATOR

It is used to read Inner Pilot Tone Generator setting. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 17 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
channel (UINT8)		
CRC16 = 0 (UINT16LE)	~ cmd = 238 (UINT8)	ETX (0x03)

where:

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

The answer is:

0	16	32
STX (0x02)	cmd = 238 (UINT8)	cookie (UINT16LE)
count = 8 (UINT16LE)		0 (not used)
answer_ok (UINT8)	0 (not used)	channel (UINT8) ON_OFF (UINT8)
PT_FREQ (UINT16LE)	PT_AMP (UINT16LE)	
CRC16 = 0 (UINT16LE)	~ cmd = 17 (UINT8)	ETX (0x03)

where:

answer_ok: 1 means valid answer

channel: is the same channel received from request

ON_OFF: 0 means 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

19 READPILOTTONEDETECTION

It is used to read Output Pilot Tone Detection setting. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 18 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
channel (UINT8)		
CRC16 = 0 (UINT16LE)	~ cmd = 237 (UINT8)	ETX (0x03)

where:

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

The answer is:

0	16	32
STX (0x02)	cmd = 237 (UINT8)	cookie (UINT16LE)
count = 12 (UINT16LE)		0 (not used)
answer_ok (UINT8)	0 (not used)	channel (UINT8) ON_OFF (UINT8)
PT_FREQ (UINT16LE)		0 (not used)
PT_THL (UINT16LE)		PT_THH (UINT16LE)
CRC16 = 0 (UINT16LE)	~ cmd = 18 (UINT8)	ETX (0x03)

where:

answer_ok: 1 means valid answer

channel: is the same channel received from request

ON_OFF: 0 means OFF.

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

PT_THL: is low threshold (Vrms) in tenths of volts

PT_THH: is high threshold (Vrms) in tenths of volts

20 READLOADMONITOR

It is used to read Output Load Monitor setting. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 19 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
channel (UINT8)		
CRC16 = 0 (UINT16LE)	~ cmd = 236 (UINT8)	ETX (0x03)

where:

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

The answer is:

0	16	32
STX (0x02)	cmd = 236 (UINT8)	cookie (UINT16LE)
count = 12 (UINT16LE)		0 (not used)
answer_ok (UINT8)	0 (not used)	channel (UINT8) ON_OFF (UINT8)
LM_FREQ (UINT16LE)		0 (not used)
LM_THL (UINT16LE)		LM_THH (UINT16LE)
CRC16 = 0 (UINT16LE)	~ cmd = 19 (UINT8)	ETX (0x03)

where:

answer_ok: 1 means valid answer

channel: is the same channel received from request

ON_OFF: 0 means 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

21 READLOADDETECT

It is used to read Inner Pilot Tone Generator setting. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 20 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
channel (UINT8)		
CRC16 = 0 (UINT16LE)	~ cmd = 235 (UINT8)	ETX (0x03)

where:

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

The answer is:

0	16	32
STX (0x02)	cmd = 235 (UINT8)	cookie (UINT16LE)
count = 8 (UINT16LE)		0 (not used)
answer_ok (UINT8)	0 (not used)	channel (UINT8) ON_OFF (UINT8)
LD_THL (UINT16LE)		LD_THH (UINT16LE)
CRC16 = 0 (UINT16LE)	~ cmd = 20 (UINT8)	ETX (0x03)

where:

answer_ok: 1 means valid answer

channel: is the same channel received from request

ON_OFF: 0 means OFF.

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

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

22 SETPILOTTONEGENERATOR

It is used to set (on or off) Pilot Tone Generation. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 21 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
channel (UINT8)	ON_OFF (UINT8)	
CRC16 = 0 (UINT16LE)	~ cmd = 234 (UINT8)	ETX (0x03)

where:

CHANNEL: is from 0 to number of channel supported, and is the channel to read
ON_OFF: 0 means OFF.

The answer is:

0	16	32
STX (0x02)	cmd = 234 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		0 (not used)
answer_ok (UINT8)	0 (not used)	channel (UINT8)
		ON_OFF (UINT8)
CRC16 = 0 (UINT16LE)	~ cmd = 21 (UINT8)	ETX (0x03)

where:

answer_ok: 1 means valid answer
channel: is the same channel received from request
ON_OFF: is the same channel received from request



23 SETPILOTTONEDETECTION

See paragraph SETPILOTTONEGENERATOR but with cmd=22



24 SETLOADMONITOR

See paragraph SETPILOTTONEGENERATOR but with cmd=23

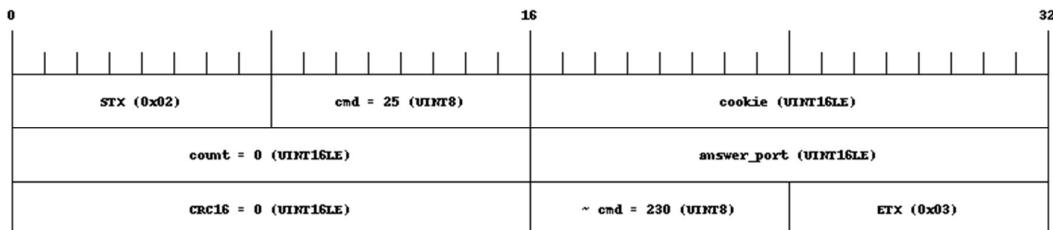


25 SETLOADDETECT

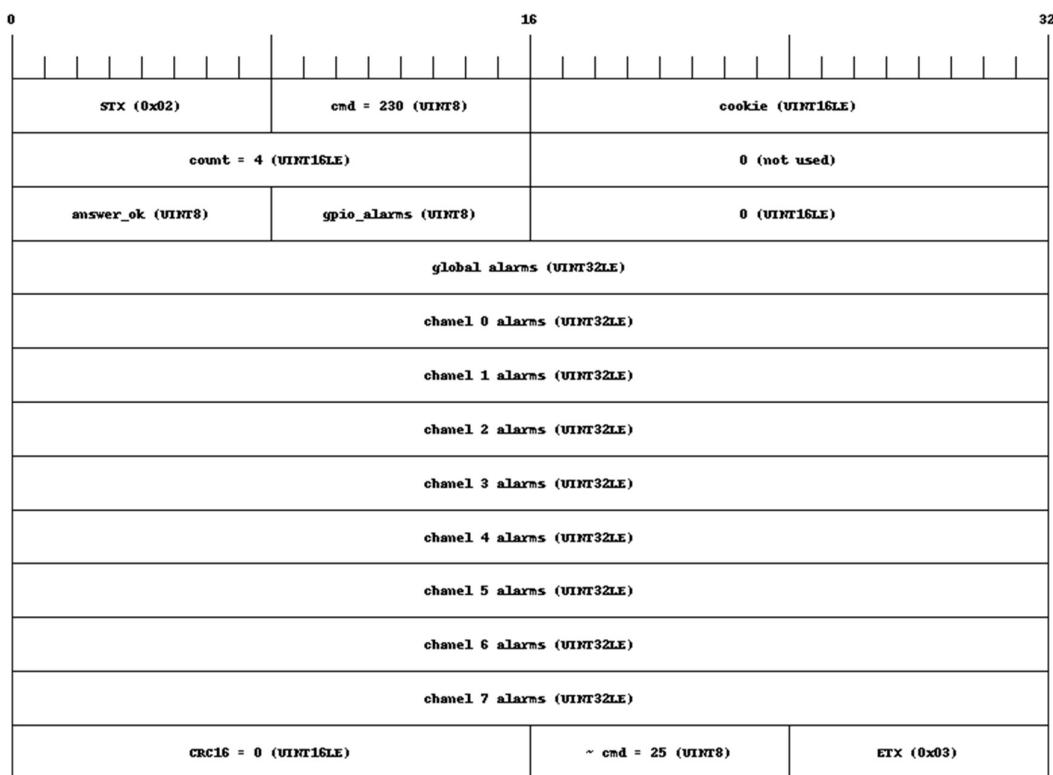
See paragraph SETPILOTTONEGENERATOR but with cmd=24

26 READALLALARMS2

It is used to read alarms status. The command is formatted according to the next schema:



The answer is:



where:

answer_ok: 1 means valid answer



gpio_alarms: status of output relays (*for Ottocanali DSP+D*) where bit 0 (LSB) is for channel 0 and bit 7 (MSB) for channel 7

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

27 READPILOTTONEGENERATOR

It is used to read Inner Pilot Tone Generator setting. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 17 (UINT8)	cookie (UINT16LE)
count = 4 (UINT16LE)		answer_port (UINT16LE)
channel (UINT8)		
CRC16 = 0 (UINT16LE)	~ cmd = 238 (UINT8)	ETX (0x03)

where:

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

The answer is:

0	16	32
STX (0x02)	cmd = 238 (UINT8)	cookie (UINT16LE)
count = 8 (UINT16LE)		0 (not used)
answer_ok (UINT8)	0 (not used)	channel (UINT8) ON_OFF (UINT8)
PT_FREQ (UINT16LE)	PT_AMP (UINT16LE)	
CRC16 = 0 (UINT16LE)	~ cmd = 17 (UINT8)	ETX (0x03)

where:

answer_ok: 1 means valid answer

channel: is the same channel received from request

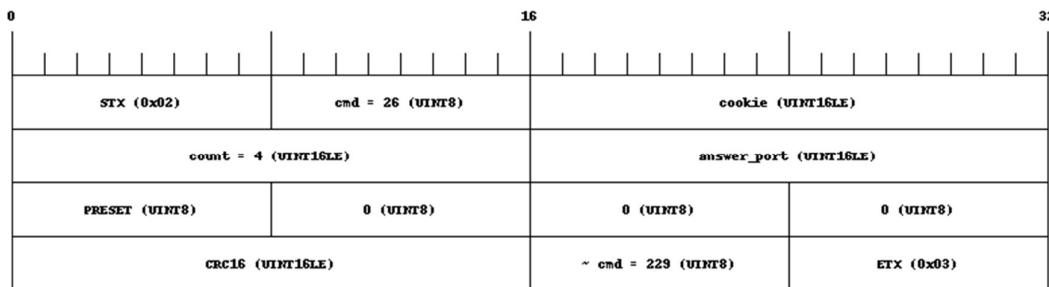
ON_OFF: 0 means OFF.

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

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

28 LOADPRESET2

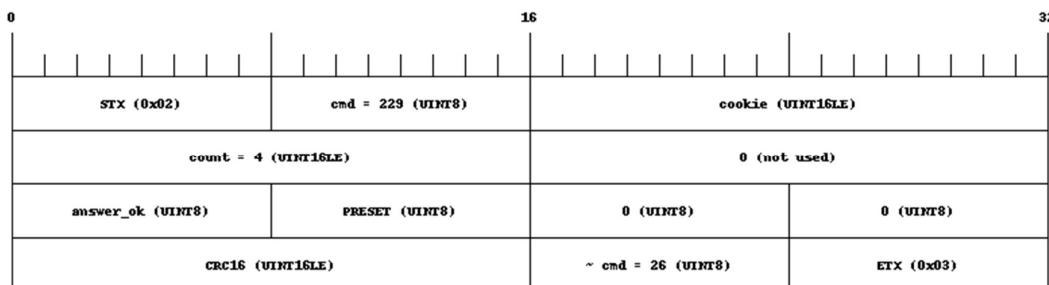
It's used to load one preset (a "Snapshot" of the entire amplifier). This command is identical to LOADPRESET but it avoid to delete group if there is no change in speaker connections. The command is formatted according to the next schema:



Where:

PRESET: is a number from 0 to 200 representing the preset to be loaded

The answer is:



where:

answer_ok: 1 means valid answer

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

29 SORCEMETER

It's used to read source meter information. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 27 (UINT8)	cookie (UINT16LE)
count = 0 (UINT16LE)		answer_port (UINT16LE)
CRC16=0 (UINT16LE)	~ cmd = 228 (UINT8)	ETX (0x03)

The answer is:

0	16	32
STX (0x02)	cmd = 228 (UINT8)	cookie (UINT16LE)
count = 8 (UINT16LE)		not used (UINT16LE)
PRESENCE_CLIP_0 (UINT8LE)	PRESENCE_CLIP_1 (UINT8LE)	PRESENCE_CLIP_2 (UINT8LE)
PRESENCE_CLIP_4 (UINT8LE)	PRESENCE_CLIP_5 (UINT8LE)	PRESENCE_CLIP_6 (UINT8LE)
		PRESENCE_CLIP_3 (UINT8LE)
		PRESENCE_CLIP_7 (UINT8LE)
CRC16 (UINT16LE)	~ cmd = 27 (UINT8)	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

30 OUTPUTMETER

It's used to read output meter information. The command is formatted according to the next schema:

0	16	32
STX (0x02)	cmd = 28 (UINT8)	cookie (UINT16LE)
count = 0 (UINT16LE)		answer_port (UINT16LE)
CRC16=0 (UINT16LE)	~ cmd = 227 (UINT8)	ETX (0x03)

The answer is:

0	16	32
STX (0x02)	cmd = 227 (UINT8)	cookie (UINT16LE)
count = 36 (UINT16LE)		not used (UINT16LE)
OUT_RMV_V_0 (INT16LE)		OUT_HEADROOM_0 (INT16LE)
OUT_RMV_V_1 (INT16LE)		OUT_HEADROOM_1 (INT16LE)
OUT_RMV_V_2 (INT16LE)		OUT_HEADROOM_2 (INT16LE)
OUT_RMV_V_3 (INT16LE)		OUT_HEADROOM_3 (INT16LE)
OUT_RMV_V_4 (INT16LE)		OUT_HEADROOM_4 (INT16LE)
OUT_RMV_V_5 (INT16LE)		OUT_HEADROOM_5 (INT16LE)
OUT_RMV_V_6 (INT16LE)		OUT_HEADROOM_6 (INT16LE)
OUT_RMV_V_7 (INT16LE)		OUT_HEADROOM_7 (INT16LE)
OUT_SIGNAL_PRESENCE (UINT8LE)	not used 1 (UINT8LE)	not used 1 (UINT16LE)
CRC16 (UINT16LE)	~ cmd = 28 (UINT8)	ETX (0x03)

where:

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

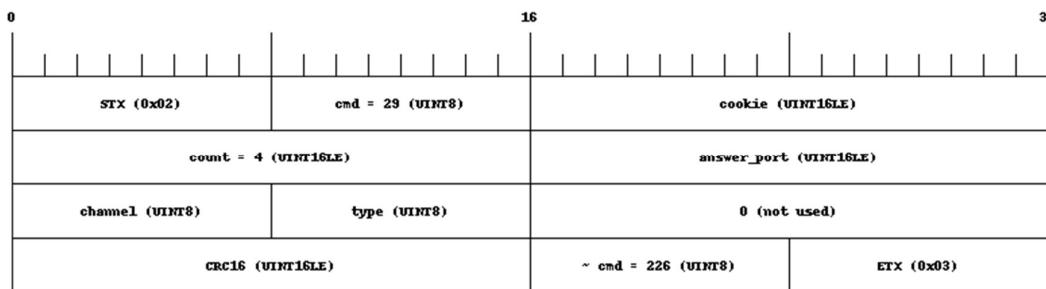
OUT_HEADROOM_<x>: is the output headroom of channel X in cents of db.

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

31 READLOADSTATUS

It is used to read the status of the load monitor for a single channel. The command is formatted according to the next schema:



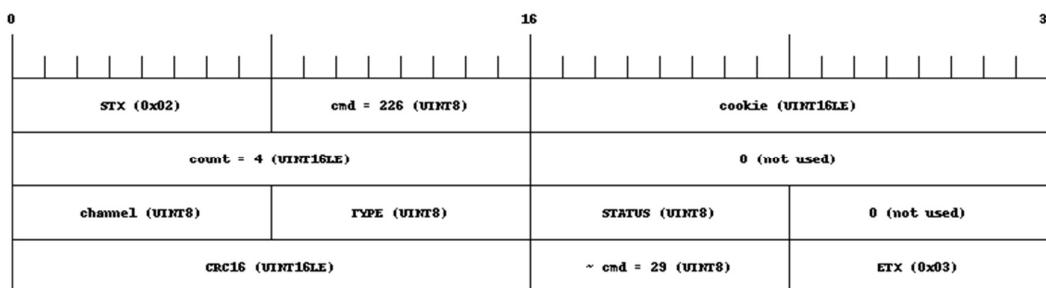
where:

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

TYPE: means the type of information to get:

- 0 for the nominal impedance
- 1 for the load monitor

The answer is:



where:

channel: is the same channel received from request

TYPE: means the type of the 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.