P300

Command Strings

P300 command strings for third-party control systems, such as Crestron or Extron.
Version: 3.4 (2021-E)

Table of Contents

P300
Command Strings

P300 Command Strings

The device is connected via Ethernet to a control system, such as AMX, Crestron or Extron.

Connection: Ethernet (TCP/IP; select "Client" in the AMX/Crestron program)
Port: 2202

If using static IP addresses, the Shure Control and the Audio Network settings must be set to manual in Designer. Use the Control IP address for TCP/IP communication with Shure devices.

Conventions

The device has 4 types of strings:

GET

> Finds the status of a parameter. After the AMX/Crestron sends a GET command, the P300 responds with a REPORT string

SET

> Changes the status of a parameter. After the AMX/Crestron sends a SET command, the P300 will respond with a REPORT string to indicate the new value of the parameter.

REP

> When the P300 receives a GET or SET command, it will reply with a REPORT command to indicate the status of the parameter. REPORT is also sent by the P300 when a parameter is changed on the P300.

SAMPLE

> Used for metering audio levels.

All messages sent and received are ASCII. Note that the level indicators and gain indicators are also in ASCII

Most parameters will send a REPORT command when they change. Thus, it is not necessary to constantly query parameters. The P300 will send a REPORT command when any of these parameters change.

The character

"x"

in all of the following strings represents the channel of the P300 and can be ASCII numbers 0 through 4 as in the following table

00	All Channels
01-08	Dante Inputs with Mic Processing
09-10	Dante Inputs
11-12	Analog Inputs

13	USB Input
14	Mobile Input
15-16	Dante Outputs
17-18	Analog Outputs
19	USB Output
20	Mobile Output
21	Automixer Output
22	AEC Reference
23-28	Dante Outputs (firmware 4.1.x and newer)

Example Scenario: Muting a System

The Acoustic Echo Canceler (AEC) and P300 automixer require constant audio signal from the microphone to operate. Do NOT send commands to the microphone to mute locally. Instead, use logic communication between the P300 and Microflex Advance devices. This allows the AEC to continue processing audio even while the system is muted, and deliver the best results when the system is unmuted.

After logic functionality is set up between Shure devices, send the command from the control system to mute the P300 automixer output. If set up correctly, the P300 automixer output will mute, and the microphone LED color will change to indicate the system is muted.

Note: Although the MXA310 LED status shows the system is muted, the audio signal is still passed to the P300 to allow continuous processing.

Crestron/AMX Control System

Crestron/AMX sends the mute command to the P300.

P300

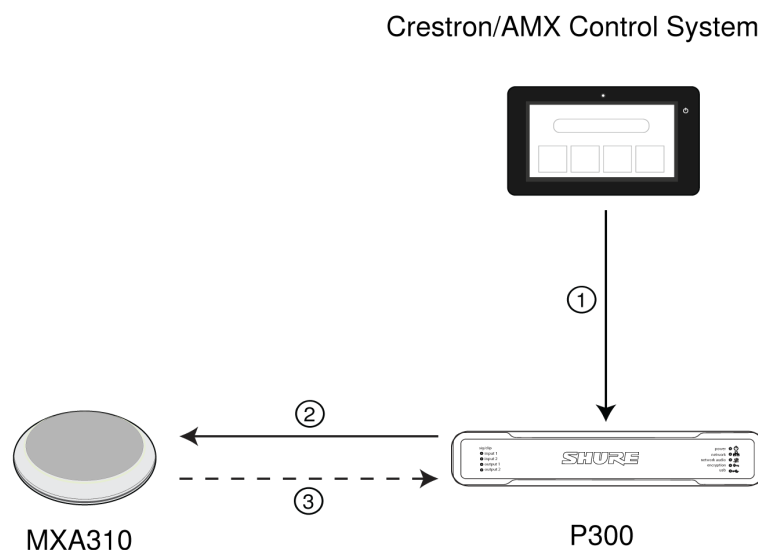The LED command to indicate mute state is sent from the P300 to the MXA310.

MXA310

The MXA310 sends audio to the P300 for continuous processing.

Required Steps for Logic Functionality

1. In the MXA310 web application, go to `Configuration > Button Control`, then set mode to Logic Out.
2. In Designer, open the P300 and go to the Input tab. Enable Logic for every channel routed from the MXA310 microphone. The device type appears at the bottom of the input channel strip.

Note: The MXA910 does not require set up for logic functionality.

Crestron/AMX Control System

MXA310   P300

① Mute Command

Crestron/AMX sends the mute command to the P300.

② LED Command

The P300 sends the LED command to the MXA310 so that the microphone LED color matches the system mute state.

③ Continuous Audio Signal

The MXA310 sends audio to the P300 for continuous processing. The system is muted from the P300 at the end of the audio chain.

Best Practices for Muting:

① Mute Button:

Press the mute button on the Crestron/AMX panel.

② Crestron/AMX sends following command to P300:

< SET 21 AUTOMXR_MUTE TOGGLE >

Note: The TOGGLE command simplifies logic within the Crestron/AMX. ON/OFF commands can be used instead, but supplemental processes must be implemented within the Crestron/AMX.

③ P300 Automixer channels mute, and P300 sends following REPORT back to Crestron/AMX:

< REP 21 AUTOMXR_MUTE ON >

This REPORT command can be used in various ways for button feedback on the control surface.

Command Strings (Common)

Get All

Command String: `< GET xx ALL >`	Where xx is ASCII channel number: 00 through 21, 23-28. Use this command on first power on to update the status of all parameters.
P300 Response: `< REP ... >`	The P300 responds with individual Report strings for all parameters.

Get Model Number

Command String: `< GET MODEL >`	
P300 Response: `< REP MODEL {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 32 characters of the model number. The P300 always responds with a 32 character model number.

Get Serial Number

Command String: `< GET SERIAL_NUM >`	
P300 Response: `< REP SERIAL_NUM {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 32 characters of the serial number. The P300 always responds with a 32 character serial number.

Get Channel Name

Command String: `< GET xx CHAN_NAME >`	Where xx is ASCII channel number: 00 through 20, 23-28.
P300 Response: `< REP xx CHAN_NAME {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 31 characters of the user name. The P300 always responds with a 31 character name.

Get Device ID

Command String: `< GET DEVICE_ID >`	The Device ID command does not contain the x channel character, as it is for the entire P300.
P300 Response: `< REP DEVICE_ID {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy is 31 characters of the device ID. The

	P300 always responds with a 31 character device ID.
Get Firmware Version	
Command String: `< GET FW_VER >`	
P300 Response: `< REP FW_VER {yyyyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyyyy *is 18 characters. The P300 always responds with 18 characters.*
Get Preset	
Command String: `< GET PRESET >`	
P300 Response: `< REP PRESET nn >`	Where nn is the preset number 01-10.
Set Preset	
Command String: `< SET PRESET nn >`	Where nn is the preset number 1-10. (Leading zero is optional when using the SET command).
P300 Response: `< REP PRESET nn >`	Where nn is the preset number 01-10.
Get Preset Name	

Command String:
 	
`< GET PRESET1 >`
`< GET PRESET2 >`
`< GET PRESET3 >`
etc
	Send one of these commands to the P300.

P300 Response:
 	
`< REP PRESET1 {yyyyyyyyyyyyyyyyyyyyyyyyy} >`
`< REP PRESET2 {yyyyyyyyyyyyyyyyyyyyyyyyy} >`
`< REP PRESET3 {yyyyyyyyyyyyyyyyyyyyyyyyy} >`
etc
	Where yyyyyyyyyyyyyyyyyyyyyyyyy *is 25 characters of the preset name. The P300 always responds with a 25 character device ID.*

Get Audio Gain	

Command String: `< GET xx AUDIO_GAIN_HI_RES >`	Where xx is ASCII channel number: 00 through 20; 22-28.
P300 Response: `< REP xx AUDIO_GAIN_HI_RES yyyy >`	Where yyyy takes on the ASCII values of 0000 to 1400. yyyy is in steps of one-tenth of a dB.
Set Audio Gain	
Command String: `< SET xx AUDIO_GAIN_HI_RES yyyy >`	Where xx is ASCII channel number: 01 through 20; 22-28. Where yyyy takes on the ASCII values of 0000 to 1400. yyyy is in steps of one-tenth of a dB.
P300 Response: `< REP xx AUDIO_GAIN_HI_RES yyyy >`	Where yyyy takes on the ASCII values of 0000 to 1400.
Increase Audio Gain by n dB	
Command String: `< SET xx AUDIO_GAIN_HI_RES INC nn >`	Where nn is the amount in one-tenth of a dB to increase the gain. nn can be single digit (n), double digit (nn), triple digit (nnn).
P300 Response: `< REP xx AUDIO_GAIN_HI_RES yyyy >`	Where yyyy takes on the ASCII values of 0000 to 1400.
Decrease Audio Gain by n dB	
Command String: `< SET xx AUDIO_GAIN_HI_RES DEC nn >`	Where nn is the amount in one-tenth of a dB to decrease the gain. nn can be single digit (n), double digit (nn), triple digit (nnn).
P300 Response: `< REP xx AUDIO_GAIN_HI_RES yyyy >`	Where yyyy takes on the ASCII values of 0000 to 1400.
Get Analog Input Gain Switch	
Command String: `< GET xx AUDIO_IN_LVL_SWITCH >`	Where xx is ASCII channel number: 00 or 11-12.

P300 Response:
 	
`< REP xx AUDIO_IN_LVL_SWITCH LINE_LVL >`
 	
`< REP xx AUDIO_IN_LVL_SWITCH AUX_LVL >`
	The P300 will respond with one of these strings.

Set Analog Input Gain Switch	

Command String:
 	
`< SET xx AUDIO_IN_LVL_SWITCH LINE_LVL >`
 	
`< SET xx AUDIO_IN_LVL_SWITCH AUX_LVL >`
	Where xx is ASCII channel number: 00, 11, or 12. Send one of these commands to the P300.

P300 Response:
 	
`< REP xx AUDIO_IN_LVL_SWITCH LINE_LVL >`
 	
`< REP xx AUDIO_IN_LVL_SWITCH AUX_LVL >`
	The P300 will respond with one of these strings.

Get Channel Audio Mute	
Command String: `< GET xx AUDIO_MUTE >`	Where xx is ASCII channel number: 00 through 20.

P300 Response:
 	
`< REP xx AUDIO_MUTE ON >`
 	
`< REP xx AUDIO_MUTE OFF >`
	The P300 will respond with one of these strings.

Mute Channel Audio	
Command String: `< SET xx AUDIO_MUTE ON >`	
P300 Response: `< REP xx AUDIO_MUTE ON >`	
Unmute Channel Audio	
Command String: `< SET xx AUDIO_MUTE OFF >`	
P300 Response: `< REP xx AUDIO_MUTE OFF >`	
Toggle Channel Audio Mute	
Command String: `< SET xx AUDIO_MUTE TOGGLE >`	
P300 Response: `< REP xx AUDIO_MUTE ON >`	The P300 will respond with one of these strings.

`< REP xx AUDIO_MUTE OFF >`	
Get Device Audio Mute	
Command String: `< GET DEVICE_AUDIO_MUTE >`	

P300 Response:
 	
`< REP DEVICE_AUDIO_MUTE ON >`
`< REP DEVICE_AUDIO_MUTE OFF >`
	The P300 will respond with one of these strings.

Set Device Audio Mute	

Command String:
 	
`< SET DEVICE_AUDIO_MUTE ON >`
`< SET DEVICE_AUDIO_MUTE OFF >`
`< SET DEVICE_AUDIO_MUTE TOGGLE >`
	Send one of these commands to the P300.

P300 Response:
 	
`< REP DEVICE_AUDIO_MUTE ON >`
`< REP DEVICE_AUDIO_MUTE OFF >`
	The P300 will respond with one of these strings.

Get Analog Output Gain Switch	
Command String: `< GET xx AUDIO_OUT_LVL_SWITCH >`	Where xx is ASCII channel number: 00, 17, or 18.

P300 Response:
 	
`< REP xx AUDIO_OUT_LVL_SWITCH LINE_LVL >`
`< REP xx AUDIO_OUT_LVL_SWITCH AUX_LVL >`
`< REP xx AUDIO_OUT_LVL_SWITCH MIC_LVL >`
	The P300 will respond with one of these strings.

Set Analog Output Gain Switch	

Command String:
 	
`< SET xx AUDIO_OUT_LVL_SWITCH LINE_LVL >`
`< SET xx AUDIO_OUT_LVL_SWITCH AUX_LVL >`
`< SET xx AUDIO_OUT_LVL_SWITCH MIC_LVL >`
	Where xx is ASCII channel number: 00, 17, or 18. Send one of these commands to the P300.

P300 Response:	The P300 will respond with one of these strings.

`< REP xx AUDIO_OUT_LVL_SWITCH LINE_LVL >`
 	
`< REP xx AUDIO_OUT_LVL_SWITCH AUX_LVL >`
 	
`< REP xx AUDIO_OUT_LVL_SWITCH MIC_LVL >`	

Set Flash Lights on P300	

Command String:
 	
`< SET FLASH ON >`
 	
`< SET FLASH OFF >`
	Send one of these commands to the P300. The flash automatically turns off after 30 seconds.

P300 Response:
 	
`< REP FLASH ON >`
 	
`< REP FLASH OFF >`
	The P300 will respond with one of these strings.

Get Flash Lights on P300	
Command String: `< GET FLASH >`	

P300 Response:
 	
`< REP FLASH ON >`
 	
`< REP FLASH OFF >`
	The P300 will respond with one of these strings.

Set Metering Rate Inputs	
Command String: `< SET METER_RATE_IN yyyyy >`	Where yyyyy is a value from 00000 to 99999 representing milliseconds. 00000=off; 00100=minimum value; 99999=maximum value. Note: values 00001 to 00099 are not valid and result in response.

P300 Response:
 	
`< REP METER_RATE_IN yyyyy >`
 	
`< SAMPLE_IN aaa bbb ccc ddd eee fff ggg hhh iii jjj kkk lll mmm nnn >`
	Where
	
	yyyyy
	
	is rate in milliseconds. Value 00000 means metering is off. Where aaa, bbb, etc is the value of the audio level received and is 000-060, which represent actual audio level of -60 to 0 dBFS.
	
	aaa
	
	is channel 1 data
	
	bbb

	is channel 2 data
	ccc
	is channel 3 data
	ddd
	is channel 4 data
	The sample data (aaa, bbb, ccc, ddd, etc.) appears in the following order, representing the 14 input channels:
	1-8: Dante Inputs with Mic Processing
	9-10: Dante Inputs
	11-12: Analog Inputs
	13: USB Input
	14: Mobile Input

Get Metering Rate Inputs

Command String: `< GET METER_RATE_IN >`	

P300 Response:
 	
`< REP METER_RATE_IN yyyyy >`
 	
`< SAMPLE_IN aaa bbb ccc ddd eee fff ggg hhh iii jjj kkk lll mmm nnn >`
	Where
	
	yyyyy
	
	is rate in milliseconds. Value 00000 means metering is off. Where aaa, bbb, etc is the value of the audio level received and is 000-060, which represent actual audio level of -60 to 0 dBFS.
	
	aaa
	
	is channel 1 data
	
	bbb
	
	is channel 2 data
	
	ccc
	
	is channel 3 data
	
	ddd
	
	is channel 4 data

	The sample data (aaa, bbb, ccc, ddd, etc.) appears in the following order, representing the 14 input channels:
	1-8: Dante Inputs with Mic Processing
	9-10: Dante Inputs
	11-12: Analog Inputs
	13: USB Input
	14: Mobile Input

Set Metering Rate Outputs

Command String: `< SET METER_RATE_OUT yyyyy >`	Where yyyyy is a value from 00000 to 99999 representing milliseconds. 00000=off; 00100=minimum value; 99999=maximum value. Note: values 00001 to 00099 are not valid and result in response.

P300 Response:
 	
`< REP METER_RATE_OUT yyyyy >`
 	
`< SAMPLE_OUT aaa bbb ccc ddd eee fff >`
	Where
	
	yyyyy
	
	is rate in milliseconds. Value 00000 means metering is off. Where aaa, bbb, etc is the value of the audio level received and is 000-060, which represent actual audio level of -60 to 0 dBFS.
	
	aaa
	
	is channel 1 data
	
	bbb
	
	is channel 2 data
	
	ccc
	
	is channel 3 data
	
	ddd
	
	is channel 4 data
	
	The sample data (aaa, bbb, ccc, ddd, etc.) appears in the following order:
	
	1-2: Dante outputs
	
	3-4: Analog outputs
	
	5: USB output

| | 6: Mobile output |
| | 7-12: Dante outputs 3-8 |

Get Metering Rate Outputs	
Command String: `< GET METER_RATE_OUT >`	

P300 Response:
`< REP METER_RATE_OUT yyyyy >`
`< SAMPLE_OUT aaa bbb ccc ddd eee fff >`
	Where
	
	yyyyy
	
	is rate in milliseconds. Value 00000 means metering is off. Where aaa, bbb, etc is the value of the audio level received and is 000-060, which represent actual audio level of -60 to 0 dBFS.
	
	aaa
	
	is channel 1 data
	
	bbb
	
	is channel 2 data
	
	ccc
	
	is channel 3 data
	
	ddd
	
	is channel 4 data
	
	The sample data (aaa, bbb, ccc, ddd, etc.) appears in the following order:
	
	1-2: Dante outputs
	
	3-4: Analog outputs
	
	5: USB output
	
	6: Mobile output
	
	7-12: Dante outputs 3-8

Set Metering Rate Processing Blocks	
Command String: `< SET METER_RATE_PROC yyyyy >`	Where yyyyy is a value from 00000 to 99999 representing milliseconds. 00000=off; 00100=minimum value; 99999=maximum value. Note: values 00001 to 00099 are not valid and result in response.

	Where yyyyy is rate in milliseconds. Value 00000 means metering is off. Where aaa, bbb, etc is the value of the audio level received and is 000-060, which represent actual audio level of -60 to 0 dBFS. aaa *is channel 1 data* bbb

P300 Response:
 	
`< REP METER_RATE_PROC yyyyy >`
 	
`< SAMPLE_PROC aaa bbb ccc ddd eee fff ggg hhh iii jjj kkk lll >`
	is channel 2 data
	
	ccc
	
	is channel 3 data
	
	ddd
	
	is channel 4 data
	
	The sample data (aaa, bbb, ccc, ddd, etc.) appears in the following order, representing the 12 channels:
	
	1-8: pre-AGC on Eight Dante Input Channels
	
	9: Automixer Output
	
	10: pre-Compressor
	
	11: AEC reference

Get Metering Rate Processing Blocks	
Command String: `< GET METER_RATE_PROC >`	

P300 Response:
 	
`< REP METER_RATE_PROC yyyyy >`
 	
`< SAMPLE_PROC aaa bbb ccc ddd eee fff ggg hhh iii jjj kkk lll >`
	Where
	
	yyyyy
	
	is rate in milliseconds. Value 00000 means metering is off. Where aaa, bbb, etc is the value of the audio level received and is 000-060, which represent actual audio level of -60 to 0 dBFS.
	
	aaa

	is channel 1 data
	bbb
	is channel 2 data
	ccc
	is channel 3 data
	ddd
	is channel 4 data
	The sample data (aaa, bbb, ccc, ddd, etc.) appears in the following order, representing the 12 channels:
	1-8: pre-AGC on Eight Dante Input Channels
	9: Automixer Output
	10: pre-Compressor
	11: AEC reference

Get LED Brightness

Command String: `< GET LED_BRIGHTNESS >`	
P300 Response: `< REP LED_BRIGHTNESS n >`	Where n can take on the following values: 0 = LED disabled 1 = LED dim 2 = LED default

Set LED Brightness

Command String: `< SET LED_BRIGHTNESS n >`	Where n can take on the following values: 0 = LED disabled 1 = LED dim 2 = LED default
P300 Response: `< REP LED_BRIGHTNESS n >`	

Get Audio IP Address	
Command String: `< GET IP_ADDR_NET_AUDIO_PRIMARY >`	
P300 Response: `< REP IP_ADDR_NET_AUDIO_PRIMARY {yyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyy is a 15 digit IP address.
Get Audio Subnet Address	
Command String: `< GET IP_SUBNET_NET_AUDIO_PRIMARY >`	
P300 Response: `< REP IP_SUBNET_NET_AUDIO_PRIMARY {yyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyy is a 15 digit subnet address.
Get Audio Gateway Address	
Command String: `< GET IP_GATEWAY_NET_AUDIO_PRIMARY >`	
P300 Response: `< REP IP_GATEWAY_NET_AUDIO_PRIMARY {yyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyy is a 15 digit gateway address.
Get Encryption Status	
Command String: `< GET ENCRYPTION >`	

P300 Response:
 	
`< REP ENCRYPTION ON >`
 	
`< REP ENCRYPTION OFF >`
	The P300 will respond with one of these strings.

Reboot P300	
Command String: `< SET REBOOT >`	
P300 Response: `< REP REBOOT >`	
Get Error Events	

Command String: `< GET LAST_ERROR_EVENT >`	Gets the last error that is logged on the P300.
P300 Response: `< REP LAST_ERROR_EVENT {yyyyyyyyyyyyyyyy} >`	Where yyyyyyyyyyyyyyyy is up to 128 characters.
Get PEQ Filter Enable	
Command String: `< GET xx PEQ yy >`	Where xx is the PEQ block; 00 means all PEQ blocks on P300; 01-08 are PEQ blocks on Dante Mic Inputs; 09-10 are EQ blocks on Dante Inputs; 11-12 are EQ blocks on Analog Inputs; 13 is EQ block on USB Input; 14 is EQ block on Mobile Input; 15-16 are PEQ blocks on Dante Outputs; 17-18 are PEQ blocks on Analog Outputs; 19 is PEQ block on USB Output; 21 is PEQ block after the automixer; 23-28 are the PEQ blocks on Dante outputs 3-8 (firmware 4.1.x and newer). Where yy is the filter number in the selected PEQ block xx. Valid values are as follows; 01-04: individual filter when xx is in range of 01-08, or 15-21; 01-02: individual filter when xx is in range of 09-14.

P300 Response:
 	
`< REP xx PEQ yy ON >`
 	
`< REP xx PEQ yy OFF >`	

Set PEQ Filter Enable	

Command String:
 	
`< SET xx PEQ yy ON >`
 	
`< SET xx PEQ yy OFF >`
	Send one of these commands to the P300.

P300 Response:
 	
`< REP xx PEQ yy ON >`
 	
`< REP xx PEQ yy OFF >`
	Where xx is the PEQ block 15-16, 17-18, 19, or 21. Where yy is the PEQ filter 01-04 within the block. 00 can be used for all blocks or all filters.

Get Input Meter Display Mode	
Command String: `< GET INPUT_METER_MODE >`	

P300 Response:
 	
`< REP INPUT_METER_MODE PRE_FADER >`
 	
`< REP INPUT_METER_MODE POST_FADER >`
	The P300 will respond with one of these strings.

Set Input Meter Display Mode	

Command String:
 	
`< SET INPUT_METER_MODE PRE_FADER >`
 	
`< SET INPUT_METER_MODE POST_FADER >`
	Send one of these commands to the P300.

P300 Response:
 	
`< REP INPUT_METER_MODE PRE_FADER >`
 	
`< REP INPUT_METER_MODE POST_FADER >`
	The P300 will respond with one of these strings.

Get Output Meter Display Mode	
Command String: `< GET OUTPUT_METER_MODE >`	

P300 Response:
 	
`< REP OUTPUT_METER_MODE PRE_FADER >`
 	
`< REP OUTPUT_METER_MODE POST_FADER >`
	The P300 will respond with one of these strings.

Set Output Meter Display Mode	

Command String:
 	
`< SET OUTPUT_METER_MODE PRE_FADER >`
 	
`< SET OUTPUT_METER_MODE POST_FADER >`
	Send one of these commands to the P300.

P300 Response:
 	
`< REP OUTPUT_METER_MODE PRE_FADER >`
 	
`< REP OUTPUT_METER_MODE POST_FADER >`
	The P300 will respond with one of these strings.

Get USB Connection Status	
Command String: `< GET USB_CONNECT >`	
P300 Response: `< REP USB_CONNECT ON >`	The P300 will respond with one of these strings.

`< REP USB_CONNECT OFF >`	
Get Matrix Mixer Routing	
Command String: `< GET xx MATRIX_MXR_ROUTE yy >`	Where xx is input channel numbers 21 or 0-14. Where yy is output channel numbers 0, 15-20, or 23-28 (firmware 4.1.x and newer).

P300 Response:
 	
`< REP xx MATRIX_MXR_ROUTE yy ON >`
 	
`< REP xx MATRIX_MXR_ROUTE yy OFF >`
	The P300 will respond with one of these strings.

Set Matrix Mixer Routing	

Command String:
 	
`< SET xx MATRIX_MXR_ROUTE yy ON >`
 	
`< SET xx MATRIX_MXR_ROUTE yy OFF >`
	Where xx is input channel numbers 21 or 0-14. Where yy is output channel numbers 0, 15-20, or 23-28 (firmware 4.1.x and newer). Send one of these commands to the P300.

P300 Response:
 	
`< REP xx MATRIX_MXR_ROUTE yy ON >`
 	
`< REP xx MATRIX_MXR_ROUTE yy OFF >`
	The P300 will respond with one of these strings.

Get Matrix Mixer Gain	
Command String: `< GET xx MATRIX_MXR_GAIN yy >`	Where xx is input channel numbers 21 or 0-14. Where yy is output channel numbers 0, 15-20, or 23-28 (firmware 4.1.x and newer).
P300 Response: `< REP xx MATRIX_MXR_GAIN yy zzzz >`	Where zzzz takes on the ASCII values of 0000 to 1400. zzzz is in steps of one-tenth of a dB.
Set Matrix Mixer Gain	
Command String: `< SET xx MATRIX_MXR_GAIN yy zzzz >`	Where xx is input channel numbers 21 or 9-14. Where yy is output channel numbers 15-20 or 23-28 (firmware 4.1.x and newer). Where zzzz takes on the ASCII values of 0000 to 1400. zzzz is in steps of one-tenth of a dB.
P300 Response: `< REP xx MATRIX_MXR_GAIN yyzzzz >`	
Increment Matrix Mixer Gain	
Get Matrix Mixer Gain	

Command String: `< SET xx MATRIX_MXR_GAIN yy INC nn >`	Where xx is input channel numbers 21 or 9-14. Where yy is output channel numbers 15-20 or 23-28 (firmware 4.1.x and newer). Where nn is in steps of one-tenth of a dB.
P300 Response: `< REP xx MATRIX_MXR_GAIN yy zzzz >`	Where zzzz takes on the ASCII values of 0000 to 1400. zzzz is in steps of one-tenth of a dB.
Decrement Matrix Mixer Gain	
Command String: `< SET xx MATRIX_MXR_GAIN yy DEC nn >`	Where xx is input channel numbers 21 or 9-14. Where yy is output channel numbers 15-20 or 23-28 (firmware 4.1.x and newer). Where nn is in steps of one-tenth of a dB.
P300 Response: `< REP xx MATRIX_MXR_GAIN yy zzzz >`	Where zzzz takes on the ASCII values of 0000 to 1400. zzzz is in steps of one-tenth of a dB.
Get Control Network MAC Address	
Command String: `< GET CONTROL_MAC_ADDR >`	
P300 Response: `< REP CONTROL_ MAC_ADDR yy:yy:yy:yy:yy:yy >`	Where yy:yy:yy:yy:yy:yy is a 17 char literal string formatted as 6 octets, each separated by a colon. Example: 00:0E:DD:FF:F1:63
Get Network Audio Channel Name	
Command String: `< GET xx NA_CHAN_NAME >`	Where xx is channel number All channels: 0 P300: 1-10, 15-16
P300 Response: `< REP xx NA_CHAN_NAME {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >`	Where xx is channel number. Where {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} is 31 char channel name. Value is padded with spaces as needed to ensure that 31 char are always reported.
Get Network Audio Device Name	
Command String: `< GET NA_DEVICE_NAME >`	
P300 Response: `< REP NA_DEVICE_NAME {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} >`	Where {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} is a text string. Most devices allow device

	id to be up to 31characters. Value is padded with spaces as needed to ensure that 31 char are always reported.
Restore Default Settings	
Command String: `< SET DEFAULT_SETTINGS >`	Request the device to set itself to default settings.
P300 Response: `< REP PRESET xx >`	Where xx = 00 if restore is successful.
Get AEC State	
Command String: `< GET xx AEC >`	Where xx is the channel number: All Dante Mic Channels: 00; P300 Dante Channel with Mic Processing: 01-08.

P300 Response:
 	
`< REP xx AEC ON >`
 	
`< REP xx AEC OFF >`
	The P300 will respond with one of these strings.

Set AEC State	

Command String:
 	
`< SET xx AEC ON >`
 	
`< SET xx AEC OFF >`
 	
`< SET xx AEC TOGGLE >`
	Send one of these commands to the P300.

P300 Response:
 	
`< REP xx AEC ON >`
 	
`< REP xx AEC OFF >`
	The P300 will respond with one of these strings.

Get AEC Reference Signal	
Command String: `< GET xx AEC_REF >`	Where xx is channel number that can be 00 or 22.
P300 Response: `< REP xx AEC_REF n >`	Where xx is channel number. Where n can take on the following values: DANTEOUT1 DANTEOUT2 DANTEOUT3 DANTEOUT4

	DANTEOUT5
	DANTEOUT6
	DANTEOUT7
	DANTEOUT8
	ANALOGOUT1
	ANALOGOUT2
	DANTEIN9
	DANTEIN10
	ANALOGIN1
	ANALOGIN2
	USBIN
	MOBILEIN

Set AEC Reference Signal

Command String: `< SET xx AEC_REF n >`	Where xx is channel number. Where n can take on the following values: DANTEOUT1 DANTEOUT2 DANTEOUT3 DANTEOUT4 DANTEOUT5 DANTEOUT6 DANTEOUT7 DANTEOUT8 ANALOGOUT1 ANALOGOUT2 DANTEIN9 DANTEIN10 ANALOGIN1 ANALOGIN2 USBIN MOBILEIN
P300 Response: `< REP xx AEC_REF n >`	Where xx is channel number. Where n can take on the following values: DANTEOUT1 DANTEOUT2 DANTEOUT3 DANTEOUT4 DANTEOUT5 DANTEOUT6 DANTEOUT7 DANTEOUT8 ANALOGOUT1 ANALOGOUT2 DANTEIN9 DANTEIN10 ANALOGIN1 ANALOGIN2

	USBIN MOBILEIN
Set ERLE Meter Rate	
Command String: `< SET METER_RATE_ERLE yyyyy >`	Where yyyyy is a value from 00000 to 99999 representing milliseconds. 00000 = off; 00100 = minimum value; 99999=maximum value. Note: values 00001 to 00099 are not valid and result in response.

P300 Response:
`< REP METER_RATE_ERLE yyyyy >`
`< SAMPLE_ERLE aaa bbb ccc ddd eee fff ggg hhh >`
	Where
	
	yyyyy
	
	= rate in milliseconds. Value 00000 means metering is off.
	
	Where aaa, bbb, etc is the sample for each channel. ERLE data is in 1 dB increment and is in the range of 00 to 40 dB
	
	aaa
	
	= channel 1 data
	
	bbb
	
	= channel 2 data
	
	ccc
	
	= channel 3 data
	
	ddd
	
	= channel 4 data
	
	eee
	
	= channel 5 data
	
	fff
	
	= channel 6 data
	
	ggg
	
	= channel 7 data
	
	hhh
	
	= channel 8 data

Get ERLE Meter Rate	

Command String:	
`< GET METER_RATE_ERLE >`	
P300 Response: `< REP METER_RATE_ERLE yyyyy >`	Where yyyyy = rate in milliseconds. Value 00000 means metering is off.
Get NLP State	
Command String: `< GET xx AEC_NLP >`	Where xx is the channel number: All Dante Mic Channels: 00; P300 Dante Channel with Mic Processing: 01-08.

P300 Response:
 	
`< REP xx AEC_NLP LOW >`
 	
`< REP xx AEC_NLP MEDIUM >`
 	
`< REP xx AEC_NLP HIGH >`
	The P300 will respond with one of these commands.

Set NLP State	

Command String:
 	
`< SET xx AEC_NLP LOW >`
 	
`< SET xx AEC_NLP MEDIUM >`
 	
`< SET xx AEC_NLP HIGH >`
	The P300 will respond with one of these commands.

P300 Response:
 	
`< REP xx AEC_NLP LOW >`
 	
`< REP xx AEC_NLP MEDIUM >`
 	
`< REP xx AEC_NLP HIGH >`
	The P300 will respond with one of these commands.

Get Noise Reduction State	
Command String: `< GET xx NOISE_RED >`	Where xx is the channel number: All Dante Mic Channels: 00; P300 Dante Channel with Mic Processing: 01-08.

P300 Response:
 	
`< REP xx NOISE_RED ON >`
 	
`< REP xx NOISE_RED OFF >`
	The P300 will respond with one of these commands.

Set Noise Reduction State

Command String:	

`< SET xx NOISE_RED ON >`
 	
`< SET xx NOISE_RED OFF >`
	The P300 will respond with one of these commands.

P300 Response:	

`< REP xx NOISE_RED ON >`
 	
`< REP xx NOISE_RED OFF >`
	The P300 will respond with one of these commands.

Get Noise Reduction Level

Command String:	
`< GET xx NOISE_RED_LVL >`	Where xx is the channel number: All Dante Mic Channels: 00; P300 Dante Channel with Mic Processing: 01-08.

P300 Response:	

`< REP xx NOISE_LVL LOW >`
 	
`< REP xx NOISE_RED_LVL MEDIUM >`
 	
`< REP xx NOISE_RED HIGH >`
	The P300 will respond with one of these commands.

Set Noise Reduction Level

Command String:	

`< SET xx NOISE_RED_LVL LOW >`
 	
`< SET xx NOISE_RED_LVL MEDIUM >`
 	
`< SET xx NOISE_RED_LVL HIGH >`
	The P300 will respond with one of these commands.

P300 Response:	

`< REP xx NOISE_RED_LVL LOW >`
 	
`< REP xx NOISE_RED_LVL MEDIUM >`
 	
`< SET xx NOISE_RED_LVL HIGH >`
	The P300 will respond with one of these commands.

Get AGC State

Command String:	
`< GET xx AGC >`	Where xx is the channel number: All Dante Mic Channels: 00; P300 Dante Channel with Mic Processing: 01-08.

P300 Response:	
`< REP xx AGC ON >`	The P300 will respond with one of these strings.

`< REP xx AGC OFF >`	
Set AGC State	

Command String:
 	
`< SET xx AGC ON >`
 	
`< SET xx AGC OFF >`
 	
`< SET xx AGC TOGGLE >`
	Send one of these commands to the P300.

P300 Response:
 	
`< REP xx AGC ON >`
 	
`< REP xx AGC OFF >`
	The P300 will respond with one of these strings.

Set AGC Metering Rate	
Command String: `< SET METER_RATE_AGC yyyyy >`	Where yyyyy is a value from 00000 to 99999 representing milliseconds. 00000 = off; 00100 = minimum value; 99999=maximum value. Note: values 00001 to 00099 are not valid and result in response.

P300 Response:
 	
`< REP METER_RATE_AGC yyyyy >`
 	
`< SAMPLE_AGC aaa bbb ccc ddd eee fff ggg hhh >`
	Where yyyyy = rate in milliseconds. Value 00000 means metering is off.
	
	Where aaa, bbb, etc is the sample for each channel. ERLE data is in 1 dB increment and is in the range of 00 to 40 dB
	
	aaa
	
	= channel 1 data
	
	bbb
	
	= channel 2 data
	
	ccc
	
	= channel 3 data
	
	ddd
	
	= channel 4 data
	
	eee
	
	= channel 5 data
	
	fff

	= channel 6 data
	ggg
	= channel 7 data
	hhh
	= channel 8 data
	AGC Gain data is in 1 dB increment. The reported data is scaled by 20 so the range is 00 to 40, representing an actual range of -20 to + 20 dB. -20 dB is represented as 00; 0 dB is represented as 20; +20 dB is represented as 40.

Get AGC Metering Rate

Command String:	
`< GET METER_RATE_AGC >`	

	Where yyyyy = rate in milliseconds. Value 00000 means metering is off.
	Where aaa, bbb, etc is the sample for each channel. ERLE data is in 1 dB increment and is in the range of 00 to 40 dB
	aaa
	= channel 1 data
	bbb
	= channel 2 data
	ccc
P300 Response:	= channel 3 data
`< REP METER_RATE_AGC yyyyy >`	ddd
`< SAMPLE_AGC aaa bbb ccc ddd eee fff ggg hhh >`	= channel 4 data
	eee
	= channel 5 data
	fff
	= channel 6 data
	ggg
	= channel 7 data

	hhh
	= channel 8 data
	AGC Gain data is in 1 dB increment. The reported data is scaled by 20 so the range is 00 to 40, representing an actual range of -20 to + 20 dB. -20 dB is represented as 00; 0 dB is represented as 20; +20 dB is represented as 40.

Get AGC Max Cut Value

Command String: `< GET xx AGC_MAX_CUT >`	Where xx is channel number: All channels: 0; P300: 01-08.
P300 Response: `< REP xx AGC_MAX_CUT yyy >`	Where xx is channel number in the range of 01-08. Where yyy is AGC Max Cut data defined in SET command.

Set AGC Max Cut Value

Command String: `< SET xx AGC_MAX_CUT yyy >`	Where xx is channel number: All channels: 0; P300: 01-08. Where yyy is AGC Max Cut data in 0.1 dB increment. The actual range of -20.0 to 0.0 dB is shifted by 20.0 and then multiplied by 10 so user data has a range of 000 to 200. -20.0 dB is represented as 000 -12.3 dB is represented as 077 -1.2 dB is represented as 188.
P300 Response: `< REP xx AGC_MAX_CUT yyy >`	Where xx is channel number in the range of 01-08. Where yyy is AGC Max Cut data in 0.1 dB increment. The actual range of -20.0 to 0.0 dB is shifted by 20.0 and then multiplied by 10 so user data has a range of 000 to 200. -20.0 dB is represented as 000 -12.3 dB is represented as 077 -1.2 dB is represented as 188.

Set AGC Max Cut Value Increment

Command String: `< SET xx AGC_MAX_CUT inc nnn >`	Where xx is channel number: All channels: 0; P300: 01-08. Where nnn is in units of one-tenth of a dB. The request-

	ed is multiplied by 10 and is three digits long. 1.2 is represented as 012 12.3 is represented as 123 The resulting Cut when the nnn is applied must be in the range of 000-200.
P300 Response: `< REP xx AGC_MAX_CUT yyy >`	Where xx is channel number in the range of 01-08. Where yyy is AGC Max Cut data defined in SET command.
Set AGC Max Cut Value Decrement	
Command String: `< SET xx AGC_MAX_CUT dec nnn >`	Where xx is channel number: All channels: 0; P300: 01-08. Where nnn is in units of one-tenth of a dB. The requested is multiplied by 10 and is three digits long. 1.2 is represented as 012 12.3 is represented as 123 The resulting Cut when the nnn is applied must be in the range of 000-200.
P300 Response: `< REP xx AGC_MAX_CUT yyy >`	Where xx is channel number in the range of 01-08. Where yyy is AGC Max Cut data defined in SET command.
Get AGC Max Boost Value	
Command String: `< GET xx AGC_MAX_BOOST >`	Where xx is channel number: All channels: 0; P300: 01-08.
P300 Response: `< REP xx AGC_MAX_BOOST yyy >`	Where xx is channel number in the range of 01-08. Where yyy is AGC Max Boost data defined in SET command.
Set AGC Max Boost Value	
Command String:	Where xx is channel number: All channels: 0; P300: 01-08. Where yyy is AGC Max Boost data. The range is 000 to

`< SET xx AGC_MAX_BOOST yyy >`	200, representing an actual range of 0.0 to +20.0 dB in 0.1 dB increment. +12.3 dB is represented as 123 +1.2 dB is represented as 012
P300 Response: `< REP xx AGC_MAX_BOOST yyy >`	Where xx is channel number in the range of 01-08. Where yyy is *AGC Max Boost data. The range is 000 to 200, representing an actual range of 0.0 to +20.0 dB in 0.1 dB increment.* +12.3 dB is represented as 123 +1.2 dB is represented as 012
Set AGC Max Boost Value Increment	
Command String: `< SET xx AGC_MAX_BOOST inc nnn >`	Where xx is channel number: All channels: 0; P300: 01-08. Where nnn is in units of one-tenth of a dB. The resulting Boost when the nnn is applied must be in the range of 000-200.
P300 Response: `< REP xx AGC_MAX_BOOST yyy >`	Where xx is channel number in the range of 01-08. Where yyy is AGC Max Boost data defined in SET command.
Set AGC Max Boost Value Decrement	
Command String: `< SET xx AGC_MAX_BOOST dec nnn >`	Where xx is channel number: All channels: 0; P300: 01-08. Where nnn is in units of one-tenth of a dB. The resulting Boost when the nnn is applied must be in the range of 000-200.
P300 Response: `< REP xx AGC_MAX_BOOST yyy >`	Where xx is channel number in the range of 01-08. Where yyy is AGC Max Boost data defined in SET command.
Get AGC Target Level	
Command String: `< GET xx AGC_TARGET >`	Where xx is channel number: All channels: 0; P300: 01-08.
P300 Response: `< REP xx AGC_TARGET yyy >`	Where xx is channel number in the range of 01-08. Where yyy is AGC Target data defined in SET command.
Set AGC Target Level	

Command String: `< SET xx AGC_TARGET yyy >`	Where xx is channel number: All channels: 0; P300: 01-08. Where yyy is AGC Target Level data in 0.1 dBFS increment. Actual range of -50.0 to 0.0 dBFS is shifted by 50 then multiplied by 10, resulting in user data in the range of 000 to 500.. -50.0 is represented as 000 -12.3 is represented as 377 -1.2 is represented as 488
P300 Response: `< REP xx AGC_TARGET yyy >`	Where xx is channel number in the range of 01-08. Where yyy is *AGC Target Level data in 0.1 dBFS increment. Actual range of -50.0 to 0.0 dBFS is shifted by 50 then multiplied by 10, resulting in user data in the range of 000 to 500..* -50.0 is represented as 000 -12.3 is represented as 377 -1.2 is represented as 488
Set AGC Target Level Increment	
Command String: `< SET xx AGC_TARGET inc nnn >`	Where xx is channel number: All channels: 0; P300: 01-08. Where nnn is in units of one-tenth of a dBFS. The requested nnn is multiplied by 10 and is three digits long: 1.2 is represented as 012 12.3 is represented as 123 The resulting Target when the nnn is applied must be in the range 000-200.
P300 Response: `< REP xx AGC_TARGET yyy >`	Where xx is channel number in the range of 01-08. Where yyy is AGC Target data defined in SET command.
Set AGC Target Level Decrement	

	Where xx is channel number: All channels: 0; P300: 01-08. Where nnn is in units of one-tenth of a dBFS. The requested nnn is multiplied by 10 and is three digits long:
	1.2 is represented as 012
Command String:	12.3 is represented as 123
`< SET xx AGC_TARGET dec n >`	The resulting Target when the
	nnn
	is applied must be in the range of 000-200.
P300 Response: `< REP xx AGC_TARGET yyy >`	Where xx is channel number in the range of 01-08. Where yyy is AGC Target data defined in SET command.
Get Gate Inhibit State (Only works on firmware < 4.1.x)	
Command String: `< GET 22 GATE_INHIBIT >`	
P300 Response: `< REP 22 GATE_INHIBIT ON >` `< REP 22 GATE_INHIBIT OFF >`	The P300 will respond with one of these strings.
Set Gate Inhibit State (Only works on firmware < 4.1.x)	
Command String: `< SET 22 GATE_INHIBIT ON >` `< SET 22 GATE_INHIBIT OFF >` `< SET 22 GATE_INHIBIT TOGGLE >`	Send one of these commands to the P300.
P300 Response: `< REP 22 GATE_INHIBIT ON >` `< REP 22 GATE_INHIBIT OFF >`	The P300 will respond with one of these strings.
Get Automixer Mode	
Command String: `< GET xx AUTOMXR_MODE >`	Where xx is the automixer channel number, 00 or 21.

P300 Response:
 	
`< REP xx AUTOMXR_MODE MANUAL >`
 	
`< REP xx AUTOMXR_MODE GAINSHARE >`
 	
`< REP xx AUTOMXR_MODE GATING >`
	The P300 will respond with one of these strings.

Set Automixer Mode	

Command String:
 	
`< SET xx AUTOMXR_MODE MANUAL >`
 	
`< SET xx AUTOMXR_MODE GAINSHARE >`
 	
`< SET xx AUTOMXR_MODE GATING >`
	Send one of these commands to the P300.

P300 Response:
 	
`< REP xx AUTOMXR_MODE MANUAL >`
 	
`< REP xx AUTOMXR_MODE GAINSHARE >`
 	
`< REP xx AUTOMXR_MODE GATING >`
	The P300 will respond with one of these strings.

Get Automixer Off Attenuation	
Command String: `< GET xx AUTOMXR_OFF_ATT >`	Where xx is the automixer channel number, 00 or 21.
P300 Response: `< REP xx AUTOMXR_OFF_ATT yyy >`	Where xx is automixer channel number. Where yyy is automixer off attenuation.
Set Automixer Off Attenuation	
Command String: `< SET xx AUTOMXR_OFF_ATT yyy >`	Where xx is automixer channel number. Where yyy is automixer off attenuation in 1 dB increment. Actual range of -110 to -3 dB is shifted by 110, so user data is in the range of 0 to 107. -110 is represented as 000 -10 is represented as 100 -3 is represented as 107
P300 Response: `< REP xx AUTOMXR_OFF_ATT yyy >`	Where xx is automixer channel number. Where yyy is automixer off attenuation.
Get Automixer Gating Sensitivity	

Command String: `< GET xx AUTOMXR_GATE_SEN >`	Where xx is the automixer channel number, 00 or 21.
P300 Response: `< REP xx AUTOMXR_GATE_SEN y >`	Where xx is automixer channel number. Where y is automixer gating sensitivity.
Set Automixer Gating Sensitivity	
Command String: `< SET xx AUTOMXR_GATE_SEN y >`	Where xx is automixer channel number. Where y is automixer gating sensitivity in the range of 1 to 9, in increment of 1.
P300 Response: `< REP xx AUTOMXR_GATE_SEN y >`	Where xx is automixer channel number. Where y is automixer gating sensitivity.
Set Automixer Gating Sensitivity Increment	
Command String: `< SET xx AUTOMXR_GATE_SEN inc n >`	Where xx is automixer channel number. Where nis increment step. The value after n is applied cannot exceed the range of 1 to 9.
P300 Response: `< REP xx AUTOMXR_GATE_SEN y >`	Where xx is automixer channel number. Where y is automixer gating sensitivity.
Set Automixer Gating Sensitivity Decrement	
Command String: `< SET xx AUTOMXR_GATE_SEN dec n >`	Where xx is automixer channel number. Where nis decrement step. The value after n is applied cannot exceed the range of 1 to 9.
P300 Response: `< REP xx AUTOMXR_GATE_SEN y >`	Where xx is automixer channel number. Where y is automixer gating sensitivity.
Get Automixer Maximum Number of Mics	
Command String: `< GET xx AUTOMXR_MAX_NOM >`	Where xx is the automixer channel number, 00 or 21.
P300 Response: `< REP xx AUTOMXR_MAX_NOM y >`	Where xx is automixer channel number. Where y is automixer Max NOM.
Set Automixer Maximum Number of Mics	

Command String: `< SET xx AUTOMXR_MAX_NOM y >`	Where xx is automixer channel number. Where y is automixer Max NOM in the range of 1 to 8, in increment of 1.
P300 Response: `< REP xx AUTOMXR_MAX_NOM y >`	Where xx is automixer channel number. Where y is automixer Max NOM.
Get Automixer Last Mic Lock On	
Command String: `< GET xx AUTOMXR_LMLO >`	Where xx is the automixer channel number, 00 or 21.

P300 Response:
 	
`< REP xx AUTOMXR_LMLO ON >`
 	
`< REP xx AUTOMXR_LMLO OFF >`
	The P300 will respond with one of these strings.

Set Automixer Last Mic Lock On	

Command String:
 	
`< SET xx AUTOMXR_LMLO ON >`
 	
`< SET xx AUTOMXR_LMLO OFF >`
 	
`< SET xx AUTOMXR_LMLO TOGGLE >`
	Send one of these commands to the P300.

P300 Response:
 	
`< REP xx AUTOMXR_LMLO ON >`
 	
`< REP xx AUTOMXR_LMLO OFF >`
	The P300 will respond with one of these strings.

Get Automixer Hold Time	
Command String: `< GET xx AUTOMXR_HOLDTIME >`	Where xx is the automixer channel number, 00 or 21.
P300 Response: `< REP xx AUTOMXR_HOLDTIME yyyy >`	Where xxis automixer channel number. Where yyyy is automixer Hold Time.
Set Automixer Hold Time	
Command String: `< SET xx AUTOMXR_HOLDTIME yyyy >`	Where xx is automixer channel number. Where yyyy is automixer Hold Time in the range of 0100 to 1500 ms, in increment of 1 ms.
P300 Response:	Where xx is automixer channel number. Where yyyy is automixer Hold Time.

`< REP xx AUTOMXR_HOLDTIME yyyy >`	
Get Automixer Gating Optimization	
Command String: `< GET xx AUTOMXR_GATE_OPT >`	Where xx is the automixer channel number, 00 or 21.
P300 Response: `< REP xx AUTOMXR_GATE_OPT mode >`	Where **mode** is the automixer gating optimization setting: NORMAL MXA310 MXA710 MXA910
Set Automixer Gating Optimization	
Command String: `< SET xx AUTOMXR_GATE_OPT mode >`	Where xx is the automixer channel number, 00 or 21. Where **mode** is the automixer gating optimization setting: NORMAL MXA310 MXA710 MXA910
P300 Response: `< REP xx AUTOMXR_GATE_OPT mode >`	
Get Automixer Channel Always On	
Command String: `< GET xx AUTOMXR_ALWAYS_ON >`	Where xx is the channel number: All: 00; P300: 01-08.

P300 Response:
 	
`< REP xx AUTOMXR_ALWAYS_ON ON >`
 	
`< REP xx AUTOMXR_ALWAYS_ON OFF >`
	The P300 will respond with one of these strings.

Set Automixer Channel Always On	
Command String:	Send one of these commands to the P300.

`< SET xx AUTOMXR_ALWAYS_ON ON >`
 	
`< SET xx AUTOMXR_ALWAYS_ON OFF >`
 	
`< SET xx AUTOMXR_ALWAYS_ON TOGGLE >`	

P300 Response:
 	
`< REP xx AUTOMXR_ALWAYS_ON ON >`
 	
`< REP xx AUTOMXR_ALWAYS_ON OFF >`
	The P300 will respond with one of these strings.

Get Automixer Channel Priority	
Command String: `< GET xx AUTOMXR_PRIORITY >`	Where xx is the channel number: All: 00; P300: 01-08.

P300 Response:
 	
`< REP xx AUTOMXR_PRIORITY ON >`
 	
`< REP xx AUTOMXR_PRIORITY OFF >`
	The P300 will respond with one of these strings.

Set Automixer Channel Priority	

Command String:
 	
`< SET xx AUTOMXR_PRIORITY ON >`
 	
`< SET xx AUTOMXR_PRIORITY OFF >`
 	
`< SET xx AUTOMXR_PRIORITY TOGGLE >`
	Send one of these commands to the P300.

P300 Response:
 	
`< REP xx AUTOMXR_PRIORITY ON >`
 	
`< REP xx AUTOMXR_PRIORITY OFF >`
	The P300 will respond with one of these strings.

Get Automixer Post Gate Mute	
Command String: `< GET xx AUTOMXR_MUTE >`	Where xx is the channel number and can only be channel 00 or 21.

P300 Response:
 	
`< REP xx AUTOMXR_MUTE ON >`
 	
`< REP xx AUTOMXR_MUTE OFF >`
	The P300 will respond with one of these strings.

Set Automixer Post Gate Mute	
Command String:	Send one of these commands to the P300.

`< SET xx AUTOMXR_MUTE ON >`
 	
`< SET xx AUTOMXR_MUTE OFF >`
 	
`< SET xx AUTOMXR_MUTE TOGGLE >`	

P300 Response:
 	
`< REP xx AUTOMXR_MUTE ON >`
 	
`< REP xx AUTOMXR_MUTE OFF >`
	The P300 will respond with one of these strings.

Get Automixer Gate Status	
Command String: `< GET xx AUTOMXR_GATE >`	Where xx is the channel number: All: 00; Individual Channel: 01-08; Gate Inhibit Channel: 22.

P300 Response:
 	
`< REP xx AUTOMXR_GATE ON >`
 	
`< REP xx AUTOMXR_GATE OFF >`
	The P300 will respond with one of these strings.

Get Compressor State	
Command String: `< GET xx COMPRESSOR >`	Where xx is the channel number 00 or 21.

P300 Response:
 	
`< REP xx COMPRESSOR ON >`
 	
`< REP xx COMPRESSOR OFF >`
	The P300 will respond with one of these strings.

Set Compressor State	

Command String:
 	
`< SET xx COMPRESSOR ON >`
 	
`< SET xx COMPRESSOR OFF >`
 	
`< SET xx COMPRESSOR TOGGLE >`
	Send one of these commands to the P300.

P300 Response:
 	
`< REP xx COMPRESSOR ON >`
 	
`< REP xx COMPRESSOR OFF >`
	The P300 will respond with one of these strings.

Get Compressor Threshold	
Command String:	Where xx is the channel number 00 or 21.

`< GET xx COMP_THRESHOLD >`	
P300 Response: `< REP xx COMP_THRESHOLD yyy >`	Where xx is the channel number 00 or 21. Where yyy is Compressor Threshold data, as defined in SET command.
Set Compressor Threshold	
Command String: `< SET xx COMP_THRESHOLD yyy >`	Where xx is the channel number 00 or 21. Where yyy is Compressor Threshold data in 0.1 dB increment. Actual range of -60.0 to 0.0 dB is shifted by 60 then multiplied by 10, resulting in user data in the range of 000 to 600. -60.0 is represented as 000 -12.3 is represented as 477 -1.2 is represented as 588
P300 Response: `< REP xx COMP_THRESHOLD yyy >`	Where xx is the channel number 00 or 21. Where yyy is Compressor Threshold data, as defined in SET command.
Get Compressor Ratio	
Command String: `< GET xx COMP_RATIO >`	Where xx is the channel number 00 or 21.
P300 Response: `< REP xx COMP_RATIO yyyy >`	Where xx is the channel number 00 or 21. Where yyyy is Compressor Ratio data, as defined in SET command.
Set Compressor Ratio	
Command String: `< SET xx COMP_RATIO yyyy >`	Where xx is the channel number 00 or 21. *Where yyyy is Compressor Ratio data in 0.1 increment. Ratio data is in the range of 1000 to 0010, representing actual range of 100.0:1 to 1.0:1 ratio.* 100.0:1 is represented as 1000 12.3:1 is represented as 0123 1.2:1 is represented as 0012
P300 Response: `< REP xx COMP_RATIO yyyy >`	Where xx is the channel number 00 or 21. Where yyyy is Compressor Ratio data, as defined in SET command.
Get Delay	

Command String: `< GET xx DELAY >`	Where xxis the channel number : All Channels: 0 Analog Out 1: 17 Analog Out 2: 18 USB Out: 19
P300 Response: `< REP xx DELAY yyyy >`	Where xx is channel number defined in GET command. Where yyyy is Delay, as defined in SET command.
Set Delay	
Command String: `< SET xx DELAY yyyy >`	Where xx is channel number defined in the GET command. Where yyyyis Delay data in 1 ms increment. Delay is in the range of 0 to 1000 ms, 0 means Delay unit is disabled.
P300 Response: `< REP xx DELAY yyyy >`	Where xx is channel number defined in GET command. Where yyyy is Delay, as defined in SET command.
Get or Set Direct Out Tap Point (firmware 4.1.x and newer)	
Command String: `< GET nn DIRECTOUT_POINT >`	Get current direct output tap point, where nn is: 0: All channels 01-08: Specific Dante input channel number
Command String: `< SET nn DIRECTOUT_POINT xx >`	Set current direct output tap point, where xx is: 0: Pre-gate/Pre Processing 1: Pre-gate/Post Processing 2: Post-gate/Pre Processing 3: Post Gate/Post Processing
P300 Response:	

`< REP nn DIRECTOUT_POINT xx >`	
Enable or Disable Call Status	

Command String:
 	
`< GET ONHOOK_ENABLE >`
 	
`< SET ONHOOK_ENABLE state >`
	Enables or disables Call status feature.
	state can be:
	
	1. ON
	2. OFF

P300 Response: `< REP ONHOOK_ENABLE state >`	
Report Call Status State	
Command String: `< GET ONHOOK_STATE >`	
P300 Response: `< REP ONHOOK_STATE state >`	When Call status is enabled, state can be: 1. ONHOOK (not in a call) 2. OFFHOOK (in a call) When Call status is disabled, state is OFFHOOK.