

---

---

## RS-232 / TCP/IP Port Hardware 3

TCP/IP Overview 3

Setting a static IP address vs DHCP 3

TCP/IP settings used by the switch 4

RS-232 Pinout and Baudrate Settings 4

Pin definitions 4

Port settings used by the switch 4

Timing information (unless specified otherwise by a command's description) 4

## Command Syntax 5

Command Syntax 5

Command Responses 5

Type of Responses and Timing Information 5

The Acknowledgement Response 6

The Error Response 6

The Query Response 7

Reference for Basic Control Commands 7

Definitions 7

Zone 7

Source 7

Using Bitmapped Parameters 7

Reading / Writing Bitmapped Parameters 7

## Input / Output Numbering 9

The Default (Native) Source Numbering 9

The ProAudio16 native source numbering 10

The ProAudio1632 native source numbering 10

The ProAudio32, ProAudio48 & ProAudio64 native source numbering 11

The Extended I/O mode 12

PCM-stereo mirroring analog outputs 12

Example ProAudio16 13

## Power, Switching, and Muting 14

Basic Control Commands 14

'P' Power Control 14

'SZ' Set Zone(s) -- Analog Audio Switch 14

'DSZ' Set Zone(s) -- Digital Audio Switch 15

'MZ' Mute Zone(s) -- Analog Audio Switch 16

'PZ' Power Down (Disconnect) Zone(s) -- Analog Audio Switch 18

'DMZ' Mute (Disconnect / Power down) Zone(s) -- Digital Audio Switch 19

## Advanced Control 21

Advanced Commands 21

Advanced Command Definitions 21

'V' Version 21

'DZ', 'DDZ', Set Zone Switching Delays 21

'XS' Control Settings 22

*'ASY' Send commands asynchronously any time a parameter changes 23*

*'ACK' Enable / Disable Acknowledgements 23*

*'ECO' Enable / Disable the 'Parameter Changed' Strings 23*

*'CRE' Enable / Disable trailing Carriage Returns Line Feeds 23*

*'XIO' Enable / Disable Extended I/O numbering 23*

*'MJP' Enable / Disable Mute Volume Jump when using Volume Increment / Decrement commands 23*

*'UVL' Enable / Disable Volume Unmute when using direct level Volume Commands 24*

'SS' Save Default Power On Settings 24

'FS' Reset to Factory Settings 25

'LI' Lighting Mode and Intensities 26

TCP/IP Control 27

'IPSET' Set the IP operation mode, DHCP or STATIC 27

'IPA' Set / View the static IP Address 28

'IPM' Set / View the static IP Mask 28

'IPG' Set / View the static IP gateway address 28

'IPAX' Retrieve the current IP address in use 29

'IPMX' Retrieve the current IP mask in use 29

'IPGX' Retrieve the current IP gateway address in use 29

## Audio Control 30

Audio Control Commands 30

Audio Command Definitions 30

'LZ' Lock a Zone to Another Zone 30

---

'MV' Set Master Volume 30  
'VZ' Set Zone's Volume 31  
'VPZ' Set Zone's Volume as a Percentage 32  
'VMIZ' Set Zone's Minimum Volume 33  
'VMAZ' Set Zone's Maximum Volume 34  
'VMZ' Audio Mute a Zone using Volume Levels 34  
'BLZ' Set Zone's Balance 35  
'GAZ' Set Zone's Gain (Output Levels) 36  
'GAI' Set Source's Gain (Input Trimming) 36  
'BAZ', 'TRZ' Set Zone's Bass and Treble Levels 36  
'EQ1Z', 'EQ2Z', 'EQ3Z', 'EQ4Z', 'EQ5Z' Set the 5 Band Equalizers' levels 38  
'FTYPZ' Low or High Pass Filter Type 39  
'FFRQZ' Set the Crossover Frequency of a Low or High Pass Filter 39  
'MXZ' Stereo to Mono Down Mix for a Zone 40  
'MXI' Stereo to Mono Down Mix for an Input 41  
'DRZ' Digital audio Routing 41  
'LSZ' Set Lip Sync delay for a Zone 42  
'LSI' Set Lip Sync delay for an Input 43  
'ATZ' Audio Type for an analog zone 44

## 12V Triggers 45

12V Trigger Overview 45  
'STRG' Set a 12V trigger 45  
'TRGZ' Set a zone to control a 12V trigger 45

## Paging and Doorbell 46

Paging and Doorbell Overview 46  
Command flow for setting up a page 46  
Back Panel Doorbell / Page Wiring 47  
Doorbell and Paging Commands 47  
'pASZ', 'pDSZ', Set the Paging Source of a Zone 47  
'pAMZ', 'pDMZ', Set the Muting of a Zone during a Page 48  
'pVZ', 'pVPZ' Set the Volume of a Zone during a Page 48  
'pVMZ', Set the Volume Mute level of a Zone during a Page 49  
'pADNDZ', 'pDDNDZ', Set the Do Not Disturb flags 49  
'pDSW', Setup the back panel doorbell / paging switches 49  
'pTIME' Set Initial Delays and Minimum Paging Times 50  
'pSET' Initiate or End a Page 51

## RS-232 / TCP/IP Port Hardware

### TCP/IP Overview

The Serial and TCP/IP port share the same protocol.

The TCP/IP connection is a very simple socket, sometimes referred to as Raw TCP/IP socket, similar to Telnet, but without the Telnet protocol overhead. Most telnet clients will allow you to telnet into the switch without error.

We use the open source package PuTTY to do our testing. It has a convenient “Raw” mode that works great with the switch, and is available in Windows and Linux (with a Mac O/S version in the works). (*We are not associated with PuTTY in anyway, but do find it a useful tool when communicating over TCP/IP and Serial port connections*)

Website: <http://www.chiark.greenend.org.uk/~sgtatham/putty>

By default, the switch uses DHCP to acquire an IP address. The TCP/IP socket's is fixed to 50005.

Once a connection is made it will remain open until closed by the client, or after 10 minutes of retries at attempting to talk to the client.

After connecting to the TCP/IP port, all commands are identical to those of the Serial port. All strings coming from the switch will be sent to both the TCP/IP sockets and Serial ports.

The switch will accept commands from both Serial ports and TCP/IP simultaneously, each port's commands will be buffered until the ending '\$' is read, at which time the commands will be executed in the order received. All responses will be sent to both Serial ports and TCP/IP connections.

### Setting a static IP address vs DHCP

The easiest way to set a static IP address is to go to our website and download the file zSetIP.zip. Inside the ZIP file is the utility zSetIP.exe. You must be connected to the switch using a serial port cable. Click on this file and it will walk you through setting a static IP address.

By default, the switch will use DHCP to retrieve a IP address, an IP mask, and the address of the router (gateway).

If using serial commands, setting an IP address is a two step process. You must first set the static values to be used for the IP address, the IP mask, and the router address, then place the switch into the Static IP address mode.

The commands to set the static IP address, mask and router are: IPA, IPM and IPG respectively. (*See: “IPA' Set / View the static IP Address” on page 28.*)

The command to change the IP address mode is: IPSET. (*See: “IPSET' Set the IP operation mode, DHCP or STATIC” on page 27.*)

The command to save these changes in EEPROM so that they are used after a power failure is: ^SS 32\$. (*See: “SS' Save Default Power On Settings” on page 24.*)

As an example, to set a static address of 192.168.1.200, a mask of 255.255.255.0 and a router address of 192.168.1.1, and save this in EEPROM, the following commands would be executed:

<code>^IPA 192,168,1,200\$</code>	- Set the static IP address
<code>^IPM 255,255,255,0\$</code>	- Set the static IP mask
<code>^IPG 192,168,1,1\$</code>	- Set the gateway address
<code>^IPSET 0\$</code>	- Set the TCP/IP mode to “Static”
<code>^SS 32\$</code>	- Save the new settings in EEPROM in case of power failure

# RS-232 / TCP/IP Port Hardware

---

## TCP/IP settings used by the switch

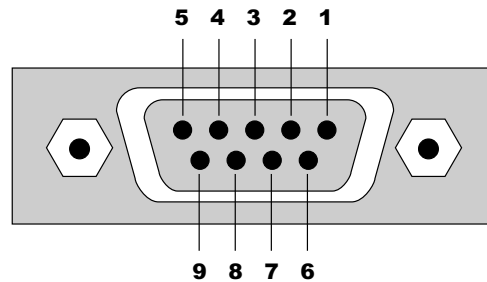
Default IP Address: DHCP  
Port Number: 50005  
Duplex: Full  
Speed: 10/100 Mbps

## RS-232 Pinout and Baudrate Settings

The RS-232 port on the switch is the same format, and pinout, as a PC modem, and uses the same type of cable as a standard serial modem would, which is a standard straight through cable. Do not use a cable that is marked as a “Null Modem” cable.

The switch can also be used with USB to RS-232 conversion cables, these are all typically straight through cables. *(Be sure to install any drivers that come with the USB to RS-232 cable you are using.)*

The RS-232 port is a female type DE-9 connector (sometimes mistakenly referred to as a DB-9 connector) with the following pinout:



## Pin definitions

1 - No Connect	6 - No Connect
2 - TX	7 - No Connect
3 - RX	8 - No Connect
4 - No Connect	9 - No Connect
5 - GND	

## Port settings used by the switch

Baudrate: 19200  
Data Bits: 8  
Stop Bits: 1  
Parity: NONE

## Timing information (unless specified otherwise by a command's description)

Min character to character time: 0ms  
Min line to line time: 0ms  
Min time between commands: 0ms  
Max time to respond to a request: 100ms

---

# Command Syntax

---

## Command Syntax

### Command Syntax

The switch serial command set uses an ASCII based protocol and a terminal emulator can be used to test the serial port of the switch. Here are some examples:

```
^SZ @1,4$      = Set audio zone 1 to source 4
^VPZ @8,+2$    = Turn up the volume 2% on zone 8
^SZ @1:32,3$   = Set analog audio zones 1-32 to source 3
^BAZ @16:32,-2 = Lower the bass on zones 16-32 by 1dB
```

Each serial command is formatted as:

```
^CMD param1,param2,...$
```

Where:

```
^      = All commands and responses start with the '^' character.
CMD    = The name of the command.
param  = Any number of parameters can follow a command.
$      = All commands and responses end with the '$' character.
```

For instance the name of the command to turn power on / off is 'P' (must be capitalized) therefore, to turn on the switch send:

```
^P 1$          -> Command sent to the audio switch
^+$           <- Acknowledgment indicating valid command
^=P 1$        <- Response from the audio switch for new setting
```

to turn off the switch send:

```
^P 0$          -> Command sent to the audio switch
^+$           <- Acknowledgment indicating valid command
^=P 0$        <- Response from the audio switch for new setting
```

**NOTE:** Only the characters between '^' and '\$' are valid, any characters sent before the '^' or after the '\$' will be ignored.

**NOTE:** Space characters are optional. For instance "**^P 1\$**" and "**^P1\$**" are the same command.

**NOTE:** Carriage Returns and Line Feeds are allowed, and are ignored.

**NOTE:** By default, the switch adds a carriage return and a line feed to the end of its responses, after the '\$'. This makes testing with terminal software easier. Since they are outside the '^' and '\$' characters, they should be ignored by software drivers. If desired, this behavior can be disabled. (See: "**'XS' Control Settings**" on page 22)

## Command Responses

### Type of Responses and Timing Information

There are three different types of responses: Acknowledgements, Errors and Query Strings.

By default, the switch will always respond to a command, there are no "time-out" modes, if you send a command and don't get a response within 100ms, something's wrong with the connection.

---

# Command Syntax

---

## The Acknowledgement Response

Every command will be followed by an acknowledgement or error response.

Anytime you issue a command and there are no errors, you will receive the acknowledgement response. Which is:

`^+ $`

This can be turned off using the “**XS’ Control Settings**” on page 22.

When the Acknowledgement Response turned off, then there will be commands that have no response at all, and the “No time-out modes” will no longer apply.

For instance the `^SS 512$` command will not respond if there are no errors.

## The Error Response

Every command will be followed by an acknowledgement or error response.

If something is wrong with the command, you will get an error response. Which is

`^!<error_number>$`

which is the ‘!’ followed by an error number (in ASCII), followed the ‘\$’ character.

For instance ‘9’ is not allowed as a parameter in the ‘P’ (power) command, so:

`^P 9$`

-> Command sent to audio switch

`^!2$`

<- Error response to an out of range parameter

which indicates there was an out of range parameter.

The following are the Error Response codes that can be returned by the switch:

- 1 - Unrecognized command.
- 2 - A parameter was out of range.
- 3 - Syntax error, or a badly formed command.
- 4 - Reserved.
- 5 - Too many or too few parameters.
- 6 - Device busy, cannot process command.
- 7 - Buffer overflow.
- 8 - The command is not valid if device is not powered on.
- 1xx Error initializing the Audio subsystem.

And some more detailed descriptions of their meanings:

**Error 1** - The command given was not recognized as a switch command. Commands are case sensitive.

**Error 2** - One of the parameters given was too large, or too small, the command will be ignored.

**Error 3** - Something was wrong with the command's syntax. There was possibly extra data at the end of the line, or non-decimal data as part of a parameter.

**Error 4** - Reserved.

**Error 5** - The number of parameters given does not match the number allowed by this command.

**Error 6** - Currently this error is not returned, but could be at some future time.

**Error 7** - An internal buffer has overflowed.

**Error 8** - Power to the device must be ‘ON’ before this command is allowed. When powered off, the audio board sets are fully powered down and can not respond to queries. In some cases an

---

# Command Syntax

---

“^!8\$” response will be returned for a command that cannot function in the powered down mode.

**Error 1xx** - An error occurred while initializing the audio sub-section.

## The Query Response

The query response is sent by the switch to indicate a setting has changed, or as a response to a query command. The query response string consists of the ‘=’ character followed by the command string of the command being queried.

For instance, in the case of the power command:

<code>^P?\$</code>	-> Send a power request command to the audio switch
<code>^+\$</code>	<- Acknowledgement (the command has no errors)
<code>^=P 1\$</code>	<- Query response indicating the power is on.

The response commands will always have a space following the command name. Parameters of the response strings will always be of a fixed size for easier parsing. Leading zeros are used to enforce the fixed size and positions of response strings.

For instance the response string of the “^SZ @1?\$” can be:

```
^=SZ @001,003$
```

The zone and source parameters of this command will always be 3 digits long. This allows easier parsing in some control systems.

## Reference for Basic Control Commands

### Definitions

The following terms are used through out this manual.

#### *Zone*

An output. Zone and Output are used interchangeably throughout the manual. For most commands, zones are indicated by using a ‘@’ prefix character. You can specify a range of zones by separating the range with the ‘:’ character.

#### *Source*

An input. Source and Inputs are used interchangeably throughout the manual. For commands that mix sources with zones, ‘sources’ are indicated by numbers without a ‘@’ prefix.

## Using Bitmapped Parameters

### Reading / Writing Bitmapped Parameters

Some commands accept “Bitmapped” parameters. These are decimal values that represent a series of flags, or bits, that control, enable and/or disable different device operations.

Binary arithmetic is used to represent bitmapped parameters, it is assumed the reader has some familiarity with binary arithmetic.

An example of a command that uses a bitmapped parameter is the “XS settings” command, which is defined as:

```
^XS settings$
```

# Command Syntax

---

Where ‘settings’ is a bitmapped parameter defined as:

Value	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Bit Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AMU	VMU	12V	AON	IRJ	IRS	IRE	KYE	CHG	SET	CSE	CRE	CHM	ECO	ACK	ASY
Default:	1	0	0	1	1	1	0	0	1	1	0	1	0	1	1	1

For information on what each bit of the XS command does, see: “XS’ Control Settings” on page 22.

The “Value” row, in the table’s header, refers to the values, that when added together, create the decimal parameter used by the command. For instances if you want the bits ‘ASY’ and ‘IRS’ to be set to 1, and the rest of the bits set to zero, the parameter’s value would be calculated as: 1+1024 (ASY’s value = 1, IRS’s value = 1024), making the parameter value: 1025.

The command to set those two bits to ones, and reset all the others would be:

```
^XS 1025$
```

Individual bits of a bitmapped parameter can be set or reset without affecting the other bits, by prefixing the bitmapped parameter with a ‘+’ to set individual bits, or a ‘-’ to reset individual bits.

For instance in the above example the bitmapped value has been set to 1025. If we would now like to enable the IR jack, by setting the ‘IRJ’ bit, the following command can be issued:

```
^XS +2048$
```

This will set the ‘IRJ’ bit, and have no effect on the others, and the new “XS” value would be: 3073

If we’d like to now disable the IR jack and the IR sensor, by clearing the ‘IRJ’ and ‘IRS’ bits, we’d use the value “2048+1024”, or 3072 and issue the command:

```
^XS -3072$
```

leaving the new “XS” value to be: 1.



# Input / Output Numbering

---

## Input / Output Numbering

### **The Default (Native) Source Numbering**

This document describes the protocol for our full line of audio switches.

By default, the source mapping of each switch is model dependent. Each switch will have its sources numbered based on the number of analog, coax, optical, etc. sources. This requires each driver to remap its sources based on the switch's model. There are many drivers on the market that have been written for our switches, using the native mappings. In an effort to keep from breaking old drivers, the switches still support the native mappings, by default, when shipped from the factory.

To help with writing newer, more universal drivers, we have added an Extended I/O mode. In this mode all of our current line of switches will use the same source numbers. We recommend supporting the Extended I/O mode when writing new drivers. It removes the need to have different I/O mappings for each switch. See: “**The Extended I/O mode**” on page 12.

This manual's examples assume the switch is set the Extended I/O mode. All the commands are identical between the Extended I/O mode, and the native I/O mode, only the source numbers change.

The following are the native I/O source numbers of each audio based switch.

---

# Input / Output Numbering

---

## The ProAudio16 native source numbering

In the native I/O mode, the sources for the analog audio zones are:

- 0 = Disconnect zone from any source.
- 1-16 = Analog audio from RCA left / right inputs.
- 17-32 = PCM-stereo audio from Coax inputs (mutes if Dolby5.1 or DTS).
- 33-48 = PCM-stereo audio from Optical inputs (mutes if Dolby5.1 or DTS).

The sources for the digital audio zones are:

- 0 = Disconnect zone from any source.
- 1-16 = Reserved. Mutes audio, same as switching source '0'.
- 17-32 = Digital audio from Coax inputs.
- 33-48 = Digital audio from Optical inputs.
- 49-64 = PCM-stereo that mirrors the analog outputs.

*Note 1: For a complete description of inputs 49-64 see: "PCM-stereo mirroring analog outputs" on page 12.*

*Note 2: Only PCM-stereo type audio can be down-converted to analog audio. You cannot down-convert Dolby5.1 or DTS.*

## The ProAudio1632 native source numbering

In the native I/O mode, the sources for the analog audio zones are:

- 0 = Disconnect zone from any source.
- 1-16 = Analog audio from RCA left / right inputs.
- 17-32 = PCM-stereo audio from Coax inputs (mutes if Dolby5.1 or DTS).
- 33-40 = PCM-stereo audio from Optical inputs (mutes if Dolby5.1 or DTS).

The sources for the digital audio zones are:

- 0 = Disconnect zone from any source.
- 1-16 = Reserved. Mutes audio, same as switching source '0'.
- 17-32 = Digital audio from Coax inputs.
- 33-40 = Digital audio from Optical inputs.
- 41-56 = PCM-stereo that mirrors the analog outputs.

*Note 1: For a complete description of inputs 49-64 see: "PCM-stereo mirroring analog outputs" on page 12.*

*Note 2: Only PCM-stereo type audio can be down-converted to analog audio. You cannot down-convert Dolby5.1 or DTS.*

# Input / Output Numbering

---

## **The ProAudio32, ProAudio48 & ProAudio64 native source numbering**

In the native I/O mode, the sources for the analog audio zones are:

- 0 = Disconnect zone from any source.
- 1-32 = Analog audio from RCA left / right inputs.
- 33-64 = PCM-stereo audio from Coax inputs (mutes if Dolby5.1 or DTS).
- 65-80 = PCM-stereo audio from Optical inputs (mutes if Dolby5.1 or DTS).

The sources for the digital audio zones are:

- 0 = Disconnect zone from any source.
- 1-32 = Reserved. Mutes audio, same as switching source '0'.
- 33-64 = Digital audio from Coax inputs.
- 65-80 = Digital audio from Optical inputs.
- 81-96 = PCM-stereo that mirrors the 16 analog outputs of the same audio bank.

*Note 1: For a complete description of inputs 81-96 see: “PCM-stereo mirroring analog outputs” on page 12.*

*Note 2: Only PCM-stereo type audio can be down-converted to analog audio. You cannot down-convert Dolby5.1 or DTS.*

---

# Input / Output Numbering

---

## The Extended I/O mode

This manual assumes all switches have been set to the Extended I/O mode.

This mode sets all of our switches to use the same source numbering. This allows writing a single driver that can be used with multiple switches.

In the Extended I/O mode, all switches use the same mappings for the sources. Setting a zone to a source that doesn't exist, (like setting a zone to analog input 17, on a switch with only 16 analog inputs) will simply mute the zone -- it's the same as setting the zone to 0.

To use the Extended I/O sources you have set the XIO flag in the 'XS' command to one (1) by sending the following commands:

```
^XS +32768$      <- Set the XIO flag to use the extended I/O mappings
^SS 8$          <- Save all 'XS' settings in EEPROM (optional)
```

The 'SS' command will save the settings in EEPROM so that the Extended I/O setting is saved through a power cycle. Otherwise you will need to send the 'XS' command each time you initialize the switch after applying power. (See: *"'XS' Control Settings"* on page 22, and *"'SS' Save Default Power On Settings"* on page 24.)

In the Extended I/O mode, the sources for the analog audio zones are (see *"'SZ' Set Zone(s) -- Analog Audio Switch"* on page 14):

```
0           = Disconnect zone from any source.
1-32       = Analog audio from RCA left / right inputs.
33-64      = PCM-stereo audio from Coax inputs (mutes if Dolby5.1 or DTS).
65-80      = PCM-stereo audio from Optical inputs (mutes if Dolby5.1 or DTS).
```

The sources for the digital audio zones are (see *"'DSZ' Set Zone(s) -- Digital Audio Switch"* on page 15):

```
0           = Disconnect zone from any source.
1-32       = Reserved. Mutes audio, same as switching source '0'.
33-64      = PCM-stereo, Dolby5.1 or DTS from Coax inputs.
65-80      = PCM-stereo, Dolby5.1 or DTS from Optical inputs.
129-144    = PCM-stereo that mirrors the 16 analog outputs of the same audio card.
```

*Note: Only PCM-stereo type audio can be down-converted to analog audio. You cannot down-convert Dolby5.1 or DTS.*

### PCM-stereo mirroring analog outputs

The inputs 129-144 allow the digital outputs to mirror those of the analog outputs.

```
129 = Mirror analog output 1.
130 = Mirror analog output 2, etc.
```

For instance if you want the coax output 3 to mirror anything coming out of analog output 1, you would send the command:

```
^DSZ @3,129$
```

After this command, anything playing on analog output 1 will be mirrored to coax output 3. Any input changes made to analog output 1, will also affect coax output 3. Any volume, tone, etc. changes made to analog output 1, will also affect coax output 3. Coax output 3 becomes an exact digital copy of what is coming out of analog 1.

---

## Input / Output Numbering

---

This mirror only works locally on each audio bank. For instance the ProAudio48 has three output audio banks. Bank one contains the analog and digital outputs 1-16, bank two contains outputs 17-32, and bank 3 contains outputs 33-48.

In this case, coax outputs 1-16 are mapped as:

129 = Mirror analog output 1.  
130 = Mirror analog output 2, etc.

Coax outputs 17-32 are mapped as:

129 = Mirror analog output 17.  
130 = Mirror analog output 18, etc.

Coax outputs 33-48 are mapped as:

129 = Mirror analog output 33.  
130 = Mirror analog output 34, etc.

On the ProAudio48, the above example:

```
^DSZ @3,129$
```

would still mirror analog output 1 on coax output 3, but the command:

```
^DSZ @19,129$
```

would mirror analog output 17 on coax output 19, and the command:

```
^DSZ @35,129$
```

### Example ProAudio16

Assuming you have one of our Proaudio16 16x16 audio switch, setting the Extended I/O mode will change the source numbers as follows:

0	= Disconnect audio from any source (mutes audio).
1-16	= Select analog inputs 1-16.
17-32	= These will mute audio. There are only 16 analog inputs on the Proaudio16
33-48	= Select coax inputs labeled 17-32 on the back of the switch.
49-64	= These will mute audio. There are only 16 coax inputs on the Proaudio16.
65-80	= Select optical inputs labeled 33-48 on the back of the switch.

All the other inputs will mute since they don't exist on the Proaudio16.

By using these values in your driver, you can easily substitute a Proaudio32 switch. The source numbers do not change in the Extended I/O mode between switches.

---

# Power, Switching, and Muting

---

## Power, Switching, and Muting

### Basic Control Commands

These commands are all that are needed for basic control of the switch and includes power on/off, remapping sources to zones.

---

#### 'P' Power Control

Turn on / off, or toggle the power state of the switch:

<code>^P 0\$</code>	Turn off power if power is not locked on.
<code>^P 1\$</code>	Turn on power.
<code>^P 2\$</code>	Turn off power regardless of "locked" state.
<code>^P 3\$</code>	Turn on power and lock it on -- only a "P 2" command can turn off power.
<code>^P +\$</code>	Toggle power
<code>^P ?\$</code>	Query for current setting

Response String:

`^=P n$`

Where:

n = Current power status, 0=Off, 1=On, 3=Locked On. (A value of '2' is never returned.)

The power lock and unlock can be used to override the a controller's on/off logic, keeping a controller from turning off the switch.

Drivers should be written to only use the "P 0" and "P 1" commands to turn on and off the switch. This allows users to override the driver's logic by manually issuing the "P 2" and "P 3" commands.

By issuing a "P 3" command (lock power on), the switch will be turned on, and locked. Only a subsequent "P 2" command will turn off the switch, The "P 0" and "P 1" commands, issued by the controller, will be ignored.

---

#### 'SZ' Set Zone(s) -- Analog Audio Switch

This command is used to map sources to any number of analog zones, on audio switch.

The zones for the analog audio switch are:

1-128 = Depending upon model number.

The source numbers for the switch, depend upon the switch's model number, and the Extended I/O setting.

For a full listing of source numbers see: "**The Default (Native) Source Numbering**" on page 9, and "**The Extended I/O mode**" on page 12.

The different forms of the 'SZ' command are:

<code>^SZ @zone, source\$</code>	Map source, to a zone.
<code>^SZ @zone, +\$</code>	Sequence zones forward through sources.
<code>^SZ @zone, -\$</code>	Sequence zones in reverse through sources.
<code>^SZ @zone?\$</code>	Read current settings of given zones.

Response Strings:

`^=SZ @zone, source$` or,

---

# Power, Switching, and Muting

---

Where:

@zone = One (or more) zones to be mapped.  
source = Source to map to given zone(s).

## 'SZ' Examples

The 'SZ' command in its simplest form:

```
^SZ @1,2$
```

maps the source 2 to the zone 1, to source 2. You can also map multiple zones using a single command. For instance:

```
^SZ @1,@3,@5,7$
```

maps the source 7 to zones 1, 3, and 5. The commas between the zones are optional:

```
^SZ @1@3@5,7$
```

is the same as the previous command.

You can also use the ':' to define a range of zones:

```
^SZ @3:12,5$
```

this will set zone 3 through 12 to source 5.

## 'SZ' Query Examples

To make parsing the response strings easier, only one response string is sent per zone. The response string is sent as a fixed length string using leading zeroes.

You can request the mapping of multiple zones, with one command, and still only one response string per zone will be returned, for instance:

```
^SZ @1@3@4?
```

could return:

^+\$	<- Indicates no errors in the command
^=SZ @001,002\$	<- Source 2 mapped to zone 1
^=SZ @003,006\$	<- Source 6 mapped to zone 3
^=SZ @004,005\$	<- Source 5 mapped to zone 4

---

## 'DSZ' Set Zone(s) -- Digital Audio Switch

This is the command used to map sources to any number of zones on the digital audio zones.

This command is only in effect if the 'DRZ' command for this zone is set to '0'. (See "**'DRZ' Digital audio Routing**" on page 41.)

The digital audio switch cannot convert analog audio to digital. If you route an analog input to a digital output, the output will simply mute. You can convert the output of an analog zone to PCM-stereo and send it out the coax connection, using the 'DRZ' command. Or by switching directly to an Analog output by using source numbers 129-145. You can only mirror the analog outputs that are in the same block as the coax outputs. For instance coax outputs 1-16 can only mirror analog outputs 1-16, and coax outputs 18-32 can only mirror analog outputs 18-32.

The zones for the digital audio switch are:

1-128 = Depending upon model.

---

## Power, Switching, and Muting

---

The source numbers for the switch, depend upon the switch's model number, and the Extended I/O setting.

For a full listing of source numbers see: “**The Default (Native) Source Numbering**” on page 9, and “**The Extended I/O mode**” on page 12.

The different forms of the ‘DSZ’ command are:

<code>^DSZ @zone,source\$</code>	Map a source, to a zone.
<code>^DSZ @zone,+\$</code>	Sequence zones forward through all digital sources.
<code>^DSZ @zone,-\$</code>	Sequence zones in reverse through all digital sources.
<code>^DSZ @zone?\$</code>	Read current settings of given zones.

Response Strings:

`^=DSZ @zone,source$`

Where:

@zone = One (or more) zones to be mapped.  
source = Source to map to given zone(s).

### ‘DSZ’ Examples

The ‘DSZ’ command in its simplest form:

`^DSZ @1,9$`

maps the source 9 to the zone 1, to source 2. You can also map multiple zones using a single command. For instance:

`^DSZ @1,@3,@4,11$`

maps the source 11 to zones 1, 3, and 4. The commas between the zones are optional:

`^DSZ @1@3@5,11$`

is the same as the previous command.

You can also use the ‘:’ to define a range of zones:

`^DSZ @3:12,5$`

this will set zone 3 through 12 to source 5.

### ‘DSZ’ Query Examples

To make parsing the response strings easier, only one response string is sent per zone. The response string is sent as a fixed length string using leading zeroes.

You can request the mapping of multiple zones, with one command, and still only one response string per zone will be returned, for instance:

`^DSZ @1@3@4?`

could return:

<code>^+\$</code>	<- Indicates no errors in the command
<code>^=DSZ @001,012\$</code>	<- Source 12 mapped to zone 1
<code>^=DSZ @003,016\$</code>	<- Source 16 mapped to zone 3
<code>^=DSZ @004,015\$</code>	<- Source 15 mapped to zone 4’

---

### ‘MZ’ Mute Zone(s) -- Analog Audio Switch



---

## Power, Switching, and Muting

---

This is the command used to hard mute, or disconnect an audio zone. When audio is disconnected, the sound is immediately switched off.

This command should *not* be used to simply mute audio on a zone. This command is nearly equivalent to setting a zone to source 0. It disconnects the zone from anything. The only difference between a “`^MZ @1,1$`” command and a “`^SZ @1,0$`”, is that the MZ command remembers its source connection. When using the MZ command you can “reconnect” to the current source. When using an “`^SZ @1,0$`” command to disconnect a zone, the controller will have to keep track of a source if a “reconnect” command is required.

The different forms of the MZ command are:

<code>^MZ @zone,mute\$</code>	Mute (disconnect) all listed zones.
<code>^MZ @zone,+\$</code>	Toggle the mute settings of zones.
<code>^MZ @zone?\$</code>	Read current settings of given zones.

Response Strings:

`^=MZ @zone,mute$`

Where:

`@zone` = One (or more) zones to be mapped.  
`mute` = Mute setting (0=Unmuted, 1=Muted).

### 'MZ' Examples

The 'MZ' command:

`^MZ @1,@3,1$`

mute zones '1' and '3'. You can also mute/unmute multiple zones using a single command. For instance:

`^MZ @1,1,@3,@4,0$`

mutes zone '1', and unmutes zones '3' and '4'.

### 'MZ' Query Examples

The query response string:

`^MZ @1,?$` >- Query request sent to the audio switch

would respond with:

`^+$` <- Indicates no errors in the command  
`^=MZ @1,1` <- Audio muted on zone '1'

To make parsing the response strings easier, only one response string is sent per zone. The response string is sent as a fixed length string using leading zeroes.

You can request the settings of multiple zones, with one command, and only one response string per zone will be returned, for instance:

`^MZ @1,@3,@4,?`

could return:

`^+$` <- Indicates no errors in the command  
`^=MZ @001,1$` <- Zone '1' is muted  
`^=MZ @003,0$` <- Zone '3' is not muted

---

# Power, Switching, and Muting

---

`^=MZ @004,0$`

`<- Zone '4' is not muted`

---

## 'PZ' Power Down (Disconnect) Zone(s) -- Analog Audio Switch

This is the command used to hard mute, or disconnect, and power down an audio zone. When audio is disconnected, the sound is immediately switched off.

This command should *not* be used to simply mute audio on a zone. This command is nearly equivalent to the MZ command. The only difference between a "`^PZ @1,0$`" command and an "`^MZ @1,1$`", is that the PZ command indicates that the zone is powered down. Powering down can also turn off a 12V output, if the 12V trigger output is mapped to the zone, and no other zones mapped to the same 12V trigger are turned on.

The different forms of the PZ command are:

<code>^PZ @zone,pwr\$</code>	Power on/off a zone or zones.
<code>^PZ @zone,+\$</code>	Toggle the settings of zones.
<code>^PZ @zone?\$</code>	Read current settings of given zones.

Response Strings:

`^=PZ @zone,pwr$`

Where:

`@zone` = One (or more) zones to be mapped.  
`pwr` = Power setting (0=Off, 1=Powered On).

## 'PZ' Examples

The 'PZ' command:

`^PZ @1,@3,1$`

Turns on the zones '1' and '3'. You can also power on/off multiple zones using a single command. For instance:

`^PZ @1,1,@3,@4,0$`

powers on zone '1', and powers off zones '3' and '4'.

## 'PZ' Query Examples

The query response string:

`^PZ @1,?$` >- Query request sent to the audio switch

would respond with:

`^+$` <- Indicates no errors in the command  
`^=PZ @1,1` <- Zone '3' is powered on

To make parsing the response strings easier, only one response string is sent per zone. The response string is sent as a fixed length string using leading zeroes.

You can request the settings of multiple zones, with one command, and only one response string per zone will be returned, for instance:

`^PZ @1,@3,@4,?`

could return:

`^+$` <- Indicates no errors in the command

---

## Power, Switching, and Muting

---

<code>^=PZ @001,1\$</code>	<- Zone '1' is powered on
<code>^=PZ @003,0\$</code>	<- Zone '3' is powered off
<code>^=PZ @004,0\$</code>	<- Zone '4' is powered off

---

### 'DMZ' Mute (Disconnect / Power down) Zone(s) -- Digital Audio Switch

This is the command used to hard mute, or disconnect, or power down a digital audio zone. When digital audio is disconnected, the sound is immediately switched off.

This command is nearly equivalent to setting a zone to source 0. It disconnects the zone from anything. The only difference between a "`^DMZ @1,1$`" command and a "`^DSZ @1,0$`", is that the 'DMZ' command remembers its source connection. When using the 'DMZ' command you can "reconnect" to the current source. When using an "`^DSZ @1,0$`" command to disconnect a zone, the controller will have to keep track of a source if a "reconnect" command is required.

The different forms of the 'DMZ' command are:

<code>^DMZ @zone,mute\$</code>	Mute (disconnect) a zone.
<code>^DMZ @zone,+\$</code>	Toggle the settings of zones.
<code>^DMZ @zone?\$</code>	Read current settings of given zones.

Response Strings:

`^=DMZ @zone,mute$`

Where:

`@zone` = One (or more) zones to be mapped.  
`mute` = Mute setting (0=Unmuted, 1=Muted).

### 'DMZ' Examples

The 'DMZ' command in its simplest form:

`^DMZ @1,@3,1$`

mutes zones '1' and '3'. You can also mute/unmute multiple zones using a single command. For instance:

`^DMZ @1,1,@3,@4,0$`

mutes zone '1', and unmutes zones '3' and '4'.

### 'DMZ' Query Examples

If digital audio is muted on zone '1', then:

`^DMZ @1,?$` >- Query request sent to the audio switch

would respond with:

<code>^+\$</code>	<- Indicates no errors in the command
<code>^=DMZ @1,1\$</code>	<- Digital audio is muted on digital zone '3'

To make parsing the response strings easier, only one response string is sent per zone. The response string is sent as a fixed length string using leading zeroes.

You can request the settings of multiple zones, with one command, and only one response string per zone will be returned, for instance:

`^DMZ @1,@3,@4,?$`

## Power, Switching, and Muting

---

could return:

<code>^+\$</code>	<- Indicates no errors in the command
<code>^=DMZ @001,1\$</code>	<- Zone '1' muted
<code>^=DMZ @003,0\$</code>	<- Zone '3' unmuted

---

# Advanced Control

---

## Advanced Control

### Advanced Commands

These commands are for more advanced control over the switch, including front panel light intensities, changes to serial port behavior, etc.

### Advanced Command Definitions

---

#### 'V' Version

Returns the current firmware version of the switch.

`^V ?$`

Response String:

`^=V "ProAudio16",1.23a,59B2S12345678$`

Where:

`"ProAudio16"` = Model  
`1.23a` = Firmware version  
`59B2S12345678` = Serial number

---

#### 'DZ', 'DDZ', Set Zone Switching Delays

These commands are used to add a "mute time" when switching between analog or digital (coax) inputs.

The 'DZ' command controls the analog audio zones, and the 'DDZ' command controls the digital zones. Only the 'DZ' command is described here, but both commands function the same way.

With no delay in place, when the switch switches from one input to another, the switching is immediate. When a delay is added, an extra step takes place when switching between inputs.

Instead of immediately switching to the new input, the zone is first muted for the 'delay' amount of time, and is then switched to the new input.

For audio devices, the sound is first muted for the amount of time given by the 'DZ' command, and then switched to the new input. This can eliminate pops on some digital audio receivers.

The different command forms are:

`^DZ @zone, delay$`                      Add a delay to a zone.  
`^DZ @zone?$`                              Read current settings of given zones.

Response Strings:

`^=DZ @zone, delay$`                      or,

Where:

`@zone` = One (or more) zones to be mapped.  
`delay` = The delay time in milliseconds (1000 milliseconds = 1 second).

#### 'DZ' Examples

For examples on using the 'DZ' and how to use breakaway options, see the 'SZ' command.

---

# Advanced Control

## 'XS' Control Settings

Turn on and off operational modes of the switch.

The format of the command is:

<code>^XS settings1,settings2\$</code>	Set the control bits to 'settings1' and 'settings2'
<code>^XS +settings1,+settings2\$</code>	Set bits indicated in 'settings1' and 'settings2' to 1
<code>^XS -settings1,+settings2\$</code>	Reset bits indicated in 'settings1' and 'settings2' to 0
<code>^XS ?\$</code>	Query for current settings

Response String:

`^=XS settings1,settings2$`

Where 'settings1' is a bitmapped parameter defined as:

Value	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Bit Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XIO	0	0	0	0	0	0	0	0	0	0	CRE	0	ECO	ACK	ASY
Default:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

ASY - 0=Only send a response string when requested	1=Send a response string on any parameter change
ACK - 0=Don't acknowledge cmds with "^+\$"	1=Acknowledge error free commands with a "^+\$"
ECO - 0=Do not send a response strings for each cmd	1=Always send response string when a serial command is issued
CRE - 0=Don't send CRs/LFs at end of responses	1=End all responses with a carriage return and a line feed
XIO - 0=Use standard numbering of I/O	1=Use the extended I/O numbering

All other bits are reserved and must be set to 0

Where 'settings2' is a bitmapped parameter defined as:

Value	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Bit Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	0	0	0	0	0	0	0	0	UVL	MJP	0
Default:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

MJP - 0=Volume increments from mute level	1=Volume jumps to unmute level before incrementing
UVL - 0='VP', 'VPZ' commands do not unmute vol	1=Volume unmutes on 'VP' or 'VPZ' command

All other bits are reserved and must be set to 0

This command uses a bitmapped parameter. Each bit can set or reset without affecting the other bits. (See: "**Using Bitmapped Parameters**" on page 7, for more information on using bitmapped parameters.)

Both parameters do not have to be present on the command line, if a parameter is left out the command, it will be left unchanged:

`^XS settings1$` Only 'settings1' is changed

Since 'settings2' was not present on the command line, it will not be affected.

A comma can be used to indicate a missing parameter:

`^XS ,settings2$` Only 'settings2' is changed

Since 'settings1' was not present on the command line, it will not be affected.

The following paragraphs define each option in more detail:

---

# Advanced Control

---

## *'ASY' Send commands asynchronously any time a parameter changes*

With this bit set to a '1', anytime a parameter changes, a response string is sent. For instance any changes to the front panel will cause a response string to be sent to the serial ports, and any open TCP/IP socket.

With this bit set to a '0', response strings are only when requested, using the '?' as a parameter.

For one way serial control, this bit, and the 'ECO' bit should probably be set to '0'. For faster responses times these two bit can also be set to '0'. If a lot of commands are being sent, the response time of the switch can slow down waiting for all the response strings to be sent.

## *'ACK' Enable / Disable Acknowledgements*

Each time a command is sent to the switch, the switch responds with either an error message, if there was a problem with the command, or an acknowledgement string of "^+ $\$$ ". This behavior can be changed by setting this bit to '0'. If the bit is reset, the "^+ $\$$ " string will not be sent.

## *'ECO' Enable / Disable the 'Parameter Changed' Strings*

Each command that makes a change to a parameter, will be echoed with a response string that indicates the new values of the parameters changed. The response strings are only issued after the changes have taken affect, and reflect the new state of the switch.

With this bit set to '0', these automatic response strings will not be sent. However if the new parameter is different than the current parameter, and the 'ASY' bit is set, then a response string will still be sent to indicate a parameter has changed. This bit forces a response string to be sent even if the parameter of the command being sent is the same as the current value of the switch.

To stop all response strings from being sent, set this bit the 'ASY bit to '0'.

## *'CRE' Enable / Disable trailing Carriage Returns Line Feeds*

A carriage return and line feed can be appended to all responses coming from the switch. This is useful when using terminal software to test command strings.

Since the carriage returns and line feeds are sent outside of the normal string (they are sent after the ending ' $\$$ '), they should be ignored by the controller. But if there are problems with this behavior, they can be turned off by setting this bit to a '0'.

## *'XIO' Enable / Disable Extended I/O numbering*

If this bit is reset (0), the normal, sequential number system is used for sources and zones.

If this bit is set (1), then an extended I/O number system is used that is the same between all switches of the same family.

By setting this bit to one (1) during the initialization process, a driver can be written that treats all switches the same. All analog sources will start with the same number. The same is true for coax and optical sources. This makes it much easier to write a driver for the full line of switches.

## *'MJP' Enable / Disable Mute Volume Jump when using Volume Increment / Decrement commands*

When this bit is set to '0' incrementing/decrementing the volume of a muted zone will cause the zone's volume to increment/decrement from the current muted level. This is the "no surprises" mode of operation. If one person mutes a zone and walks out of the room, someone else can walk into the room and pick up the volume control and start adjusting the volume with no surprises. The volume will simply start increasing from the current level.

When this bit is set to '1' incrementing the volume of a muted zone will cause the zone's volume to jump back to its unmuted level before incrementing the level. This mode is most useful when only one

## Advanced Control

person is in control of the volume. For instance if the phone rings, the mute button can be pressed. Once the call is over, pressing any volume button will cause the volume to jump back to it's previous setting. Since the same user pressed the mute button, he/she won't be surprised when the volume jumps back to its previous value.

The above behaviors are supported by the '+' and '-' options of the 'VZ' and 'VPZ' commands. Using the 'VZ' or 'VPZ' commands to set the volume directly (leaving out the '+' or '-' characters), will not be affected by this flag.

### 'UVL' Enable / Disable Volume Unmute when using direct level Volume Commands

If this bit is cleared, using the 'VZ' or 'VPZ' commands to set the volume directly (leaving out the '+' or '-' characters), will not affect the mute setting. This allows a muted zone to have its volume set to a predetermined value, and then unmuted.

Control systems that use sliding touch pads will usually precalculate a volume, and send the volume using the 'VZ' or 'VPZ' commands without the '+' or '-' characters. If the zone is muted, then the commands will have no affect on the volume, until the zone is manually unmuted.

Setting this bit causes the zone to unmute whenever a 'VZ' or 'VPZ' command is sent, regardless of whether it's an incremental command or an absolute volume setting.

If, for some reason, you need to send precalculated volume settings to the switch, without unmuting the zone's volume, the you will want to clear this bit.

This flag only affects the 'VZ' and 'VPZ' commands that set the volume directly ('VZ' and 'VPZ' commands that do not use the '+' and '-' prefixes).

### 'SS' Save Default Power On Settings

When all power is lost to the switch (for instance the power cord is unplugged), the current settings will be lost. When power is restored all settings will be set to their initial default values. This command allows you to change those initial values.

To use this command, the switch should have the parameters to be saved, set to the values that are to be used as the initial values.

Once the current settings are in place, this command is issued to save the current settings as the initial values.

This command uses a bitmapped parameter to indicate which values are to be saved. A bit set to '1' indicates the associated setting should be saved, a bit set to '0' indicates the setting should not be saved and the previously saved setting will remain unchanged.

Command formats:

^ss settings\$ Set the enable bits to 'settings'

Response String:

*There is no response string to the SS command, this command is write only. If the 'ACK' bit is set in the "XS" command, then a "^+ \$" acknowledgement string will be sent to indicate the commands was accepted.*

Where 'settings' is a bitmapped parameter defined as:

Value	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit-0 - Save zone source/output mappings, and power state



# Advanced Control

- Bit-1 - Save switch delay times set by the 'DZ' commands
- Bit-2 - \*Reserved\*
- Bit-3 - Save control settings (the 'XS' command settings)
- Bit-4 - \*Reserved\*
- Bit-5 - Save IP parameters and settings ('IPA', 'IPM', 'IPG', and 'IPSET' settings)
- Bit-6 - \*Reserved\*
- Bit-7 - \*Reserved\*
- Bit-8 - Save zones' volume settings
- Bit-9 - Save zones' audio parameters (bass, treble, eq, mute level, zone gains, zone lipsync, etc. all but volume)
- Bit-10 - Save source audio parameters (source gains, source lipsync delay)
- Bit-11 - Save global audio parameters (master volume, volume and mute fade times)
- Bit-12 - Save all paging parameters.

The 'SS' command is used to saving things like a static IP address, power on volume defaults, bass, treble, EQ settings, etc. into EEPROM. The values in EEPROM are used as the power on defaults when power is restored.

As an example, to change the power on volume settings from straight pass-thru (0dB gain), to muted, issue the following commands:

```
^VZ @1:8,0$           Set zones 1-8 to volume 0.
^SS 256$              Save volume levels as the new defaults.
```

Now when power is lost, the switch will power back on with all volume levels muted.

## 'FS' Reset to Factory Settings

This command will reset selected parameters of the switch to their factory default settings.

This command uses a bitmapped parameter to indicate which values are to be reset. A bit set to '1' indicates the associated setting should be reset, a bit set to '0' indicates the setting should not be reset and the previous setting will remain unchanged.

Command formats:

```
^FS settings$        Reset the parameters indicated by 'settings'
```

Response String:

*There is no response string to the SS command, this command is write only. If the 'ACK' bit is set in the "XS" command, then a "^+\$" acknowledgement string will be sent to indicate the commands was accepted.*

Where 'settings' is a bitmapped parameter defined as:

Value	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- Bit-0 - Reset zone source/output mappings, and power state
- Bit-1 - Reset switch timings (the 'DZ' command settings)
- Bit-2 - \*Reserved\*
- Bit-3 - Reset control settings (the 'XS' command settings)
- Bit-4 - \*Reserved\*
- Bit-5 - Reset IP parameters
- Bit-6 - \*Reserved\*
- Bit-7 - \*Reserved\*
- Bit-8 - Reset zones' volume settings
- Bit-9 - Reset zones' audio parameters (bass, treble, eq, mute level, zone gains, zone lipsync, etc. all but volume)
- Bit-10 - Reset source audio parameters (source gains, source lipsync delay)
- Bit-11 - Reset global audio parameters (master volume, volume and mute fade times)

# Advanced Control

---

Bit-12 - Reset paging parameters.

---

## 'LI' Lighting Mode and Intensities

Allows changing the behavior of the front panel backlight LEDs, and adjusting their intensities.

`^LI mode,dim,bright,off$`                      Set mode and intensities  
`^LI ?$`                                              Query for current settings

Response String:

`^=LI mode,dim,bright,off$`                      Set mode and intensities

Where:

`mode`                      = Lighting mode (Settings: 0=Off, 1=Dim, 2=Bright, 3=Auto-dim)  
`dim`                        = Intensity of dimmed LEDs (Range: 0-100)  
`bright`                    = Intensity of brightened LEDs (Range: 0-100)  
`off`                        = Intensity of the standby power LED (Range: 0-100)

Response Example:

`^=LI 3,020,090,010$`                              Mode=Auto, dim=20%, bright=90%, off=10%

The mode settings allow you to set the front panel LEDs to always be: 0 - off, 1 - Always at the 'dim' intensity, 2 - always at the 'bright' intensity, or 3 - auto-dim from 'bright' to 'dim' when the front panel is not in use.

The intensities range for 0, which is off, to 100, which is full intensity.

When setting parameters, not all parameter have to be present, if a parameter is not present, it will be left unchanged. For instance to change just the 'bright' value to 50%:

`^LI ,,50$`                                              -> Set the 'bright' intensity to 50%

Only the 'bright' intensity is affected.

---

# Advanced Control

---

## TCP/IP Control

The switch can use DHCP to negotiate an address with a router, or it can be programmed to use a static IP address.

The 'IPSET' command is used to switch between the two modes. 'IPSET 0' indicates a static address, and 'IPSET 1' indicates a DHCP negotiated address.

To set a new static IP address, first issue the 'IPA', 'IPM' and 'IPG', commands to setup the IP address, IP mask, and IP gateway, respectively. Then issue the 'IPSET 0' command.

---

### 'IPSET' Set the IP operation mode, DHCP or STATIC

This command is used to set the TCP/IP operating mode of the switch. It allows you to choose between DHCP and Static IP address operations.

When set to the DHCP, the switch will negotiate an IP address, IP mask, and gateway address (the address of the router), using DHCP.

When set to Static mode, the IP address, IP mask, and gateway address, given by the 'IPA', 'IPM' and 'IPG' commands will be used.

You can easily switch between the two modes without losing the static address setting. In other words when switching to DHCP a new IP address values will be obtained from the router, but the static IP address values of the 'IPA', 'IPM' and 'IPG' commands are not lost, and when returning to static mode, these settings do not need to be resent.

The command format is:

<code>^IPSET mode\$</code>	Set new TCP/IP operating mode
<code>^IPSET ?\$</code>	Read the current TCP/IP operating mode

Response String:

`^=IPSET mode$`

Where:

mode = Current operating mode. 0=Static, 1=DHCP.

To change the static IP address the 'IPSET 0' form of this command must be issued. This is true even if you are already operating in the static mode. For instance if your static address is 192.168.1.200 and you wish to change it to 192.168.1.201, you must first issue a 'IPA' command and then reissue the 'IPSET 0' command:

<code>^IPA 192,168,1,201\$</code>	Set the new static address
<code>^IPSET 0\$</code>	Tell the <b>switch</b> to use the new static address

---

---

## Advanced Control

---

### 'IPA' Set / View the static IP Address

This command does not immediately change the TCP/IP address. It sets up the static IP address that will be used the next time the 'IPSET 0' command is issued.

The command format is:

<code>^IPA xxx,xxx,xxx,xxx\$</code>	Set new TCP/IP address
<code>^IPA ?\$</code>	Read the current TCP/IP address

Response String:

`^=IPA xxx,xxx,xxx,xxx$`

Where:

`xxx,xxx,xxx,xxx` = IP address in standard form (Ex: 192,168,001,200)

The response string of the 'IPA' command is not the current IP address of the switch, but the IP address that will be used the next time the 'IPSET 0' command is issued. To retrieve the current IP address of the switch, use the read only command 'IPAX' (See: "**'IPAX' Retrieve the current IP address in use**" on page 29.)

---

### 'IPM' Set / View the static IP Mask

This command does not immediately change the TCP/IP mask. It sets up the static mask that will be used the next time the 'IPSET 0' command is issued.

The command format is:

<code>^IPM xxx,xxx,xxx,xxx\$</code>	Set new TCP/IP mask
<code>^IPM ?\$</code>	Read the current TCP/IP mask

Response String:

`^=IPM xxx,xxx,xxx,xxx$`

Where:

`xxx,xxx,xxx,xxx` = IP mask in standard form (Ex: 255,255,255,000)

The response string of the 'IPM' command is not the current IP mask of the switch, but the IP mask that will be used the next time the 'IPSET 0' command is issued. To retrieve the current IP mask of the switch, use the read only command 'IPMX' (See: "**'IPMX' Retrieve the current IP mask in use**" on page 29.)

---

### 'IPG' Set / View the static IP gateway address

This command does not immediately change the TCP/IP gateway. It sets up the TCP/IP gateway address that will be used the next time the 'IPSET 0' command is issued.

The command format is:

<code>^IPG xxx,xxx,xxx,xxx\$</code>	Set new TCP/IP router address
<code>^IPG ?\$</code>	Read the current TCP/IP router address

Response String:

`^=IPG xxx,xxx,xxx,xxx$`

Where:

`xxx,xxx,xxx,xxx` = Gateway address in standard form (Ex: 192,168,001,001)

---

---

## Advanced Control

---

The response string of the 'IPG' command is not the current gateway address of the switch, but the gateway address that will be used the next time the 'IPSET 0' command is issued. To retrieve the current IP address of the switch, use the read only command 'IPGX' (See: "**IPGX** Retrieve the current IP gateway address in use" on page 29.)

---

### 'IPAX' Retrieve the current IP address in use

This command is read only and will return the IP address being used by the switch. If DHCP is being used, this will return the address given to the switch by the router. If static mode is being used, this will return the static address currently in use by the switch.

The command format is:

**^IPAX ?\$** Read the current IP address

Response String:

**^=IPAX xxx,xxx,xxx,xxx\$**

Where:

**xxx,xxx,xxx,xxx** = IP address in standard form (Ex: 192,168,001,200)

---

### 'IPMX' Retrieve the current IP mask in use

This command is read only and will return the IP mask being used by the switch. If DHCP is being used, this will return the mask given to the switch by the router. If static mode is being used, this will return the static mask currently in use by the switch.

The command format is:

**^IPMX ?\$** Read the current IP mask

Response String:

**^=IPMX xxx,xxx,xxx,xxx\$**

Where:

**xxx,xxx,xxx,xxx** = IP mask in standard form (Ex: 255,255,255,000)

---

### 'IPGX' Retrieve the current IP gateway address in use

This command is read only and will the gateway address being used by the switch. If DHCP is being used, this will return the address given to the switch by the router. If static mode is being used, this will return the static address currently in use by the switch.

The command format is:

**^IPGX ?\$** Read the current TCP/IP router address

Response String:

**^=IPGX xxx,xxx,xxx,xxx\$**

Where:

**xxx,xxx,xxx,xxx** = Gateway address in standard form (Ex: 192,168,001,001)

---

---

# Audio Control

---

## Audio Control

### Audio Control Commands

The commands are used to control the audio features of the switch.

Audio features include: Volume, Bass and Treble controls, a 5 Band Equalizer for each zone, lipsync delays, source and output gain settings, and more.

### Audio Command Definitions

---

#### 'LZ' Lock a Zone to Another Zone

When zones are locked, they share a single source, and volume, bass and treble controls. When the zone being followed changes its source, the locked zone will follow and will change to the same source. The same is true when the volume or tone (bass and treble) controls are changed.

This command is commonly used to lock a subwoofer to another source. But multiple zones can follow (be locked to) a zone. For instances if there are multiple zones driving speakers in a backyard or hallway, they can all be locked together and share a single source, and volume, bass and treble controls.

Command format:

`^LZ @zone ,zoneToFollow? $`                      Lock zone(s) to 'zoneToFollow'.

Response Strings:

`^=LZ @zone ,zoneToFollow $`

Where:

`@zone`                      = One (or more) zones to be locked.  
`zoneToFollow`            = The zone that the above zones will be locked to.

Caveats:

If you try to lock a zone already being followed to another zone, all the zones following the zone being locked, will revert to the unlocked (normal) mode.

You can't set a zone to follow itself, which can be useful. If you set a zone already being followed to follow itself, all the zones following it will revert to the unlocked mode, and it will also remain unlocked.

---

#### 'MV' Set Master Volume

This command controls the overall volume of the switch. It can be used to mute all zone (by setting it to '000'), or increase the overall gain of the switch, by setting it beyond 200 (0dB).

When set to 0dB (when the volume parameter is set to 200), the Master Volume has no effect on the audio path. Values above 0dB add gain to the overall audio levels, and levels below 0dB lower the gain.

Volume gain is set in 0.5dB steps.

Command format:

`^MV vol $`                                              Set the overall volume of the audio switch.

Response Strings:

`^=MV vol $`

---

# Audio Control

---

Where:

vol = Volume in 0.5dB steps with an offset of 200. Range is 0 to 248.

The volume command is given in 0.5dB steps and uses an offset of 200 to indicate a attenuation of 0.0dB, 199 indicates 0.5dB of attenuation, and so forth, all way down to 1, which indicates an attenuation of 99.5dB (or a gain of -99.5dB, depending on how you look at things).

The value 0 is special in that it indicates full attenuation, or fully muted audio (more than -115dB).

---

## 'VZ' Set Zone's Volume

Each zone has a volume level that can be set from 0dB attenuation (full volume), to -99.5dB attenuation, with a value of -100dB representing a full (more than -115dB) of mute.

Volume gain is set in 0.5dB steps.

To set a new volume level:

<code>^VZ @zone ,vol\$</code>	Set the volume of a zone or zones.
<code>^VZ @zone ,+step\$</code>	Add 'step' number of 0.5dB steps to current volume.
<code>^VZ @zone ,-step\$</code>	Subtract 'step' number of 0.5dB steps from current volume.
<code>^VZ @zone?\$</code>	Read current volume settings of given zone(s).

Response Strings:

`^=VZ @zone ,vol$`

Where:

@zone = One (or more) zones to be affected.

vol = Volume in 0.5dB steps with an offset of 200.

The volume command is given in 0.5dB steps and uses an offset of 200 to indicate a attenuation of 0.0dB, 199 indicates 0.5dB of attenuation, and so forth, all way down to 1, which indicates a gain of -99.5dB (or an attenuation of 99.5dB, depending on how you look at things).

The affect of this command, when the volume is muted, (*See: "VMZ' Audio Mute a Zone using Volume Levels" on page 34.*), is dependent upon the settings of the 'MJP' and 'UVL' bits of the 'XS' command (*See: "XS' Control Settings" on page 22.*).

If the 'MJP' flag is set, then sending a 'VZ' command with a '+' or '-' prefix will cause the zone to unmute, and the volume will jump to it's previous unmuted value. Subsequent '+' or '-' commands will increment or decrement from the new, unmuted value.

If the 'MJP' flag is cleared (reset), then sending a 'VZ' command with a '+' or '-' prefix will cause the zone to unmute, but the volume of the zone will not be restored to the original unmuted level. The '+' or '-' commands will increment or decrement the volume from the muted level.

If the 'UVL' flag is set, then sending a 'VZ' command without a '+' or '-' prefix will cause the zone to unmute. It will then be set to given level of the 'VZ' command.

If the 'UVL' flag is cleared (reset), then sending a 'VZ' command without a '+' or '-' prefix will not unmute the audio. The volume level will be changed, but the zone will remain muted. When the zone is unmuted it will be set to this new 'VZ' setting.

When setting, raising or lowering the volume, the 'VZ' command will not allow you to go above 'maxVol' setting (*See "VMAZ' Set Zone's Maximum Volume" on page 34.*), or below the 'minVol' setting (*See "VMIZ' Set Zone's Minimum Volume" on page 33.*), with an exception of '0', which is always allowed).

The value 0 is special in that it indicates full attenuation, or fully muted audio.

---

# Audio Control

---

The value returned from the VZ command will always be in the 0-248 range, regardless of the fade settings. For instance if you set the volume to '10100' (fade to 100 over time), the value returned will be 100, not 10100.

## 'VZ' Examples

The 'VZ' command in its simplest form:

```
^VZ @1,200$
```

causes the audio on zone '1' to be set to the level '200', which is full volume, without gain.

The easiest way to use the volume on the switch is to know that 200 is full volume, and 0 is off, and everything step from 0 to 200 makes the audio louder by 0.5dB.

*(Or perhaps an even easier way is to use the VPZ command instead of the VZ command. The VPZ command allows the volume to be set using a number between 0 and 100 regardless of the minimum and maximum volume settings. See "'VPZ' Set Zone's Volume as a Percentage" on page 32)*

To set the volume in absolute decibels, taking the gain in decibels, multiply by 2 (to account for 0.5dB steps), add 200 and use that as the volume level.

For instance if you wanted to attenuate the audio coming in by 24dB (the output level would be 24dB lower than the source level), you take the gain of -24dB, multiply by 2, and add 200. So:  $vol = -24 * 2 + 200 = 152$ , and:

```
^VZ @1,152$      -> Set the volume on zone 1 to -24.0dB
```

would set the volume gain on zone 1 to -24.0dB.

You can also easily add or subtract from the current volume in 0.5dB steps, for instance:

```
^VZ @1,+6$      -> Increase the volume (make it louder) on zone 1 by 3.0dB
^VZ @3,-9$      -> Decrease the volume (make it softer) on zone 3 by 4.5dB
```

---

## 'VPZ' Set Zone's Volume as a Percentage

The switch audio zones can have their volumes adjusted as a percentage level instead of a dB level. This can make writing drivers easier for many systems.

The VPZ command always uses a setting of 0 to 100, regardless of the minimum volume settings and the maximum volume settings. (See "'VMIZ' Set Zone's Minimum Volume" on page 33, and "'VMAZ' Set Zone's Maximum Volume" on page 34).

A value of 0 is always full mute.

A value of 1 is always the minimum volume level of the VMIZ command.

A value of 100 is always the maximum volume level of the VMAZ command.

Any value between 1 and 100 will be a percentage between minimum and maximum volume, to the nearest 0.5dB.

The format is:

<pre>^VPZ @zone,vol\$</pre>	Set the volume of a zone or zones.
<pre>^VPZ @zone,+step\$</pre>	Add 'step' number of 0.5dB steps to current volume.
<pre>^VPZ @zone,-step\$</pre>	Subtract 'step' number of 0.5dB steps from current volume.
<pre>^VPZ @zone?\$</pre>	Read current volume settings of given zone(s).

Response Strings:

```
^=VPZ @zone,vol$
```



---

# Audio Control

---

Where:

@zone = One (or more) zones to be affected.  
vol = Volume as a percentage: 0 through 100.

The affect of this command, when the volume is muted, (See: “**‘VMZ’ Audio Mute a Zone using Volume Levels**” on page 34.), is dependent upon the settings of the ‘MJP’ and ‘UVL’ bits of the ‘XS’ command (See: “**‘XS’ Control Settings**” on page 22).

If the ‘MJP’ flag is set, then sending a ‘VZ’ command with a ‘+’ or ‘-’ prefix will cause the zone to unmute, and the volume will jump to it’s previous unmuted value. Subsequent ‘+’ or ‘-’ commands will increment or decrement from the new, unmuted value.

If the ‘MJP’ flag is cleared (reset), then sending a ‘VZ’ command with a ‘+’ or ‘-’ prefix will cause the zone to unmute, but the volume of the zone will not be restored to the original unmuted level. The ‘+’ or ‘-’ commands will increment or decrement the volume from the muted level.

If the ‘UVL’ flag is set, then sending a ‘VZ’ command without a ‘+’ or ‘-’ prefix will cause the zone to unmute. It will then be set to given level of the ‘VZ’ command.

If the ‘UVL’ flag is cleared (reset), then sending a ‘VZ’ command without a ‘+’ or ‘-’ prefix will not unmute the audio. The volume level will be changed, but the zone will remain muted. When the zone is unmuted it will be set to this new ‘VZ’ setting.

All the examples given in by the ‘VZ’ command (See “**‘VZ’ Set Zone’s Volume**” on page 31) apply for the ‘VPZ’ command as well. Simply substitute the VPZ for the VZ command and use values between 0 and 100.

---

## **‘VMIZ’ Set Zone’s Minimum Volume**

Each zone has a minimum and maximum volume setting.

The maximum prevents a user from overdriving the speakers, amplifier, or even the switch switch.

The minimum allows you to set the volume to just below the noise level of a quiet room. The volume setting switch can be adjusted down to -99.5dB before muting. This is usually quite a bit beyond the noise level of any room, or the noise level of most amplifiers. Since adjusting the volume below the noise level is just a waste of time, the minimum level allows you to limit the volume range.

Attempting to set the volume below the minimum level (using the ‘VZ’ command) will cause the switch go into full mute (no sound). When adjusting the volume upward from full mute, the volume will immediately jump to the minimum level.

The level is set in 0.5dB steps.

Command format:

<b>^VMIZ @zone,minVol\$</b>	Set the minimum volume of a zone or zones.
<b>^VMIZ @zone,+step\$</b>	Add ‘step’ number of 0.5dB steps to current minimum.
<b>^VMIZ @zone,-step\$</b>	Sub ‘step’ number of 0.5dB steps from current minimum.
<b>^VMIZ @zone?\$</b>	Read current minimum setting(s) of given zone(s).

Response Strings:

**^=VMIZ @zone,minvol\$**

Where:

@zone = One (or more) zones to be affected.  
minVol = Minimum volume in 0.5dB steps with an offset of 200. Range is 0 to ‘maxVol’-99.

---

# Audio Control

---

The gain command is given in 0.5dB steps and uses an offset of 200 to indicate a gain of 0.0dB. Values above 200 raise gain, and values below 200 lower gain.

The minimum and maximum volume settings must have at least a 49.5dB (99 steps) difference between them. Any attempt to set the minimum volume closer than 99 steps of the maximum, will result in a setting of exactly 99 steps below the maximum volume setting.

---

## 'VMAZ' Set Zone's Maximum Volume

Each zone has a minimum and maximum volume setting.

The maximum prevents a user from overdriving the speakers, amplifier, or even the switch switch.

Setting the volume above the maximum level (using the 'VZ' command) is not allowed, and will be limited to the maximum volume set by this command.

The level is set in 0.5dB steps.

Command format:

<code>^VMAZ @zone,maxVol\$</code>	Set the minimum volume of a zone or zones.
<code>^VMAZ @zone,+step\$</code>	Add 'step' number of 0.5dB steps to current minimum.
<code>^VMAZ @zone,-step\$</code>	Sub 'step' number of 0.5dB steps from current minimum.
<code>^VMAZ @zone?\$</code>	Read current minimum setting(s) of given zone(s).

Response Strings:

`^=VMAZ @zone,minvol$`

Where:

`@zone` = One (or more) zones to be affected.

`maxVol` = Minimum volume in 0.5dB steps with an offset of 200. Range is 'minVol'+99 - 248.

The gain command is given in 0.5dB steps and uses an offset of 200 to indicate a gain of 0.0dB. Values above 200 raise gain, and values below 200 lower gain.

The minimum and maximum volume settings must have at least a 49.5dB (99 steps) difference between them. Any attempt to set the maximum volume closer than 99 steps of the minimum, will result in a setting of exactly 99 steps above the minimum volume setting.

---

## 'VMZ' Audio Mute a Zone using Volume Levels

The VMZ command allows you to mute the volume on the switch using the audio processor. This is the proper command to mute the audio of the switch. This differs from the MZ command which is used to completely shut down, or disconnect a zone.

You can mute to full off, or you can mute by a specific amount, -20dB for instance.

Also available are mute fade options. Instead of instantly jumping to a mute setting, you can fade over time, or fade at a given speed, to the new mute level.

There are four options when using mute:

- 0=Unmute - This disables muting, and returns the volume back to its original unmuted level.
- 1=Mute - This enables muting. It attenuates the volume by the mute level (See: VMLZ).
- 2=NoMute - This disables muting, but sets the volume level to the current muted level.
- +=Toggle - This will toggle the current setting. If muted then unmute, if unmuted then mute.

The '2' (NoMute) option allows the mute setting to be cancelled without affecting the volume level. Response strings for the 'VZ' and 'VPZ' commands will be sent to indicate the new volume to the

---

# Audio Control

---

controller. The response string for the 'VMZ' command will be a '0' to indicate muting is now disabled. A '2' will never be returned in the response string.

The VMZ command format is:

**^VMZ @zone,mute\$** ;Mute a zone.  
**^VMZ @zone?\$** ;Read current settings of given zones.

Response Strings:

**^=VMZ @zone,mute\$**

Where:

**@zone** = One (or more) zones to be mapped.  
**mute** = Mute setting (0=Unmuted, 1=Muted, 2=NoMute).

A value of '2' is never returned by the response string. If muting is disabled by the '2' (NoMute) command, a value of '0' will be returned to indicate muting is disabled.

Mute fading and the level of muting are all predetermined by the 'VMLZ' and 'VMT' commands. For instance if mute fading has been previously set using the 'VMLZ' command, then sending the "**^VMZ @1,1\$**" command will cause the volume on zone1 to fade to the muted level.

To toggle the mute setting use the '+' setting:

**^VMZ @1,+\$** ;Toggle the mute setting on zone 1.

---

## 'BLZ' Set Zone's Balance

Each zone has a balance level that can be set from 0 attenuation (full left), to 400 (full right).

Balance is set in 0.5dB steps.

Command format:

**^BLZ @zone,bal\$** Set the balance of a zone or zones.  
**^BLZ @zone,+step\$** Add 'step' number of 0.5dB steps to current balance.  
**^BLZ @zone,-step\$** Subtract 'step' number of 0.5dB steps from current balance.  
**^BLZ @zone?\$** Read current balance settings of given zone(s).

Response Strings:

**^=BLZ @zone,bal\$**

Where:

**@zone** = One (or more) zones to be affected.  
**bal** = Balance in 0.5dB steps from left to right, 200=Center. Range is 0 to 400.

The balance command is given in 0.5dB steps and uses an offset of 200 to indicate center, 0 indicates full left (no audio on right channel), 400 indicates full right (no audio on left channel).

The balance command does not add gain, it only attenuates. Moving the balance to the left will lower the volume on the right channel, but does not increase the volume on the left channel.

The value 200 is "center" and allows audio to pass unaffected on both left and right channels.

For instance:

**^BLZ @1,206\$** ;Move zone 1's balance to the right by lowering left channel by 3dB.  
**^BLZ @1,194\$** ;Move zone 1's balance to the left by lowering right channel by 3dB.

---

# Audio Control

---

## 'GAZ' Set Zone's Gain (Output Levels)

Each zone has a gain level that can be set from -24dB to +24dB. The gain control allows you to match the volume levels of each zone to the sensitivity of the amplifiers used in each zone. This allows volume settings to sound the same for each zone, even when using different model/brand of amplifiers or different model/brand of speakers.

Zone gain is set in 0.5dB steps.

Command format:

<code>^GAZ @zone,gain\$</code>	Set the gain of a zone or zones.
<code>^GAZ @zone,+step\$</code>	Add 'step' number of 0.5dB steps to current gain.
<code>^GAZ @zone,-step\$</code>	Subtract 'step' number of 0.5dB steps from current gain.
<code>^GAZ @zone?\$</code>	Read current gain settings of given zone(s).

Response Strings:

`^=GAZ @zone,gain$`

Where:

`@zone` = One (or more) zones to be affected.  
`gain` = Gain in 0.5dB steps with an offset of 200. Range is 152 to 248.

The gain command is given in 0.5dB steps and uses an offset of 200 to indicate a gain of 0.0dB. Values above 200 raise gain, and values below 200 lower gain.

---

## 'GAI' Set Source's Gain (Input Trimming)

Each source has a gain level that can be set from -24dB to +24dB. The gain control allows you to match the volume levels of each source to the different levels of each source. This allows volume settings to sound the same, when switching between source devices with different output levels.

Source gain is set in 0.5dB steps.

Command format:

<code>^GAI @source,gain\$</code>	Set the gain of a source or sources.
<code>^GAI @source,+step\$</code>	Add 'step' number of 0.5dB steps to current source.
<code>^GAI @source,-step\$</code>	Subtract 'step' number of 0.5dB steps from current source.
<code>^GAI @source?\$</code>	Read current gain settings of given source(s).

Response Strings:

`^=GAZ @source,gain$`

Where:

`@source` = One (or more) sources to be affected.  
`gain` = Gain in 0.5dB steps with an offset of 200. Range is 152 to 248.

The gain command is given in 0.5dB steps and uses an offset of 200 to indicate a gain of 0.0dB. Values above 200 raise gain, and values below 200 lower gain.

---

## 'BAZ', 'TRZ' Set Zone's Bass and Treble Levels

Each zone has a Bass and Treble level that can range from -20.0dB to +20.0dB. Interaction between the Bass and Treble controls and the 5 Band Equalizer settings are limited to a +/-20.0dB range. Add-

---

# Audio Control

---

ing a boost of +20.0dB to the Bass setting as well as the EQ1 and/or EQ2 equalizer bands, will not result in a +40.0dB, but will be limited to +20.0dB.

The levels are set in -0.5dB steps.

Command formats:

<code>^BAZ @zone,level\$</code>	Set the bass level of a zone or zones.
<code>^BAZ @zone,+step\$</code>	Add 'step' number of 0.5dB steps to current bass level.
<code>^BAZ @zone,-step\$</code>	Subtract 'step' number of 0.5dB steps from current level.
<code>^BAZ @zone?\$</code>	Read current bass level settings of given zone(s).
<code>^TRZ @zone,level</code>	Set the treble level of a zone or zones.
<code>^TRZ @zone,+step\$</code>	Add 'step' number of 0.5dB steps to current treble level.
<code>^TRZ @zone,-step\$</code>	Subtract 'step' number of 0.5dB steps from current level.
<code>^TRZ @zone?\$</code>	Read current bass level settings of given zone(s).

Response Strings:

`^=BAZ @zone,level$`  
`^=TRZ @zone,level$`

Where:

`@zone` = One (or more) zones to be affected.  
`level` = Levels 0.5dB steps with offsets of 128 (128 = 0.0dB).

The levels in the commands are given in 0.5dB steps and use an offset of 128 to indicate a boost / cut of 0.0dB. Values above 128 boost level by 0.5dB per step, for instance 129 = +0.5dB (a boost of 0.5dB), and 127 = -0.5dB (a cut of 0.5dB).

## 'BAZ' ('TRZ') Examples

Examples are only given for the bass command 'BAZ', for treble commands use 'TRZ' in the place of 'BAZ'. The simplest form of the command is:

`^BAZ @1,128$`

and causes the bass level on zone '1' to be set to 0.0dB which is flat (no boost or cut).

The easiest way to use the volume on the switch is to know that 168 is full boost, and 88 is full cut and every step from 88 to 168 boosts the bass level by 0.5dB, with 128 used to indicate no boost or cut.

Or you can set a zone to an absolute level, in decibels, by taking the gain (which is positive for boost and negative for cut), multiplying by 2 (to take into account the 0.5dB steps), adding 128, and using that as the new level.

For instance if you wanted to boost bass by 9dB, you'd take the gain: 9dB multiply by 2 and add 200. So 'level = 9\*2 + 128 = 146', and:

`^BAZ @1,146$`      -> Boost bass by 9.0dB

would set the bass boost on zone 1 to 9.0dB.

You can also easily add or subtract from the current level in 0.5dB steps, for instance:

`^BAZ @1,+6$`      -> Boost bass on zone 1 by 3.0dB  
`^BAZ @3,-9$`      -> Cut bass on zone 3 by 4.5dB

When using the '+' and '-' prefixes to add or subtract from current levels, the 'BAZ' command will not allow you to go above 168 (+20.0dB) or below 88 (-20.0dB), though it's perfectly valid to send a command that would attempt this.

---

# Audio Control

---

## 'EQ1Z', 'EQ2Z', 'EQ3Z', 'EQ4Z', 'EQ5Z' Set the 5 Band Equalizers' levels

Each zone has a 5 Band Equalizer associated with it, each of the five bands have levels that range from -20.0dB to +20.0dB. Interaction between the Bass and Treble controls and the 5 Band Equalizer settings are limited to a +/-20.0dB range. Adding a boost of +20.0dB to the Bass setting as well as the EQA or EQB equalizer bands, will not result in a +40.0dB, but will be limited to +20.0dB.

The levels are set in -0.5dB steps.

The band center frequencies for each command are:

EQ1Z - 100Hz (and below, this is a low frequency slope filter.)  
EQ2Z - 330Hz  
EQ3Z - 1,000Hz  
EQ4Z - 3,300Hz  
EQ5Z - 10,000Hz (and above, this is a high frequency slope filter.)

Only the 'EQ1Z' command will be described, all the bands use identical syntax.

Command format:

<code>^EQ1Z @zone,level\$</code>	Set the EQ level of a zone or zones.
<code>^EQ1Z @zone,+step\$</code>	Add 'step' number of 0.5dB steps to current EQ level.
<code>^EQ1Z @zone,-step\$</code>	Subtract 'step' number of 0.5dB steps from current level.
<code>^EQ1Z @zone?\$</code>	Read current EQ level settings of given zone(s).

Response String:

`^=EQ1Z @zone,level$`

Where:

@zone = One (or more) zones to be affected.  
level = Levels 0.5dB steps with offsets of 128 (128 = 0.0dB).

The levels in the commands are given in 0.5dB steps and use an offset of 128 to indicate a boost / cut of 0.0dB. Values above 128 boost level by 0.5dB per step, for instance 129 = +0.5dB (a boost of 0.5dB), and 127 = -0.5dB (a cut of 0.5dB).

### 'EQ1Z' Examples

Examples are only given for the EQ band-1 command 'EQ1Z', for other bands, substitute the proper command for the desired band. The simplest form of the command is:

`^EQ1Z @1,128$`

and causes the EQ level on zone '1' to be set to 0.0dB which is flat (no boost or cut).

The easiest way to use the equalizer settings on the switch is to know that 168 is full boost, and 88 is full cut and every step from 88 to 168 boosts the EQ level by 0.5dB, with 128 used to indicate no boost or cut.

Or you can set a zone to an absolute level, in decibels, by taking the gain (which is positive for boost and negative for cut), multiplying by 2 (to take into account the 0.5dB steps), adding 128, and using that as the new level.

For instance if you wanted to boost the EQ level by 9dB, you'd take the gain: 9dB multiply by 2 and add 128. So 'level = 9\*2 + 128 = 146', and:

`^EQ1Z @1,146$` -> Boost EQ band's level by 9.0dB

would set the EQ level on zone 1 to a 9.0dB boost.

---

# Audio Control

---

You can also easily add or subtract from the current level in 0.5dB steps, for instance:

`^EQ1Z @1,+6$`      -> Boost EQ band's level on zone 1 by 3.0dB  
`^EQ1Z @3,-9$`      -> Cut EQ band's level on zone 3 by 4.5dB

When using the '+' and '-' prefixes to add or subtract from current levels, the 'EQxZ' commands will not allow you to go above 168 (+20.0dB) or below 88 (-20.0dB), though it's perfectly valid to send a command that would attempt this.

---

## 'FTYPZ' Low or High Pass Filter Type

This command enables a low or high pass filter on a given zone. This is typically used for subwoofer control. By implementing a High Pass Filter, you can remove bass from smaller bookshelf speakers that may be damaged by too much bass. By implementing a low pass filter, you can send only the bass to a subwoofer to prevent high frequencies distortions that occur when high frequency audio is sent to a subwoofer.

To enable a filter, you must set both the filter type, using this command, and the crossover frequency, using the 'FFRQZ' command. If either of these commands are set to '0', the filter will be disabled. By default, both of these commands are set to '0'.

Command format:

`^FTYPZ @zone,type$`      Set the output filter of zone(s).

Response Strings:

`^=FTYPZ @zone,type$`

Where:

`@zone` = One (or more) zones to be affected.  
`type` = The filter type being selected.  
0 = No filter selected, the filter is bypassed.  
1 = Reserved, (acts the same as above).  
2 = 12dB per octave Low Pass Filter.  
3 = 12dB per octave High Pass Filter.  
4 = 24dB per octave Low Pass Filter.  
5 = 24dB per octave High Pass Filter.

---

## 'FFRQZ' Set the Crossover Frequency of a Low or High Pass Filter

This command set the frequency of a low or high pass filter on a given zone. This is typically used for subwoofer control. By implementing a High Pass Filter, you can remove bass from smaller bookshelf speakers that may be damaged by too much bass. By implementing a low pass filter, you can send only the bass to a subwoofer to prevent high frequencies distortions that occur when high frequency audio is sent to a subwoofer.

To enable a filter, you must set both the filter type, using this command, and the filter type, using the 'FTYPZ' command. If either of these commands are set to '0', the filter will be disabled. By default, both of these commands are set to '0'.

You can select crossover frequencies from 50Hz to 300Hz

Command format:

`^FFRQZ @zone,freq$`      Set the filter's crossover frequency on zone(s).

Response Strings:

`^=FFRQZ @zone,freq$`

---

# Audio Control

---

Where:

@zone = One (or more) zones to be affected.  
freq = The crossover frequency.  
0 = No filter selected, the filter is bypassed.  
1 = 50Hz.  
2 = 53Hz.  
3 = 56Hz.  
4 = 59Hz.  
5 = 63Hz.  
6 = 67Hz.  
7 = 71Hz.  
8 = 75Hz.  
9 = 79Hz.  
10 = 84Hz.  
11 = 89Hz.  
12 = 94Hz.  
13 = 100Hz.  
14 = 106Hz.  
15 = 112Hz.  
16 = 119Hz.  
17 = 126Hz.  
18 = 133Hz.  
19 = 141Hz.  
20 = 150Hz.  
21 = 159Hz.  
22 = 168Hz.  
23 = 178Hz.  
24 = 189Hz.  
25 = 200Hz.  
26 = 212Hz.  
27 = 224Hz.  
28 = 238Hz.  
29 = 252Hz.  
30 = 267Hz.  
31 = 283Hz.  
32 = 300Hz.

---

## 'MXZ' Stereo to Mono Down Mix for a Zone

Stereo audio can be mix in a number of different ways. This includes swapping channels, and a number of ways of creating mono audio.

Command format:

**^MXZ @zone,mix\$** Set the stereo mix down of a zone or zones.

Response Strings:

**^=MXZ @zone,mix\$**

Where:

@zone = One (or more) zones to be affected.



---

# Audio Control

---

mix = Stereo mix down flag.  
0 = No change to audio.  
1 = Swap left and right channels.  
2 = Mix to mono by adding left and right channels.  
3 = Mix to mono by using the left channel for both outputs.  
4 = Mix to mono by using the right channel for both outputs.  
5 = Mix to mono by subtracting the right channel from the left.  
6 = Mix to mono by subtracting the left channel from the right.

---

## 'MXI' Stereo to Mono Down Mix for an Input

Stereo audio can be mix in a number of different ways. This includes swapping channels, and a number of ways of creating mono audio.

Command format:

`^MXI @source ,mix$` Set the stereo mix down of an input or inputs.

Response Strings:

`^=MXI @source ,mix$`

Where:

@source = One (or more) inputs to be affected.  
mix = Stereo mix down flag.  
0 = No change to audio.  
1 = Swap left and right channels.  
2 = Mix to mono by adding left and right channels.  
3 = Mix to mono by using the left channel for both outputs.  
4 = Mix to mono by using the right channel for both outputs.  
5 = Mix to mono by subtracting the right channel from the left.  
6 = Mix to mono by subtracting the left channel from the right.

---

## 'DRZ' Digital audio Routing

This command allows you to use a coax output as an independent digital audio switch, or to connect it to the analog output of its associated zone.

When used as an independent switch, the 'DSZ' command is used to switch between all the digital sources. All digital audio is passed through the switch untouched. This includes Dolby5.1 and DTS audio. However, you cannot switch the coax output to any analog audio sources.

When connected to an associated analog zone, the setting of the 'DSZ' command is ignored, and all analog audio from the associated analog zone will be converted to PCM stereo and sent to the coax audio output. (Use the 'SZ' command is used to switch between the analog zone's sources. The coax outputs will mirror the analog zones.)

The associated analog zone for each coax is the matching analog zone number. Coax output 1's associated zone is analog output 1. Coax 2, associates with analog output 2, etc.

Command format:

`^DRZ @zone ,routing$` Set the digital routing option.

Response Strings:

`^=DRZ @zone ,routing$`

---

# Audio Control

---

Where:

@zone = One (or more) coax zones to be affected.  
routing = Routing flag.  
0 = Coax output is an independent digital audio zone set by the 'DSZ' command.  
1 = Coax output mirrors its associated analog audio zone.

For example, to use Coax output 1 as digital routing zone, send:

`^DRZ @1,0$` ; Set the digital output up as a digital audio zone.

You can now use the 'DSZ' command to route audio from any digital source to coax output 1:

`^DSZ @1,17$` ;Set the digital output 1 to coax source 17.

Audio from input 17 will be routed to coax output 1.

Keeping with the above example, if you now send a:

`^DRZ @1,1$` ;Connect coax output to mirror analog output.

The DSZ setting will be ignored, and the coax output will now be a PCM-stereo output that mirrors analog output 1. Anything on analog output 1 will also be sent to coax output 1. All volume, bass, treble, etc. commands that affect analog output 1 will also affect coax output 1.

Note: A more versatile form of this command is to use the source inputs 129-144 of the 'DSZ' command. These sources allow you to mirror any of the analog outputs (of the same block) to a coax output. See: **"DSZ' Set Zone(s) -- Digital Audio Switch"** on page 15.

---

## 'LSZ' Set Lip Sync delay for a Zone

Each zone is capable of delaying audio for up to 170.65mS. The delay resolution is 1/48000 of a second, or 48 counts per millisecond.

Command format:

`^LSZ @zone ,delay$` Set the lip sync delay of a zone or zones.  
`^LSZ @zone ,+step$` Add 'step' to zone's delay.  
`^LSZ @zone ,-step$` Subtract 'step' from zone's delay.

Response Strings:

`^=LSZ @zone ,delay$`

Where:

@zone = One (or more) zones to be affected.  
delay = Lip sync delay in 48KHz samples, each count delays audio by 1/48000 of a second.

A value of 0 indicates no delay.

There are two forms of the lip sync delay commands, the 'LSZ' command which sets lip sync delay on a zone by zone basis, and the 'LSI' command which sets delays on a source by source basis. Which one to use depends on what is causing the video delay, and where it's connected in the video path.

If the source of a video processor (or monitor) that adds a video delay is connected to a zone, then the 'LSZ' command should be used to delay the audio by the same amount of time the video is being delayed. Only audio that is being sent to the video processor, attached to the zone, will be delayed.

If on the other hand, if the output of a video processor that adds a video delay, is connected to an source, then the 'LSI' command should be used. This allows all zones that connect to the video processor, to have their delays automatically adjusted to compensate for the processor's video delays.

---

# Audio Control

---

Both sources and zones can have delays set, however the overall delay cannot exceed 8191 counts (or 170.65mS). If an source and zone combination exceeds 8191 counts, no error will be generated and the delay will be set to 8191.

## 'LSZ' Examples

The lip sync delay is given in samples of a 48KHz clock, or 48 counts per millisecond. To delay the audio by a given number of milliseconds, take the number of milliseconds, multiply by 48, and use that as the parameter for the command. For instance to delay zone 3 by 24mS:  $24\text{mS} * 48 = 1152$

`^LSZ @3,1152$`                    -> Delay zone 3 by 1152 samples (24mS)

Another common timing to apply to lip sync delay is video frame times. Most video processor delay video by a set number of frames, matching audio to video in these cases is done by delay audio by the same number of frames. For instance 1080p60, is 60 frames per second, while 1080p24, is 24 frames per second. To calculate the value needed to delay a set number of frame, divide the frames per second into 48000, and multiply by the number of frames to delay. For instance if your video signal is 60 frames per second, and you want to delay zone 3 by 2 frames:  $(48000 / 60\text{fps}) * 2 = 1600$

`^LSZ @3,1600$`                    -> Delay zone 3 by 1600 samples (two 60fps frames)  
`^LSZ @2,+800$`                    -> Add one 60fps frame of delay to zone 2

---

## 'LSI' Set Lip Sync delay for an Input

Each source is capable of delaying audio for up to 170.65mS. The delay resolution is 1/48000 of a second, or 48 counts per millisecond.

Command format:

<code>^LSI @source,delay\$</code>	Set the lip sync delay an source or sources.
<code>^LSI @source,+step\$</code>	Add 'step' to source's delay.
<code>^LSI @source,-step\$</code>	Subtract 'step' from source's delay.

Response Strings:

`^=LSI @source,delay$`

Where:

`@source`    = One (or more) sources to be affected.  
`delay`       = Lip sync delay in 48KHz samples, each count delays audio by 1/48000 of a second.

A value of 0 indicates no delay.

There are two forms of the lip sync delay commands, the 'LSZ' command which sets lip sync delay on a zone by zone basis, and the 'LSI' command which sets delays on a source by source basis. Which one to use depends on what is causing the video delay, and where it's connected in the video path.

If the source of a video processor (or monitor) that adds a video delay is connected to a zone, then the 'LSZ' command should be used to delay the audio by the same amount of time the video is being delayed. Only audio that is being sent to the video processor, attached to the zone, will be delayed.

If on the other hand, if the output of a video processor that adds a video delay, is connected to an source, then the 'LSI' command should be used. This allows all zones that connect to the video processor, to have their delays automatically adjusted to compensate for the processor's video delays.

Both sources and zones can have delays set, however the overall delay cannot exceed 8191 counts (or 170.65mS). If an source and zone combination exceeds 8191 counts, no error will be generated and the delay will be set to 8191.

For examples, see the examples of the 'LSZ' command on page 42.

---

# Audio Control

---

## 'ATZ' Audio Type for an analog zone

This command is read only and returns the type of audio that is being sent to an analog zone.

Command format:

`^ATZ @zone?$` Request the audio type of a zone.

Response Strings:

`^=ATZ @zone, audio_type$`

Where:

`@zone` = analog audio zone.

`audio_type delay` = Type of audio present at the zone's source.

The audio types that can be returned are:

- 0 - Analog input, or transitioning (mode unknown).
- 1 - No SPDIF detected on a digital input.
- 2 - Stereo PCM detected on a digital input.
- 3 - Encoded SPDIF (Dolby, DTS, etc) detected on a digital input.

This will be sent anytime the audio source changes its status.

The switch cannot down convert Dolby or DTS audio and when anything other than PCM stereo is detected, the switch will mute the output. This command can be used to indicate to the user that Dolby/DTS is being received and that the output is purposely being muted.

---

# 12V Triggers

---

## 12V Triggers

### 12V Trigger Overview

The 12V trigger outputs can be controlled discretely using the ‘STRG’ command. Or by using the ‘TRGZ’ command, a trigger can setup to follow the power states of given zones, and to turn on when any of the given zones are powered on, and turn off when all of the given zones are powered off.

---

#### ‘STRG’ Set a 12V trigger

This command allows you to turn on, or turn off, a 12V trigger output. Or to setup the output to be controlled by the power setting of a zone.

Command format:

`^STRG @trg,setting$`                      Set the trigger state.

Response Strings:

`^=STRG @trg,setting$`

Where:

`@trg`            = One (or more) 12V triggers to be affected.  
`setting`        = The 12V trigger setting as follows:  
                  0 = Turn off the 12V trigger output, overrides any zone power settings.  
                  1 = Turn on the 12V trigger output, overrides any zone power settings.  
                  2 = 12V trigger is turned on/off based on a zone, or zones power state(s).

---

#### ‘TRGZ’ Set a zone to control a 12V trigger

This command allows you to associate a zone with a 12V trigger. When a zone is associate with a 12V trigger, the 12V trigger will turn on any time the zone is powered on.

If more that one zone is associated with a 12V trigger, the 12V trigger will turn on whenever any asso-  
ciated zone is powered on, and will turn off only when all associated zones are powered off.

The ‘PZ’ command (See: “‘PZ’ Power Down (Disconnect) Zone(s) -- Analog Audio Switch” on  
page 18), determines if an audio zone is powered on or off.

For a trigger to be controlled by a zone, the trigger’s ‘STRG’ command must be set to ‘2’.

Command format:

`^TRGZ @zone,trg$`                      Set the 12V trigger associated with this zone.

Response Strings:

`^=TRGZ @zone,trg$`

Where:

`@zone`            = One (or more) zones to be associated with a 12V trigger.  
`trg`                = The 12V trigger to be associated with the zone(s).

---

---

# Paging and Doorbell

---

## Paging and Doorbell

### Paging and Doorbell Overview

To use paging, paging “presets” are setup in advance. Once these are setup, a paging command is issued, and the switch will switch to the given presets. Once the page is over, either by timing out, or an “end page” command is sent, the switch will return to its previous settings.

This allows the controller to setup all paging in advance. During a page, only a single command is sent to change all sources, and paging volumes.

Each zone has an individual “Do Not Disturb” flag setting. This allows zones to be setup in advance, but can then be disabled on a zone by zone basis. It allows the controller to select which zones are affected without having rewrite all the presets.

The switch does not have a doorbell sound generator built in. To generate a doorbell sound, a doorbell sound generator of some type must be attached to an input. You can then use the paging presets to switch zones to the doorbell source during a page.

To implement a paging function, a microphone preamp is needed. The preamp is connected to an audio input, and paging presets are used to switch zones to the microphone source for the duration of the page.

### Command flow for setting up a page

To setup a page, you must start by issuing pASZ (and or pDSZ) commands to indicate which inputs to switch to during a page. This can be an input connected to a chime generator in the case of a doorbell, or a microphone preamp in the case of a page.

Use the pVZ or the pVPZ commands to set the desired paging / doorbell volumes for each zone.

By default each zone has it's DND (Do Not Disturb) flag set to Do Not Disturb '1'. You must set the DND flag to '0' if you want the zone to switch to the doorbell / paging input when the doorbell / page is triggered.

If you want a zone to only mute or lower its volume during a page, but not switch to the doorbell chime (for instance, to turn off hallway speakers, but only play the chime in the living room), set the DND flag to '2'.

Setting the DND flag to '1' will cause the zone to be ignored during a page. This allows you to remove a zone from a page temporarily, for instance when the baby is asleep you might want to set a DND flag for the baby's room to '1'.

If you are going to use the back panel connections for paging, you must setup the behavior of the back panel triggers using the pDSW command. If you are going to let the user decide how long a page last (by hold down a button), look into using the pTIME command to set a minimum page time, so that simply tapping a switch allows enough time for a minimum doorbell ring.

Once you have setup the paging commands you may want to issue a “^SS 4096\$” command to save the settings in EEPROM so that are not lost during a power fail.

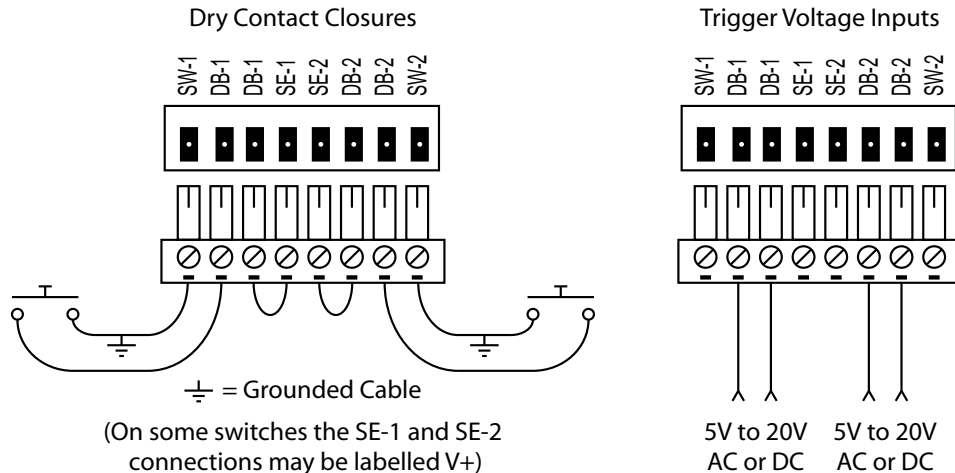
Once all the zones are programmed for switching, and volume levels set, and the do-not-disturb flags are set properly, and the back panel switch behaviors are setup, you are ready to test paging.

You can now start a page by either issuing a 'pSET x' command. Or by triggering a page using the back panel dry contact closures (a button or relay), or a voltage trigger.

# Paging and Doorbell

## Back Panel Doorbell / Page Wiring

The back panel can be wired to accept dry contact closures (for push buttons or relay triggers), or for voltage triggers. You can mix connection types and setup one connection for dry closures, and the other for voltage triggers.



To use dry contact closures, add a jumper between the SE and DB connections, and connect the switch to other DB input and the SW input. Note that the SW-1 and SW-2 connections are also connected to the chassis ground. If you are using a shielded cable to the switch, connect the SW side to the shield. If one side of your button is grounded, the side that is grounded should use the SW connection.

The voltage trigger inputs can be wired directly to the doorbell transformer, so that when the doorbell rings, paging will also occur. When used in this manner, the doorbell system can be setup to mute or lower the volume of the audio during a doorbell press, to allow the doorbell to be heard.

## Doorbell and Paging Commands

### 'pASZ', 'pDSZ', Set the Paging Source of a Zone

Setup the source a zone will use during a page.

Each zone can be preset to switch to a given source. This can be overridden using the do not disturb settings.

The 'pASZ' command controls the analog audio sources, and is similar to the 'SZ' command.

The 'pDSZ' command controls the digital audio sources, and is similar to the 'DSZ' command.

Only the 'pASZ' command will be described, but both commands work the same way.

Command format:

```
^pASZ preset,@zone,source$
```

Response Strings:

```
^=pASZ preset,@zone,source$
```

Where:

preset = A preset level greater than 0. A preset of 0 is ignored.

zone = Zone number.

source = Source to switch to during a page.

---

## Paging and Doorbell

---

When a 'pSET x' command is issued, the preset sources set by this command, will used.

---

### 'pAMZ', 'pDMZ', Set the Muting of a Zone during a Page

Setup the mute value used during a page. These mute settings are hard mutes and are the same as setting a source to '0'.

If you want to mute an analog audio zone during a page, you should use the 'pVMZ' command. The 'pVMZ' command mutes by smoothly turning down the volume. The 'pAMZ' command has the same affect as switching a zone to source '0'.

Each zone can be preset to mute during a page. This can be overridden using the do not disturb settings.

The 'pAMZ' command controls the analog audio sources, and is similar to the 'MZ' command.

The 'pDMZ' command controls the digital audio sources, and is similar to the 'DMZ' command.

Only the 'pAMZ' command will be described, but both commands work the same way.

Command format:

```
^pAMZ preset, @zone, mute_flag$
```

Response Strings:

```
^=pAMZ preset, @zone, mute_flag$
```

Where:

preset = A preset level greater than 0. A preset of 0 is ignored.  
zone = Zone number.  
mute\_flag = Mute flag (0 or 1).

When a 'pSET x' command is issued, the zone(s) will mute, based on these settings.

---

### 'pVZ', 'pVPZ' Set the Volume of a Zone during a Page

Paging volumes can be setup as absolute or relative volumes.

Absolute volumes are used to jump to a preset level, regardless of the current level of a zone.

Relative volumes are used to adjust the volume up or down (or no change), from the current zone's volume.

Absolute volumes are given with no prefix, relative volumes are given with a '+' or '-' prefix.

A value of "+0", is allowed, and it indicates the volume is not to change from is current level.

The 'pVZ' uses volumes given in dBs (0 = mute, 1 = -99.5dB, 200 = 0db, 248 = +24dB).

The 'pVPZ' uses volumes given in percentages from 0 to 100%.

These commands only work on the analog audio zones.

Command format:

```
^pVZ preset, @zone, vol_db$
```

```
^pVPZ preset, @zone, vol_percent$
```

Response Strings:

```
^=pVZ preset, @zone, vol_db$
```

```
^=pVPZ preset, @zone, vol_percent$
```



---

# Paging and Doorbell

---

Where:

preset = A preset level greater than 0. A preset of 0 is ignored.  
zone = Zone number.  
vol\_db = Volume in dB.  
vol\_percent = Volume as a percentage.

---

## **'pVMZ', Set the Volume Mute level of a Zone during a Page**

The command is the same as the 'VMZ' command, and is active during a page.

Command format:

`^pVMZ preset,@zone,mute_flag$`

Response Strings:

`^=pVMZ preset,@zone,mute_flag$`

Where:

preset = A preset level greater than 0. A preset of 0 is ignored.  
zone = Zone number.  
mute\_flag = Mute flag (0 or 1).

---

## **'pADNDZ', 'pDDNDZ', Set the Do Not Disturb flags**

Each zone can be set to "do not disturb". This is used to ignore the 'pSET' command on a zone by zone basis. By default, all zones are preset to 'Do Not Disturb', and must be changed before any paging commands will have an effect.

There are three levels of "Do Not Disturb".

Level 0 = Allow paging. The zone will respond fully to all paging requests.  
Level 1 = Do Not Disturb. All paging requests will be ignored by this zone.  
Level 2 = Allow paging requests, but do not change sources.

Level 2 is useful when you want a source to mute (or lower its volume), but you do not want it to switch away from its current source. If a zone is set to level 2, it will pay attention to 'pXMZ', 'pVMZ', 'pVZ', and 'pVPZ' settings, but will ignore 'pXSZ' settings.

The 'pADNDZ' command is used to control the analog audio zones.

The 'pDDNDZ' command is used to control the digital audio zones.

Only the 'pADNDZ' command will be described, but both commands work the same way.

Each preset has its own Do Not Disturb flags.

Command format:

`^pADNDZ preset,do_not_disturb_flag$`

Response Strings:

`^=pADNDZ preset,do_not_disturb_flag$`

Where:

preset = A preset level greater than 0. A preset of 0 is ignored.  
do\_not\_disturb\_flag = The "Do Not Disturb" flag (0, 1, or 2).

---

## **'pDSW', Setup the back panel doorbell / paging switches**

---

## Paging and Doorbell

---

There are two voltage triggers / dry contact closures inputs on the back panel of the switch. These can be setup to trigger a doorbell / page.

Command format:

```
^pDSW switch ,action ,preset ,time$
```

Response string:

```
^=pDSW switch ,action ,preset ,time$
```

Where:

switch	= The switch number 1 or 2.
action	= The switch action that triggers a page: 0 = Disable switch. 1 = Active on (switch closed, or trigger voltage present). 2 = Active off (switch open, or trigger voltage absent). 3 = Timed on (switch closed, or trigger voltage present). 4 = Timed off (switch open, or trigger voltage absent).
preset	= The paging preset level greater than 0. A preset of 0 is ignored.
time	= When using a timed mode, this is how long the page is active.

The 'switch' parameter refers to one of the two contact / voltage trigger inputs on the back panel.

The 'action' parameter refers to the type of switch action that triggers a page.

Active on = When the contact is closed, or a trigger voltage is present, initiate paging. Page will last as long as the contact is closed, or trigger voltage present. The 'time' parameter is ignored.

Active off = When the contact is open, or trigger voltage is absent, initiate paging. Page will last as long as the contact is open, or trigger voltage is absent. The 'time' parameter is ignored.

Timed on = When the contact is closed, or a trigger voltage appears, paging will start. The paging will last until the 'time' parameter amount of time has passed. A new page will not be triggered until a the contacts are opened and re-closed, or the trigger voltage is removed and restored.

Timed off = When the contact is opened, or a trigger voltage is removed, paging will start. The paging will last until the 'time' parameter amount of time has passed. A new page will not be triggered until a the contacts are closed and re-opened, or the trigger voltage appears and then removed.

The 'time' parameter is used by the timed settings to indicate the amount of time the page will last when a button is pressed, or a trigger voltage is applied. It is ignored in the 'active on' and 'active off' modes.

Note: There is an overall timeout of two minutes for any page. For instance holding a contact closure closed, in the 'active on' mode, will only page for two minutes. After two minutes the switch will return to normal mode.

---

### 'pTIME' Set Initial Delays and Minimum Paging Times

When paging is triggered, you can set a fixed amount of time to wait before sources are changed, and you can set the minimum amount of time a page will last.

The 'initial delay' can be used to allow a power amp to turn on before switching to the doorbell sound (or any other reason you may not want a page to be instantly active). This is currently limited to 30 seconds. When a user presses a doorbell, nothing will happen until the 'initial delay' amount of time has passed. This value should be kept as small as possible.

---

## Paging and Doorbell

---

The 'minimum time' is used to allow at least enough time to play a doorbell sound. This is useful when a doorbell button is pressed. You may want to setup paging so that a page is active the full time a user holds a button down. But a user may press and release a button very quickly. If the user releases a button before the 'minimum' amount of time has elapsed, the page will continue until the 'minimum' amount of time has passed before going back to normal operations.

The switch's paging/doorbell times are hard limited to a maximum duration of two minutes. If a button is held longer than this, or a 'pSET x' command is sent without a follow up 'pSET 0' command, the page will timeout in 2 minutes.

Command format:

```
^pTIME preset,init_dly,min_time$
```

Response Strings:

```
^=pTIME preset,init_dly,min_time$
```

Where:

preset = The preset level.  
init\_dly = Initial delay before switching sources, in milliseconds (1000 = 1 second).  
min\_time = The minimum time of a page, in milliseconds (1000 = 1 second).

---

### 'pSET' Initiate or End a Page

A controller can use this command to initiate or end a page.

There are three (3) paging presets. Paging preset 0, is the normal mode of the switch (paging is off).

Paging presets 1 and 2 are used for paging. One preset cannot interrupt another level. For instance, if a 'pSET 1' is sent to the switch, it cannot be followed by 'pSET 2', only a 'pSET 0' is allowed to follow an active page, anything else, will be ignored.

If a 'time' value is given, the page will last for the amount of time given, and then return to normal mode. Otherwise, the page will last until a 'pSET 0' is given, to stop the page.

Pages are hard limited to a maximum duration of two minutes. If a 'pSET 0' is not issued in two minutes, the page will timeout and return to normal mode.

Command format:

```
^pSET preset,time$
```

Response Strings:

```
^=pSET preset$
```

Where:

preset = The preset level. 0=Stop page, otherwise start page using the given preset.  
time = The amount of time (in milliseconds) to page. 0=Wait for a 'pSET 0' command.

Note: The 'time' parameter is not returned as part of the response string. This value is not saved and must be sent each time the 'pSET' command is sent.