

**RS-232 Control
of the
Advantage SPM522D**

advantage 

Introduction

This document contains technical information relating to computer control of the Biamp Advantage SPM522D Stereo Preamp/Mixer. This information is intended for advanced users - in particular for those who wish to develop their own computer programs to control the SPM522D. It is assumed that the reader is an experienced programmer and has some familiarity with standard programming practices, binary and hexadecimal numbers, the ASCII character set, asynchronous serial data communications, and RS-232 interfaces.

Hexadecimal, ASCII-Hex, and "Pseudo-Hex" Numbers

Throughout this document, hexadecimal numbers shall be represented by preceding the number with "0x". For example: the hexadecimal equivalent of the decimal number 255 is 0xFF.

Individual ASCII characters, except control characters, will be enclosed in single quotes. For example: the ASCII character 'A' has the hexadecimal value 0x41. The ASCII "carriage return" control character shall be represented as ↵. An ASCII code chart is included with this document for your convenience.

When an 8-bit binary data value is being transmitted over a serial data communications link, it is a common practice to transmit the byte as two "ASCII-hex" characters - one character represents the most significant nibble of the data value and the other character represents the least significant nibble (a nibble is 4-bits; half of a byte). Each ASCII-hex character is in the range of '0' thru '9' or 'A' thru 'F' (from the ASCII code chart, 0x30 thru 0x39 or 0x41 thru 0x46). For example, the *hexadecimal* equivalent of the decimal number 61 is 0x3D. To transmit this in an *ASCII-hex* format, first transmit the ASCII character '3' (whose hex value is 0x33), followed by the ASCII character 'D' (whose hex value is 0x44). This is the standard way to transmit an ASCII-hex value. In some cases, the lower case characters 'a' thru 'f' (0x61 thru 0x66) are accepted in addition to the upper case letters 'A' thru 'F'.

The Advantage SPM522D does not utilize standard ASCII-hex format. The SPM522D computer commands implement what I call "pseudo-hex" notation (also sometimes referred to as a "poor man's" ASCII-hex notation). Instead of representing the hexadecimal value using the ASCII characters '0' thru '9' and 'A', 'B', 'C', 'D', 'E', and 'F', the pseudo-hex format uses the ASCII characters '0' thru '9' and the characters ':', ';', '<', '=', '>', and '?'. As you can see by studying the ASCII code chart, a binary nibble may be converted to its equivalent pseudo-hex character by simply adding 0x30.

A binary/decimal/hexadecimal/pseudo-hex conversion chart is included at the end of this manual for your convenience.

Serial Interface - Data Communications Parameters

The SPM522D communicates through its serial interface at a data rate ("baud" rate) of 2400 bits per second with 8 data bits, 1 stop bit, and no parity. Although this is currently the only supported data rate, 9600 bits per second may be supported in the future - contact the factory for details. The SPM522D utilizes a subset of the standard 7-bit ASCII character set. The eighth data bit (bit 7 - the most significant bit) of each character transmitted by the computer should always be 0. The computer should not echo the characters it receives from the SPM522D.

Since the SPM522D only has a single-character input buffer for its incoming serial data, a form of flow control must be implemented by the computer in order to guarantee that no characters are lost. Neither hardware (DTR) nor XON/XOFF (also known as DC1/DC3 or control-S/control-Q) handshaking is supported by the SPM522D. However, each character which the SPM522D receives will be "echoed" back to the computer through the SPM522D's serial interface. Flow control may be implemented by the computer software by simply waiting for each character's echo from the SPM522D before transmitting the next character, since the SPM522D doesn't retrieve and echo an incoming character until it has finished processing the previous character.

Computer Control

The Advantage SPM522D has an RS-232-compatible serial interface which allows it to be controlled by a computer. The SPM522D offers two methods of computer control:

- Control Button Emulation. This method of computer control allows the computer to emulate Biamp's standard infrared remote control transmitter or wall-mount remote control panel. Using this method, the computer outputs ASCII characters which are equivalent to the characters generated by the remote control. Control Button Emulation is simple to perform, however, it only provides "one-way" control of the SPM522D - it allows the computer to send simple commands *to* the SPM522D, but it does not provide any mechanism for requesting status information *from* the SPM522D.
- Advanced Computer Control. This method of computer control provides advanced commands which allow "two-way" control of the SPM522D. Using Advanced Computer Control commands, the computer may request status information *from* the device as well as send commands *to* the device. Advanced Computer Control and Control Button Emulation are not mutually exclusive - both methods may be freely intermixed.

Control Button Emulation

Control Button Emulation is the simplest form of computer control of the Advantage SPM522D. This method of operation allows the computer to emulate the operation of a standard Biamp remote control transmitter.

The SPM522D provides two rear-panel remote control ports plus a remote translator port in addition to the internal front-panel infrared receiver. The signals from these four inputs are routed to a proprietary infrared decoder IC for translation before being sent on to the SPM522D's main microprocessor. When the decoder IC detects a proper bitstream which indicates that a button is being pressed on one of the remote control devices, it converts the button code into a standard ASCII character and transmits it to the main microprocessor's serial port. This is a shared serial port - the same serial port which is used for computer control. When the microprocessor receives a character through its serial port, it doesn't know (and doesn't care) whether that character came from a remote control device or whether it came from the computer. So, in order for the computer to emulate the operation of a remote control transmitter, all it has to do is transmit the same ASCII characters that the decoder IC transmits for each detected button press.

The following table summarizes the ASCII characters corresponding to each of the 40 remote control buttons supported by the SPM522D. These button codes are also summarized on the ASCII code chart provided at the end of this manual. The Advantage PC Control Software Package also indicates on its button definition screen the equivalent ASCII character for each of the 40 individual remote control buttons. The remote control buttons on the standard Biamp transmitter are numbered from left to right going from bottom to top with the lower left-hand button being button number 1.

Control Button Equivalent ASCII Codes (with factory default button definitions shown)

button 1	'B' (0x42)	Mic 1 volume down	button 21	'V' (0x56)	Main recall preset A
button 2	'C' (0x43)	Mic 2 volume down	button 22	'W' (0x57)	Main recall preset B
button 3	'D' (0x44)	Zone volume down	button 23	'X' (0x58)	Main recall preset C
button 4	'E' (0x45)	Main volume down	button 24	'Y' (0x59)	Main select source 5
button 5	'F' (0x46)	Mic 1 volume up	button 25	'Z' (0x5A)	Main select source 1
button 6	'G' (0x47)	Mic 2 volume up	button 26	'[' (0x5B)	Main select source 2
button 7	'H' (0x48)	Zone volume up	button 27	'\`' (0x5C)	Main select source 3
button 8	'I' (0x49)	Main volume up	button 28	']' (0x5D)	Main select source 4
button 9	'J' (0x4A)	Mic 1 toggle mute	button 29	'^' (0x5E)	NOP (no operation)
button 10	'K' (0x4B)	Mic 2 toggle mute	button 30	'_' (0x5F)	NOP (no operation)
button 11	'L' (0x4C)	Zone toggle mute	button 31	'`' (0x60)	NOP (no operation)
button 12	'M' (0x4D)	Main toggle mute	button 32	'b' (0x62)	NOP (no operation)
button 13	'N' (0x4E)	Zone recall preset E	button 33	'c' (0x63)	NOP (no operation)
button 14	'O' (0x4F)	Zone recall preset F	button 34	'd' (0x64)	NOP (no operation)
button 15	'P' (0x50)	Zone recall preset G	button 35	'e' (0x65)	NOP (no operation)
button 16	'Q' (0x51)	Zone select source 5	button 36	'f' (0x66)	NOP (no operation)
button 17	'R' (0x52)	Zone select source 1	button 37	'g' (0x67)	NOP (no operation)
button 18	'S' (0x53)	Zone select source 2	button 38	'h' (0x68)	NOP (no operation)
button 19	'T' (0x54)	Zone select source 3	button 39	'i' (0x69)	NOP (no operation)
button 20	'U' (0x55)	Zone select source 4	button 40	'j' (0x6A)	NOP (no operation)

When a remote control button is first pressed, the decoder IC sends the character which corresponds to that button to the SPM522D's main microprocessor. If the button is

pressed for longer than 110 milliseconds, the decoder IC will then send a "repeat code" to the main microprocessor and will continue to send additional repeat codes every 110 milliseconds for as long as the button is being pressed. The ASCII character corresponding to the repeat code is '@' (0x40).

Device Select Prefix Characters

When using Advanced Computer Control, up to eight SPM522Ds may be linked together and individually controlled by the computer (if each device is first assigned a unique device number). When using Control Button Emulation, a limited subset of device addressing may be performed, which allows individual control of up to four SPM522Ds (with device numbers 1 thru 4). This is accomplished by transmitting a device select prefix code immediately prior to each control button ASCII character code. The device select prefix code is inspected by each device to determine whether or not the device should react to the button code which immediately follows. (Note: do not transmit prefix codes prior to repeat codes.) If a button code is not immediately preceded by a device select prefix character, then all SPM522Ds in the system will react to that button code. The following table summarizes the ASCII characters to use for selecting various devices. This information is also summarized in the ASCII code chart provided at the end of this manual.

Device Select Prefix Codes

select device 1	'l' (0x6C)
select device 2	'm' (0x6D)
select devices 1 & 2	'n' (0x6E)
select device 3	'o' (0x6F)
select devices 1 & 3	'p' (0x70)
select devices 2 & 3	'q' (0x71)
select devices 1 & 2 & 3	'r' (0x72)
select device 4	's' (0x73)
select devices 1 & 4	't' (0x74)
select devices 2 & 4	'u' (0x75)
select devices 1 & 2 & 4	'v' (0x76)
select devices 3 & 4	'w' (0x77)
select devices 1 & 3 & 4	'x' (0x78)
select devices 2 & 3 & 4	'y' (0x79)
select devices 1 & 2 & 3 & 4	'z' (0x7A)

Note: Device select prefix codes are not supported (and will be ignored) by SPM522Ds with firmware versions earlier than July 2, 1995.

Advanced Computer Control

The Advanced Computer Control command set includes commands which allow the SPM522D to return information about the system to the computer, unlike Control Button Emulation which is basically a one-way control mechanism. The following list summarizes the commands available using Advanced Computer Control, including the ASCII command character associated with each command:

!	virtual-button	perform the specified button actions.
"	button-definition	define or retrieve the definition for the specified button.
#	source-settings	define or retrieve the stereo source settings (tone, balance).
\$	preset-definition	define or retrieve the definition for the specified preset.
%	do-misc	miscellaneous functions.
&	do-button	perform button actions for specified button.
'	do-preset-action	perform the specified preset action.
(do-volume-action	perform the specified volume or balance action.
(set-volume	set the volume for the specified fader.
)	do-tone-action	perform the specified bass or treble or balance action.
*	do-source-select	perform the specified stereo source selection action.
+	sleep-for-10-sec.	sleep for 10 seconds, ignoring all commands.
,	read-memory	retrieve data from non-volatile configuration memory.
-	write-memory	store data in non-volatile configuration memory.
.	set-factory-defaults	resets configuration parameters to factory defaults.
/	get-version	retrieve the model information and firmware version date.

Each Advanced Computer Control command requires at least two parameter bytes (four pseudo-hex nibbles) to be sent prior to the command character. Each command will be explained in detail on the following pages.

Some of the commands cause the SPM522D to return information to the computer. For each string of information returned to the computer, the SPM522D terminates the string by transmitting the ASCII carriage return character (0x0D - represented in this document as ↵). An internal dip-switch may be set to cause the SPM522D to append an ASCII line feed character (0x0A) after each carriage return. This dip-switch is labelled "Opt. X". When the Opt. X dip-switch is off, the option is disabled (no line feed); when the dip-switch is on, the option is enabled (a line feed follows each carriage return). The factory default setting is Opt. X disabled (no line feed).

As mentioned earlier, the Advantage SPM522D will echo all characters it receives, regardless of whether or not the characters are valid commands or parameters. Characters greater than 0x7F are reserved and should not be transmitted by the computer. The SPM522D utilizes a subset of the standard ASCII character set. The following characters have meaning to the SPM522D:

character	hexadecimal	operation
ASCII control characters	(0x00 - 0x1F)	no operation
ASCII SPACE character	(0x20)	no operation
! thru /	(0x21 - 0x2F)	Advanced Computer Control commands
0 thru ?	(0x30 - 0x3F)	pseudo-hex parameters for Advanced Computer Control commands
@	(0x40)	Control Button Emulation Repeat Code
A	(0x41)	no operation
B thru `	(0x42 - 0x60)	Control Button Emulation commands (buttons 01 - 31)
a	(0x61)	no operation
b thru j	(0x62 - 0x6A)	Control Button Emulation commands (buttons 32 - 40)
k thru z	(0x6B - 0x7A)	Control Button Emulation Device Select Prefix commands
{ thru DEL	(0x7B - 0x7F)	no operation
0x80 thru 0xFF	(0x80 - 0xFF)	RESERVED

An ASCII code chart showing all SPM522D commands and codes is provided later in this document for your convenience. One key point to observe is that the computer may feel free to transmit spaces, tabs, carriage returns, line feeds, or any other control characters *at any time* (even between two nibbles of a pseudo-hex parameter byte) without having *any* affect on the operation of the SPM522D. The SPM522D will simply echo them and then ignore them.

Device Type Bitmask and Device Number Bitmask

In a system which has more than one Advantage product connected to the computer, the Device Type Bitmask and Device Number Bitmask command parameters provide a mechanism for the computer to individually address a particular device (or a combination of devices). Every command in the Advanced Computer Control command set requires that a Device Type Bitmask and a Device Number Bitmask be transmitted as the last two parameter bytes before the computer transmits the command character itself. These two bitmask parameters bytes provide a device addressing capability to specify which of the devices in the system should execute the command. All devices which are not specifically addressed by these two bitmask values will ignore the command.

The Device Type Bitmask parameter byte supports up to eight distinct device types - one bit per device type. The eight device types are:

- 0x01 (bit 0) Biamp Advantage DRC 4+4 digital remote control
- 0x02 (bit 1) Biamp Advantage EQ28X digitally-controlled graphic equalizer
- 0x04 (bit 2) Biamp Advantage SPM522D stereo preamp/mixer
- 0x08 (bit 3) Biamp Advantage PMX84 programmable matrix switch
- 0x10 (bit 4) (reserved for future product)
- 0x20 (bit 5) (reserved for future product)
- 0x40 (bit 6) (reserved for future product)
- 0x80 (bit 7) (reserved for future product)

The Advantage SPM522D will only respond to Advanced Computer Control commands if bit 2 of the Device Type Bitmask parameter byte is a '1'. A command may be directed to more than one device type in the system by setting all of the corresponding bits in the Device Type Bitmask to '1's.

The Device Number Bitmask parameter byte supports up to eight distinct device numbers - one bit per device number. The eight device numbers are:

- 0x01 (bit 0) Select Device Number 1
- 0x02 (bit 1) Select Device Number 2
- 0x04 (bit 2) Select Device Number 3
- 0x08 (bit 3) Select Device Number 4
- 0x10 (bit 4) Select Device Number 5
- 0x20 (bit 5) Select Device Number 6
- 0x40 (bit 6) Select Device Number 7
- 0x80 (bit 7) Select Device Number 8

A particular Advantage SPM522D will only respond to Advanced Computer Control commands if the bit in the Device Number Bitmask parameter byte corresponding to its device number is a '1'. A command may be directed to more than one device number in the system by setting all of the corresponding bits in the Device Number Bitmask to '1's.

The Advanced Computer Control command set supports, in theory, up to sixty-four devices in a system - eight devices of each of the eight device types. In order for any particular device in the system to respond to an Advanced Computer Control command, the appropriate bit in both the Device Type and Device Number bitmask parameter bytes must be set to '1'.

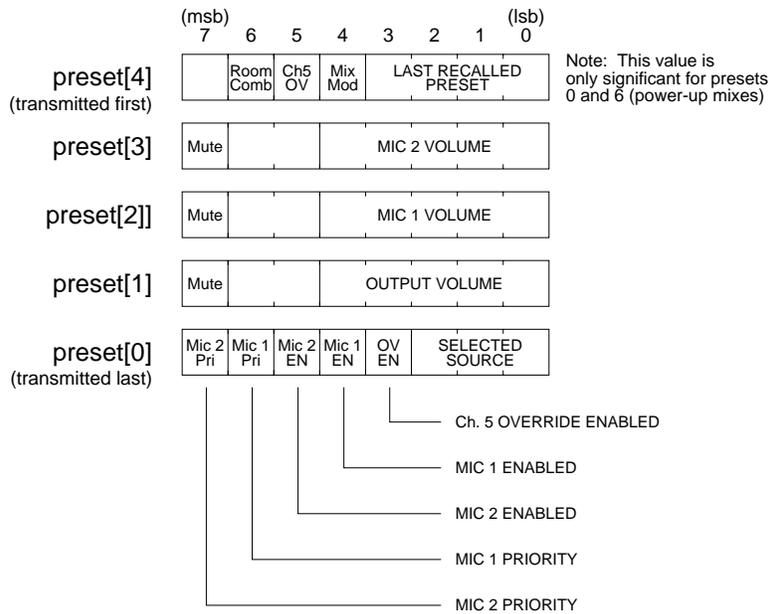
Preset Data Structure

The SPM522D firmware maintains data structures for 12 presets:

preset 0	Main power-up preset	preset 6	Zone power-up preset
preset 1	Main preset 'A'	preset 7	Zone preset 'E'
preset 2	Main preset 'B'	preset 8	Zone preset 'F'
preset 3	Main preset 'C'	preset 9	Zone preset 'G'
preset 4	Main preset 'D'	preset 10	Zone preset 'H'
preset 5	Main "current mix"	preset 11	Zone "current mix"

SPM522D Preset Data Structure

5-byte array with elements numbered preset[0] thru preset[4]



The volume levels are 5-bit values which range from 0x00 (step 0 - minimum volume) to 0x1F (step 31 - maximum volume). Bit 7 of the volume bytes indicates whether or not that particular fader is muted. If the fader is muted, the *actual* volume level sent to the VCA is 0x00, but the volume level specified in the 5 least significant bits of the volume byte indicates the volume level which will be restored when that fader becomes unmuted.

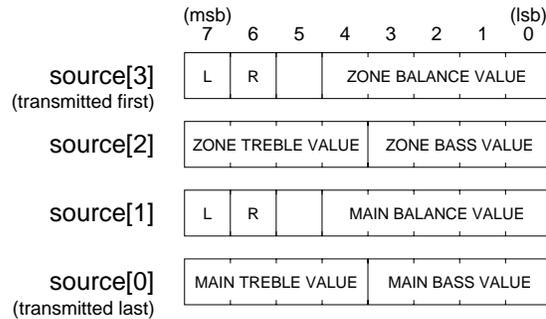
The fifth byte of the preset data structure (preset[4]) is used only for presets 0 and 6, the power-up presets. The SPM522D periodically stores its current mix status in its non-volatile configuration memory as preset 0 (Main) and preset 6 (Zone). Unless configured otherwise, the SPM522D will recall presets 0 and 6 upon power-up.

Stereo Source Data Structure

The SPM522D maintains data structures for 6 stereo sources. Source 0 is not used. Sources 1 thru 5 correspond to the five stereo inputs. These data structures contain the Main treble, Main bass, Main balance, Zone treble, Zone bass, and Zone balance information for each of the five stereo sources. Please note that the tone and balance information is directly associated with each stereo source and *not* associated with presets.

SPM522D Stereo Source Data Structure

4-byte array with elements numbered source[0] thru source[3]

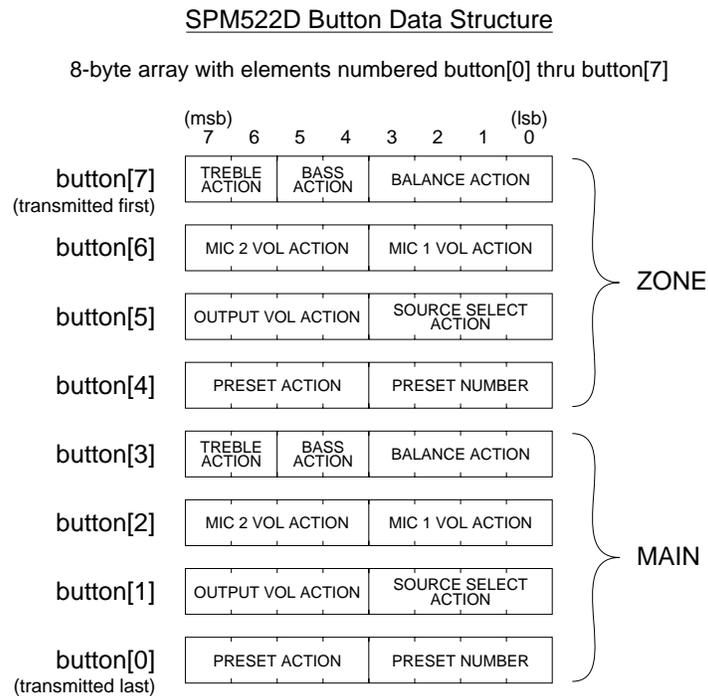


Bass and Treble values range from 0x00 (maximum cut -12 dB) to 0x0C (maximum boost +12 dB), with the value 0x06 corresponding to a "flat" response (no cut, no boost).

To best understand how the Balance control works, visualize two separate faders - one which affects the volume level of the left channel and one which affects the right channel. Each fader has a range of 0x08 (-24 dB) to 0x14 (0 dB). When the balance is centered, both faders are set at their maximum values of 0x14. To shift the balance to the right, decrease the left fader value. To shift the balance from the right back to the left, increase the left fader value until it once again reaches maximum (0x14) and then continue to shift to the left by decreasing the right fader value. At any given time, one or the other fader will be at its maximum value (and they will both be at maximum when the balance is centered). Since one value is always at maximum, there is no point in allocating space in the data structure to store two fader values to represent one balance setting. Instead, only one value is stored along with a flag bit to indicate whether the value is the left fader (bit 7) or the right fader value (bit 6). When the balance is centered, both fader values will be maximum (0x14) and it doesn't matter whether you specify the left fader value or the right fader value (the SPM522D will always report the value as if it is the left fader value).

Button Definition Data Structure

The SPM522D maintains data structures for 41 buttons (button 0 thru button 40 - button 0 is not used). Each button definition data structure consists of eight bytes - four bytes for Main button actions and four bytes for Zone button actions. The following illustration indicates which bit fields of which bytes are used for each type of action.



The bit patterns for the various button action codes are shown in the following tables:

Preset action codes:

0	0	0	0	NOP (no operation)
0	0	0	1	Recall Preset
0	0	1	0	Store Preset
0	0	1	1	Enter Combined Mode then Recall
0	1	0	0	(reserved)
0	1	0	1	Cancel Combined Mode then Recall
0	1	1	0	(reserved)
0	1	1	1	(reserved)
1	0	0	0	Toggle Combined Mode then Recall

Source Select action codes:

0	0	0	0	NOP (no operation)
0	0	0	1	select channel 1
0	0	1	0	select channel 2
0	0	1	1	select channel 3
0	1	0	0	select channel 4
0	1	0	1	select channel 5
0	1	1	0	(reserved)
0	1	1	1	toggle channel 5 Override
1	0	0	0	channel 5 Override
1	0	0	1	cancel channel 5 Override
1	0	1	0	enable mic 1
1	0	1	1	disable mic 1
1	1	0	0	toggle mic 1 enable
1	1	0	1	enable mic 2
1	1	1	0	disable mic 2
1	1	1	1	toggle mic 2 enable

Mic 1, Mic 2, and Output Volume action codes:

0	0	0	0	NOP (no operation)
0	0	0	1	volume down
0	0	1	0	volume up
0	0	1	1	toggle mute
0	1	0	0	mute
0	1	0	1	un-mute
0	1	1	0	minimum volume
0	1	1	1	maximum volume
1	0	0	0	(reserved)
1	0	0	1	set volume to predefined level

Balance action codes:

0	0	0	0	NOP (no operation)
1	1	0	1	shift to the Left
1	1	1	0	shift to the Right
1	1	1	1	set balance to center

Bass and Treble action codes:

0	0	NOP (no operation)
0	1	cut
1	0	boost
1	1	set to flat

Advanced Computer Control Command Notation

For the following descriptions of the Advanced Computer Control command set, the following conventions will be used:

Each ASCII character which represents a pseudo-hex nibble will be shown in *italics*, with the following letters representing certain types of parameters:

- a* a pseudo-hex nibble specifying an action code.
- b* one of the pseudo-hex nibbles specifying a button data structure. Also used to as a pseudo-hex nibble to specify a bass action or to select a memory bank.
- c* a pseudo-hex nibble specifying a checksum value.
- d* one of the pseudo-hex nibbles in the device number bitmask which indicates which device numbers should accept the following command.
- e* a pseudo-hex nibble specifying a memory address in the non-volatile configuration memory of the SPM522D (the ending address of a range of addresses).
- f* a pseudo-hex nibble specifying a fader selection.
- n* a pseudo-hex nibble specifying a preset number, button number, or stereo source number.
- o* a pseudo-hex nibble specifying a command option byte.
- p* one of the pseudo-hex nibbles specifying a preset data structure.
- r* a pseudo-hex nibble specifying the "room" selection (Main vs. Zone).
- s* a pseudo-hex nibble specifying a memory address in the non-volatile configuration memory of the SPM522D (the starting address of a range of addresses). Also used as one of the pseudo-hex nibbles specifying a stereo source data structure.
- t* one of the pseudo-hex nibbles specifying a treble action code.
- v* a pseudo-hex nibble specifying a volume level
- x* a pseudo-hex nibble specifying a generic data value.

! virtual-button

Description:

The virtual-button command causes the specified button actions to be immediately performed. The actions are defined using the Button Definition Data Structure. The button definition is not stored in the SPM522D's nonvolatile memory, nor is the button definition associated with a button number. Normally, when the SPM522D receives a control button command, it looks-up the control button definition for that button number in its control button definition table and then performs the actions defined for that button. The virtual-button command allows the computer to specify the actions to be performed without having the SPM522D look-up an entry in its control button definition table. This command essentially provides the SPM522D (when under computer control) with an unlimited number of button definitions.

Syntax of Command:

bbbbbbbbbbbbbbbb04dd!

where:

<i>bbbbbbbbbbbbbbbb</i>	=	Button Definition Data Structure (pseudo-hex)
<i>04</i>	=	SPM522D Device Type Bitmask
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>!</i>	=	virtual-button command character (0x21)

Syntax of Response:

(no response)

Example:

00455300004452000420!

This example causes SPM522D number 6 to immediately execute the following button actions:

Main - un-mute the output, select stereo source 2, mute Mic 1, mute Mic 2
Zone - un-mute the output, select stereo source 3, un-mute Mic 1, mute Mic 2.

Comments:

" define-button

Description:

The define-button command provides a new definition for the specified button number. The SPM522D will store this new button definition in its control button definition lookup table, replacing the definition that was there.

Syntax of Command:

```
bbbbbbbbbbbbbbbbnn04dd"
```

where:

<i>bbbbbbbbbbbbbbbb</i>	=	Button Definition Data Structure (pseudo-hex)
<i>nn</i>	=	Button Number + 128 (pseudo-hex)
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>"</i>	=	define-button command character (0x22)

Syntax of Response:

(no response)

Example:

```
0000000000000031940408"
```

This example causes SPM522D number 4 to redefine button number 20 (0x14) to enter Room Combining mode and recall preset mix 'A'.

Comments:

The decimal value 128 must be added to the Button number. This indicates to the SPM522D that this is the define-button form of the button-definition command.

" **get-button-definition** "

Description:

The `get-button-definition` command causes the SPM522D to return the definition of the specified button. The button definition will be returned in the Button Definition Data Structure format.

Syntax of Command:

`nn04dd"`

where:

<code>nn</code>	=	Button Number + 64 (pseudo-hex)
<code>04</code>	=	Device Type Bitmask (pseudo-hex)
<code>dd</code>	=	Device Number Bitmask (pseudo-hex)
<code>"</code>	=	<code>get-button-definition</code> command character (0x22)

Syntax of Response:

`bbbbbbbbbbbbbbbbbb↵`

where:

`bbbbbbbbbbbbbbbbbb` = Button Definition Data Structure (pseudo-hex)

Example:

command:	response:
<code>5;0401"</code>	<code>00020000000020000↵</code>

This example causes SPM522D number 1 to retrieve its button definition for control button number 27 (0x1B) and return it to the computer. In this example, the button definition was: increment the volume of Mic 1 in both the Main and Zone outputs.

Comments:

The decimal value 64 must be added to the Button number. This indicates to the SPM522D that this is the get-button form of the button-definition command.

define-source-settings

Description:

The `define-source-settings` command provides new treble, bass, and balance settings for the specified stereo source. This command includes the settings for both Main and Zone. The new settings are stored in the SPM522D's non-volatile configuration memory. If the specified source is the currently selected source for either the Main or Zone output, the new settings become effective immediately.

Syntax of Command:

```
ssssssnn04dd#
```

where:

<i>sssssss</i>	=	Stereo Source Data Structure
<i>nn</i>	=	Source number + 128 (pseudo-hex)
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>#</i>	=	define-source-settings command character (0x23)

Syntax of Response:

(no response)

Example:

```
94669457820401#
```

This example causes SPM522D number 1 to change its settings for stereo source 2 to:
Main : bass boosted one step and treble cut one step, balance centered
Zone : bass and treble flat, balance centered

Comments:

The decimal value 128 must be added to the Source number. This indicates to the SPM522D that this is the define-source-settings form of the source-settings command.

get-source-settings

Description:

The `get-source-settings` command causes the SPM522D to return the treble, bass, and balance settings for the specified stereo source. The settings will be returned in the Stereo Source Data Structure format. This includes the settings for both Main and Zone.

Syntax of Command:

nn04dd#

where:

<i>nn</i>	=	Source number + 64 (pseudo-hex)
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>#</i>	=	get-source-settings command character (0x23)

Syntax of Response:

sssssss↵

where:

sssssss = Stereo Source Data Structure (pseudo-hex)

Example:

command:	response:
420401#	94669466↵

This example causes SPM522D number 1 to retrieve its treble, bass, and balance settings for stereo source number 2 and return them to the computer. In this example, the source settings were:

Main : bass and treble flat, balance center
Zone : bass and treble flat, balance center

Comments:

The decimal value 64 must be added to the Source number. This indicates to the SPM522D that this is the get-source-settings form of the source-settings command.

\$ define-preset

Description:

The define-preset command provides settings for the specified preset. If either preset 5 (the Main "current mix") or preset 11 (the Zone "current mix") are specified, the new settings immediately go into effect. Otherwise, the new settings are simply stored in the SPM522D's non-volatile configuration memory for future recall.

Syntax of Command:

```
ppppppppppnn04dd$
```

where:

<i>pppppppppp</i>	=	Preset data structure
<i>nn</i>	=	Preset number + 128 (pseudo-hex)
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>\$</i>	=	define-preset command character (0x24)

Syntax of Response:

(no response)

Example:

```
00191?1<79830401$
```

This example causes SPM522D number 1 to change its settings for Main Preset 'C' to:
Stereo Source 1 selected
Mics 1 and 2 both enabled, with Mic 1 priority
Main output volume set to volume step 28
Mic 1 volume set to maximum (volume step 31)
Mic 2 volume set to volume step 25

Comments:

The decimal value 128 must be added to the Preset number. This indicates to the SPM522D that this is the define-preset form of the preset-definition command. Preset numbers 1, 2, 3, 4 correspond to Main Presets 'A', 'B', 'C', 'D'. Preset numbers 7, 8, 9, 10 correspond to Zone Presets 'E', 'F', 'G', 'H'.

\$ get-preset-definition

Description:

The get-preset-definition command causes the SPM522D to return the settings for the specified preset. The settings will be returned in the Preset Data Structure format. Preset number 5 refers to the Main "current mix" and preset number 11 refers to the Zone "current mix".

Syntax of Command:

nn04dd\$

where:

<i>nn</i>	=	Preset number + 64 (pseudo-hex)
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>\$</i>	=	get-preset-definition command character (0x24)

Syntax of Response:

pppppppppp↵

where:

pppppppppp = Preset Data Structure (pseudo-hex)

Example:

command:	response:
4;0401\$	18001?1?14↵

This example causes SPM522D number 1 to return its settings for the Zone "current mix" to the computer. In this example, the settings were: Stereo Source 4 selected, Ch.5 Override not allowed, Mic 1 enabled, Zone output volume and Mic 1 volume both set to maximum (step 31), Mic 2 volume set to minimum (step 0), the most recently recalled Zone preset was 'F' (8).

Comments:

The decimal value 64 must be added to the Preset number. This indicates to the SPM522D that this is the get-preset-definition form of the preset-definition command. Preset numbers 1, 2, 3, 4 correspond to Main Presets 'A', 'B', 'C', 'D'. Preset numbers 7, 8, 9, 10 correspond to Zone Presets 'E', 'F', 'G', 'H'.

% do-misc-ch5-override

Description:

The do-misc-ch5-override command allows the computer to allow or disallow the Channel 5 Override feature for the Main or Zone "current mix".

Syntax of Command:

xxaa04dd%

where:

<i>xx</i>	=	01 = Ch5 Override allowed. 00 = not allowed.
<i>aa</i>	=	Action code: 81 = Main. 82 = Zone.
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>%</i>	=	do-misc command character (0x25)

Syntax of Response:

(no response)

Example:

01810402%

This example causes SPM522D number 2 to allow Channel 5 Override for the Main output.

Comments:

% do-misc-mic-priority

Description:

The do-misc-mic-priority command allows the computer to specify the mic priority for the Main or Zone "current mix".

Syntax of Command:

xxaa04dd%

where:

<i>xx</i>	=	Priority: 00 = None, 01 = Mic 1, 02 = Mic 2.
<i>aa</i>	=	Action code: 83 = Main. 84 = Zone.
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>%</i>	=	do-misc command character (0x25)

Syntax of Response:

(no response)

Example:

02840401%

This example causes SPM522D number 1 to assign priority to Mic 2 for the Zone output "current mix".

Comments:

% do-misc-mic-enable

Description:

The do-misc-mic-enable command allows the computer to enable or disable ("assign" or "un-assign") Mic 1 or Mic 2 for the Main or Zone "current mix".

Syntax of Command:

xxaa04dd%

where:

<i>xx</i>	=	01 = Enable. 00 = Disable.
<i>aa</i>	=	Action code: 85 = Mic 1 Main. 86 = Mic 2 Main. 87 = Mic 1 Zone. 88 = Mic 2 Zone.
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>%</i>	=	do-misc command character (0x25)

Syntax of Response:

(no response)

Example:

01850401%01870401%

This example consists of two commands, causing SPM522D number 1 to enable Mic 1 for both the Main and Zone "current mixes".

Comments:

& do-button

Description:

The do-button command causes the SPM522D to look-up and perform the actions for the specified button number.

Syntax of Command:

nn04dd&

where:

<i>nn</i>	=	Button Number (pseudo-hex)
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
&	=	do-button command character (0x26)

Syntax of Response:

(no response)

Example:

28040?&

This example causes SPM522D numbers 1, 2, 3, and 4 to look-up and perform the actions defined in their respective control button definition tables for button number 40 (0x28).

Comments:

Button number must be 01 to 40 (0x01 to 0x28), otherwise the command will be ignored.

' do-preset-action

Description:

The do-preset-action command causes the SPM522D to perform the specified preset or room-combining action.

Syntax of Command:

an04dd'

where:

<i>a</i>	=	Action code (pseudo-hex nibble) 1 = Recall. 2 = Store. 3 = Combine then Recall. 5 = Cancel Combining then Recall. 8 = Toggle Combining then Recall.
<i>n</i>	=	Preset Number (pseudo-hex nibble) 1, 2, 3, 4 = Main Preset 'A', 'B', 'C', 'D' 7, 8, 9, : = Zone Preset 'E', 'F', 'G', 'H'
04	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
'	=	do-preset-action command character (0x27)

Syntax of Response:

(no response)

Example:

120401'

This example causes SPM522D number 1 to recall preset 'B'.

Comments:

(do-volume-action

Description:

The do-volume-action command causes the SPM522D to perform the specified volume action.

Syntax of Command:

*aa**ff**04**dd*(

where:

<i>aa</i>	=	Action code. 01 = Volume Down. 02 = Volume Up. 03 = Toggle Mute. 04 = Mute. 05 = Un-mute. 06 = Minimum Vol. 07 = Max Vol.
<i>ff</i>	=	Fader Selection Bitmask (pseudo-hex) 01 = Mic 1 Zone. 02 = Mic 2 Zone. 04 = Mic 1 Main. 08 = Mic 2 Main. 10 = Main Output. 20 = Zone Output.
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
(=	do-volume-action command character (0x28)

Syntax of Response:

(no response)

Example:

04100401(

This example causes SPM522D number 1 to Mute the Main output.

Comments:

Multiple faders may be addressed simultaneously by combining bit values in the fader selection bitmask. For example, **04300401(** mutes both the Main and the Zone outputs of SPM522D number 1. Another example: **010?0401(** turns down the volume (by one step) of both Mic 1 and Mic 2 in both the Main and Zone outputs.

(set-volume

Description:

The set-volume command causes the SPM522D to set the specified "fader" to a new volume level. The volume level is a 5-bit value which ranges from 0x00 (step 0 - minimum volume) to 0x1F (step 31 - maximum volume). Bit 7 of the volume byte indicates whether or not that particular fader should be muted. When bit 7 indicates that the fader should be muted (bit 7 = '1'), the 5 least significant bits (bits 0 - 4) indicate the volume level which will be restored by the SPM522D if that fader subsequently becomes un-muted.

Syntax of Command:

vv09ff04dd(

where:

<i>vv</i>	=	Volume level (pseudo-hex).
<i>09</i>	=	set-volume action code.
<i>ff</i>	=	Fader Selection Bitmask (pseudo-hex) 01 = Mic 1 Zone. 02 = Mic 2 Zone. 04 = Mic 1 Main. 08 = Mic 2 Main. 10 = Main Output. 20 = Zone Output.
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>(</i>	=	set-volume command character (0x28)

Syntax of Response:
(no response)

Example:

1709100401(

This example causes SPM522D number 1 to set its Main Output volume to approximately 75% of maximum (volume step 0x17, or 23 decimal).

Comments:

This command is not implemented in firmware versions prior to 23-Aug-95. Multiple faders may be addressed simultaneously by combining bit values in the fader selection bitmask. This is a variation of the do-volume-action command, and hence has the same command character as the do-volume-action command.

(do-balance-action

Description:

The do-balance-action command causes the SPM522D to perform the specified balance action.

Syntax of Command:

`aa rr 04 dd (`

where:

<code>aa</code>	=	Action code (2 pseudo-hex nibbles):
<code>0=</code>		Shift to the Left one step
<code>0></code>		Shift to the Right one step
<code>0?</code>		Reset the Balance to center
<code>rr</code>	=	Room (Main/Zone) Selection Bitmask (pseudo-hex)
		40 = Main. 80 = Zone. <0 = both Main and Zone.
<code>04</code>	=	Device Type Bitmask (pseudo-hex)
<code>dd</code>	=	Device Number Bitmask (pseudo-hex)
<code>(</code>	=	do-balance-action command character (0x28)

Syntax of Response:

(no response)

Example:

`0?400401(`

This example causes SPM522D number 1 to re-center the balance control *for the currently selected stereo source* for the Main output.

Comments:

This is a variation of the do-volume-action command, and hence has the same command character as the do-volume-action command.

) do-tone-action

Description:

The do-tone-action command causes the SPM522D to perform the specified tone actions.

Syntax of Command:

tbrr04dd)

where:

<i>t</i>	=	Treble Action code: 0 = NOP (no action). 1 = Cut. 2 = Boost. 3 = Reset Treble to Flat.
<i>b</i>	=	Bass Action code: 0 = NOP (no action). 1 = Cut. 2 = Boost. 3 = Reset Bass to Flat.
<i>rr</i>	=	Room (Main/Zone) Selection Bitmask (pseudo-hex) 40 = Main. 80 = Zone. <0 = both Main and Zone.
04	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
)	=	do-tone-action command character (0x29)

Syntax of Response:

(no response)

Example:

33<00401)

This example causes SPM522D number 1 to set both the Treble and Bass "flat" *for the currently selected stereo source* for the both the Main and Zone outputs.

Comments:

*** do-source-select**

Description:

The do-source-select command causes the SPM522D to perform the specified stereo source/channel selection.

Syntax of Command:

*aarr04dd**

where:

- aa* = Action code:
 - 01 = channel 1. 02 = channel 2. 03 = channel 3.
 - 04 = channel 4. 05 = channel 5.
 - 07 = toggle channel 5 Override.
 - 08 = activate channel 5 Override.
 - 09 = cancel channel 5 Override.
- rr* = Room (Main/Zone) Selection Bitmask (pseudo-hex)
 - 40 = Main. 80 = Zone. <0 = both Main and Zone.
- 04 = Device Type Bitmask (pseudo-hex)
- dd* = Device Number Bitmask (pseudo-hex)
- * = do-source-select command character (0x2A)

Syntax of Response:

(no response)

Example:

03800401*

This example causes SPM522D number 1 to select stereo source (channel) 3 for the Zone output.

Comments:

The channel 5 Override functions will be executed regardless of whether or not the channel 5 Override feature is enabled or disabled in the "current" mix.

+ sleep-for-10-seconds

Description:

The sleep-for-10-seconds command causes the SPM522D to "go to sleep" for 10 seconds, ignoring *all* data communications, including Control Button commands as well as Advanced Computer Control commands. During this time, characters received will be ignored and will not be echoed. Also, during this time, all remote control devices will be ignored. This command was implemented to facilitate remote computer control of the SPM522D via modem (with an auto-answer modem at the SPM522D). When an on-line session with a modem is finished and one modem or the other decides to disconnect or "hang up the phone", typically a spurt of unwanted spurious garbage characters occurs on the line. The SPM522D has no way of distinguishing between "garbage" characters and real characters. The last thing the computer should do before telling its modem to hang up is to issue the sleep-for-10-seconds command. This will allow plenty of time for the line to disconnect and the SPM522D will ignore all characters which it might receive during this hang-up process.

Syntax of Command:

ttdd+

where:

<i>tt</i>	=	Device Type Bitmask (pseudo-hex) 04 = SPM522D
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>+</i>	=	sleep-for-10-seconds command character (0x2B)

Syntax of Response:

(no response)

Example:

????+

This example causes *all* Advantage devices in the system to sleep for 10 seconds, ignoring all data communications.

Comments:

Note that the command character '+' is typically also the character used to return a Hayes-compatible modem to its command mode.

. read-memory

Description:

The read-memory command allows the computer to read the contents of one or more locations of the SPM522D's non-volatile configuration memory. The SPM522D has 512 bytes of configuration memory, which is arranged as two banks of 256 bytes each.

Syntax of Command:

*bb**ee**ss*04*dd*,

where:

<i>bb</i>	=	memory Bank select. 00 = bank 0. 01 = bank 1.
<i>ee</i>	=	Ending memory address (pseudo-hex)
<i>ss</i>	=	Starting memory address (pseudo-hex)
04	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
,	=	read-memory command character (0x2C)

Syntax of Response:

xx...(up to 256 data values)...↵

where:

xx = Data byte (pseudo-hex). Value in end addr is sent first.

Example:

command:

0006000401,

response:

03020;05020401↵

This example causes SPM522D number 1 to return the data values which are currently stored in locations 00 thru 06 of bank 0 of the configuration memory. In this example, memory location 00 contains the pseudo-hex value 01 and memory location 06 contains the pseudo-hex value 03.

Comments:

To read the contents of only one location, set both the starting and ending address to the same value (the desired address). The starting address should always be less than or equal to the ending address. Valid addresses for each bank are 0x00 thru 0xFF (pseudo-hex 00 thru ??). The data is sent in reverse order (last location first).

Contents of Bank 0:

0x00 - 0x3F Global Configuration Parameters
0x40 - 0x57 Stereo Source Tone and Balance Settings
0x58 - 0x93 Main and Zone Configuration Presets
0x94 - 0xB7 (reserved - currently unused)
0xB8 - 0xFF Button Definitions for Buttons 0 thru 8 (button 0 is unused)

Contents of Bank 1:

0x00 - 0xFF Button Definitions for Buttons 9 thru 40

- write-memory

Description:

The write-memory command allows the computer to store one or more data values in the SPM522D's non-volatile configuration memory beginning at a specified location. The SPM522D has 512 bytes of configuration memory, which is arranged as two banks of 256 bytes each. Each memory-write command may include up to 16 data values to be stored in a contiguous range of memory locations for the specified memory bank. This command provides the computer with a mechanism for setting or changing the global configuration parameters for the device. It also provides the computer with a method for totally messing things up.

Syntax of Command:

xx...(up to 16 data values)...ssoocc04dd-

where:

<i>xx..</i>	=	up to 16 data values (pseudo-hex). The values are sent in reverse order (highest memory address first)
<i>ss</i>	=	starting (lowest) memory address (pseudo-hex)
<i>oo</i>	=	options & byte count (pseudo-hex) bits 0, 1, 2, 3 = number of data values - 1 bit 4 - reserved (must be 0) bit 5 - memory Bank select. 0 = Bank 0. 1 = Bank 1. bit 6 - reserved (must be 0) bit 7 - 1 = activate new global config params now.
<i>cc</i>	=	computed checksum value (pseudo-hex)
<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>-</i>	=	write-memory command character (0x2D)

Syntax of Response:
(no response)

Example:

03020138823?0401-

This example causes SPM522D number 1 to store the values 0x01, 0x02, 0x03 in configuration memory bank 0 beginning at address 0x38 and causes the unit to retrieve and activate its global configuration parameters.

Comments:

In order to insure data integrity, a checksum value is included with this command. The checksum is computed by performing a simple 8-bit (modulo 256) sum of all of the data values plus the starting address value plus the option byte value and then performing a 1's complement of the sum. Note that the checksum calculation is performed on the actual 8-bit binary values, before any values are converted to pseudo-hex format for transmission. The SPM522D will perform the same calculation on the data values which it receives and compare its computed checksum to the received checksum. If the two values do not match, the SPM522D will ignore the entire command.

The SPM522D normally retrieves and activates its global configuration parameters from the non-volatile configuration memory only during its power-up sequence. If you use the write-memory command to change a global configuration parameter (which is why the command was provided), the SPM522D has to specifically be told to re-load and activate its configuration parameters after the new value is stored in its memory. Bit 7 of the options & byte count parameter may be set to tell the unit to automatically retrieve and activate its configuration parameters after the memory write operation is complete.

. set-factory-defaults

Description:

The set-factory-defaults command may be used to force the SPM522D to reset some or all of its configuration information to the factory default settings. The first two parameters for this command (< and >) are dummy parameters which were implemented to help prevent an accidental restoration of the factory defaults due to an error in data transmission.

Syntax of Command:

`<>oo04dd.`

where:

<	=	the ASCII character < (0x3C)
>	=	the ASCII character > (0x3E)
oo	=	Option byte (pseudo-hex): bit 0 - 1 = set Button Definition defaults bit 1 - 1 = set Preset defaults bit 2 - 1 = set Global Config Parameter defaults bit 3 - 1 = set Stereo Source Tone/Balance defaults * bit 7 - 1 = activate new Global Config Params now
04	=	Device Type Bitmask (pseudo-hex)
dd	=	Device Number Bitmask (pseudo-hex)
.	=	set-factory-defaults command character (0x2E)

Syntax of Response:

(no response)

Example:

`<>84040?.`

This example causes SPM522D numbers 1, 2, 3, and 4 to restore their global configuration parameters to the factory default settings and to retrieve and activate those new settings.

Comments:

Firmware versions prior to July 2, 1995 have a bug relating to the set Stereo Tone/Balance Defaults option - attempting to set bit 3 to activate this option will result in the SPM522D ignoring the entire command.

/ **get-version**

Description:

The get-version command causes the SPM522D to return its model identification code and firmware version to the computer. The firmware version number is simply the release date of the firmware, in a slightly modified standard American format of *mm:dd:yy*. These values are decimal digits, not pseudo-hex notation. For example, May 23, 1995 would be represented as **05:23:95**. The colon character (:) is used as a separator instead of the more conventional slash character, since the slash character is used as a computer command character by the SPM522D.

Syntax of Command:

04dd/

where:

<i>04</i>	=	Device Type Bitmask (pseudo-hex)
<i>dd</i>	=	Device Number Bitmask (pseudo-hex)
<i>/</i>	=	get-version command character (0x2F)

Syntax of Response:

01 *mm:dd:yy*␣

where:

01	=	initial model i.d. (0x30 followed by 0x31)
	=	ASCII space character (0x20)
<i>mm</i>	=	2-digit decimal month number
:	=	ASCII character : (0x3A)
<i>dd</i>	=	2-digit decimal day of the month
:	=	ASCII character : (0x3A)
<i>yy</i>	=	2-digit decimal year number

Example:

command:	response:
0401/	01 05:23:95 ␣

This example causes SPM522D number 1 to return its model identification code and firmware version.

Comments:

Advanced Computer Command Summary

<i>bbbbbbbbbbbbbb04dd!</i>	virtual-button
<i>bbbbbbbbbbbbbbnn04dd"</i>	define-button
<i>nn04dd"</i>	get-button-definition
<i>ssssssnn04dd#</i>	define-source-settings
<i>nn04dd#</i>	get-source-settings
<i>ppppppppnn04dd\$</i>	define-preset
<i>nn04dd\$</i>	get-preset-definition
<i>xxaa04dd%</i>	do-misc-ch5-override
<i>xxaa04dd%</i>	do-misc-mic-priority
<i>xxaa04dd%</i>	do-misc-mic-enable
<i>nn04dd&</i>	do-button
<i>an04dd'</i>	do-preset-action
<i>aaff04dd(</i>	do-volume-action
<i>vv09ff04dd(</i>	set-volume
<i>aarr04dd(</i>	do-balance-action
<i>tbrr04dd)</i>	do-tone-action
<i>aarr04dd*</i>	do-source-select
<i>04dd+</i>	sleep-for-10-seconds
<i>bbeess04dd,</i>	read-memory
<i>....xsssoocc04dd-</i>	write-memory
<i><>oo04dd.</i>	set-factory-defaults
<i>04dd/</i>	get-version

- a* a pseudo-hex nibble specifying an action code.
- b* one of the pseudo-hex nibbles specifying a button data structure. Also used to as a pseudo-hex nibble to specify a bass action or to select a memory bank.
- c* a pseudo-hex nibble specifying a checksum value.
- d* one of the pseudo-hex nibbles in the device number bitmask which indicates which device numbers should accept the following command.
- e* a pseudo-hex nibble specifying a memory address in the non-volatile configuration memory of the SPM522D (the ending address of a range of addresses).
- f* a pseudo-hex nibble specifying a fader selection.
- n* a pseudo-hex nibble specifying a presets number, button number, or stereo source number.
- o* a pseudo-hex nibble specifying a command option byte.
- p* one of the pseudo-hex nibbles specifying a presets data structure.
- r* a pseudo-hex nibble specifying the "room" selection (Main vs. Zone).
- s* a pseudo-hex nibble specifying a memory address in the non-volatile configuration memory of the SPM522D (the starting address of a range of addresses). Also used as one of the pseudo-hex nibbles specifying a stereo source data structure.
- t* one of the pseudo-hex nibbles specifying a treble action code.
- v* a pseudo-hex nibble specifying a volume level
- x* a pseudo-hex nibble specifying a generic data value.

ASCII Code Chart

with Decimal & Hexadecimal Equivalents and Advantage SPM522D Commands

000. 0x00	016. 0x10	032. 0x20	048. 0x30	064. 0x40	080. 0x50	096. 0x60	112. 0x70
NUL	DLE	(space)	0	@	P	`	p
			nibble 0x0	repeat code	button 15	button 31	select 1,3
001. 0x01	017. 0x11	033. 0x21	049. 0x31	065. 0x41	081. 0x51	097. 0x61	113. 0x71
SOH	DC1	!	1	A	Q	a	q
		virtual button	nibble 0x1		button 16		select 2,3
002. 0x02	018. 0x12	034. 0x22	050. 0x32	066. 0x42	082. 0x52	098. 0x62	114. 0x72
STX	DC2	"	2	B	R	b	r
		button defn	nibble 0x2	button 01	button 17	button 32	select 1,2,3
003. 0x03	019. 0x13	035. 0x23	051. 0x33	067. 0x43	083. 0x53	099. 0x63	115. 0x73
ETX	DC3	#	3	C	S	c	s
		source settings	nibble 0x3	button 02	button 18	button 33	select 4
004. 0x04	020. 0x14	036. 0x24	052. 0x34	068. 0x44	084. 0x54	100. 0x64	116. 0x74
EOT	DC4	\$	4	D	T	d	t
		preset defn	nibble 0x4	button 03	button 19	button 34	select 1,4
005. 0x05	021. 0x15	037. 0x25	053. 0x35	069. 0x45	085. 0x55	101. 0x65	117. 0x75
ENQ	NAK	%	5	E	U	e	u
		do-misc	nibble 0x5	button 04	button 20	button 35	select 2,4
006. 0x06	022. 0x16	038. 0x26	054. 0x36	070. 0x46	086. 0x56	102. 0x66	118. 0x76
ACK	SYN	&	6	F	V	f	v
		do-button	nibble 0x6	button 05	button 21	button 36	select 1,2,4
007. 0x07	023. 0x17	039. 0x27	055. 0x37	071. 0x47	087. 0x57	103. 0x67	119. 0x77
BEL	ETB	'	7	G	W	g	w
		do preset	nibble 0x7	button 06	button 22	button 37	select 3,4
008. 0x08	024. 0x18	040. 0x28	056. 0x38	072. 0x48	088. 0x58	104. 0x68	120. 0x78
BS	CAN	(8	H	X	h	x
		do volume	nibble 0x8	button 07	button 23	button 38	select 1,3,4
009. 0x09	025. 0x19	041. 0x29	057. 0x39	073. 0x49	089. 0x59	105. 0x69	121. 0x79
HT	EM)	9	I	Y	i	y
		do tone	nibble 0x9	button 08	button 24	button 39	select 2,3,4
010. 0x0A	026. 0x1A	042. 0x2A	058. 0x3A	074. 0x4A	090. 0x5A	106. 0x6A	122. 0x7A
LF	SUB	*	:	J	Z	j	z
		do-source	nibble 0xA	button 09	button 25	button 40	select 1,2,3,4
011. 0x0B	027. 0x1B	043. 0x2B	059. 0x3B	075. 0x4B	091. 0x5B	107. 0x6B	123. 0x7B
VT	ESC	+	;	K	[k	{
		sleep 10 sec.	nibble 0xB	button 10	button 26	select none	
012. 0x0C	028. 0x1C	044. 0x2C	060. 0x3C	076. 0x4C	092. 0x5C	108. 0x6C	124. 0x7C
FF	FS	,	<	L	\	l	
		read memory	nibble 0xC	button 11	button 27	select 1	
013. 0x0D	029. 0x1D	045. 0x2D	061. 0x3D	077. 0x4D	093. 0x5D	109. 0x6D	125. 0x7D
CR	GS	-	=	M]	m	}
		write memory	nibble 0xD	button 12	button 28	select 2	
014. 0x0E	030. 0x1E	046. 0x2E	062. 0x3E	078. 0x4E	094. 0x5E	110. 0x6E	126. 0x7E
SO	RS	.	>	N	^	n	~
		set defaults	nibble 0xE	button 13	button 29	select 1,2	
015. 0x0F	031. 0x1F	047. 0x2F	063. 0x3F	079. 0x4F	095. 0x5F	111. 0x6F	127. 0x7F
SI	US	/	?	O	_	o	DEL
		get version	nibble 0xF	button 14	button 30	select 3	

HEXADECIMAL CONVERSION CHART

binary	decimal	hex	pseudo												
0000 0000	0.	0x00	00	0100 0000	64.	0x40	40	1000 0000	128.	0x80	80	1100 0000	192.	0xc0	<0
0000 0001	1.	0x01	01	0100 0001	65.	0x41	41	1000 0001	129.	0x81	81	1100 0001	193.	0xc1	<1
0000 0010	2.	0x02	02	0100 0010	66.	0x42	42	1000 0010	130.	0x82	82	1100 0010	194.	0xc2	<2
0000 0011	3.	0x03	03	0100 0011	67.	0x43	43	1000 0011	131.	0x83	83	1100 0011	195.	0xc3	<3
0000 0100	4.	0x04	04	0100 0100	68.	0x44	44	1000 0100	132.	0x84	84	1100 0100	196.	0xc4	<4
0000 0101	5.	0x05	05	0100 0101	69.	0x45	45	1000 0101	133.	0x85	85	1100 0101	197.	0xc5	<5
0000 0110	6.	0x06	06	0100 0110	70.	0x46	46	1000 0110	134.	0x86	86	1100 0110	198.	0xc6	<6
0000 0111	7.	0x07	07	0100 0111	71.	0x47	47	1000 0111	135.	0x87	87	1100 0111	199.	0xc7	<7
0000 1000	8.	0x08	08	0100 1000	72.	0x48	48	1000 1000	136.	0x88	88	1100 1000	200.	0xc8	<8
0000 1001	9.	0x09	09	0100 1001	73.	0x49	49	1000 1001	137.	0x89	89	1100 1001	201.	0xc9	<9
0000 1010	10.	0x0a	0:	0100 1010	74.	0x4a	4:	1000 1010	138.	0x8a	8:	1100 1010	202.	0xca	<:
0000 1011	11.	0x0b	0;	0100 1011	75.	0x4b	4;	1000 1011	139.	0x8b	8;	1100 1011	203.	0xcb	<;
0000 1100	12.	0x0c	0<	0100 1100	76.	0x4c	4<	1000 1100	140.	0x8c	8<	1100 1100	204.	0xcc	<<
0000 1101	13.	0x0d	0=	0100 1101	77.	0x4d	4=	1000 1101	141.	0x8d	8=	1100 1101	205.	0xcd	<=
0000 1110	14.	0x0e	0>	0100 1110	78.	0x4e	4>	1000 1110	142.	0x8e	8>	1100 1110	206.	0xce	<>
0000 1111	15.	0x0f	0?	0100 1111	79.	0x4f	4?	1000 1111	143.	0x8f	8?	1100 1111	207.	0xcf	<?
0001 0000	16.	0x10	10	0101 0000	80.	0x50	50	1001 0000	144.	0x90	90	1101 0000	208.	0xd0	=0
0001 0001	17.	0x11	11	0101 0001	81.	0x51	51	1001 0001	145.	0x91	91	1101 0001	209.	0xd1	=1
0001 0010	18.	0x12	12	0101 0010	82.	0x52	52	1001 0010	146.	0x92	92	1101 0010	210.	0xd2	=2
0001 0011	19.	0x13	13	0101 0011	83.	0x53	53	1001 0011	147.	0x93	93	1101 0011	211.	0xd3	=3
0001 0100	20.	0x14	14	0101 0100	84.	0x54	54	1001 0100	148.	0x94	94	1101 0100	212.	0xd4	=4
0001 0101	21.	0x15	15	0101 0101	85.	0x55	55	1001 0101	149.	0x95	95	1101 0101	213.	0xd5	=5
0001 0110	22.	0x16	16	0101 0110	86.	0x56	56	1001 0110	150.	0x96	96	1101 0110	214.	0xd6	=6
0001 0111	23.	0x17	17	0101 0111	87.	0x57	57	1001 0111	151.	0x97	97	1101 0111	215.	0xd7	=7
0001 1000	24.	0x18	18	0101 1000	88.	0x58	58	1001 1000	152.	0x98	98	1101 1000	216.	0xd8	=8
0001 1001	25.	0x19	19	0101 1001	89.	0x59	59	1001 1001	153.	0x99	99	1101 1001	217.	0xd9	=9
0001 1010	26.	0x1a	1:	0101 1010	90.	0x5a	5:	1001 1010	154.	0x9a	9:	1101 1010	218.	0xda	=:
0001 1011	27.	0x1b	1;	0101 1011	91.	0x5b	5;	1001 1011	155.	0x9b	9;	1101 1011	219.	0xdb	=;
0001 1100	28.	0x1c	1<	0101 1100	92.	0x5c	5<	1001 1100	156.	0x9c	9<	1101 1100	220.	0xdc	=<
0001 1101	29.	0x1d	1=	0101 1101	93.	0x5d	5=	1001 1101	157.	0x9d	9=	1101 1101	221.	0xdd	==
0001 1110	30.	0x1e	1>	0101 1110	94.	0x5e	5>	1001 1110	158.	0x9e	9>	1101 1110	222.	0xde	=>
0001 1111	31.	0x1f	1?	0101 1111	95.	0x5f	5?	1001 1111	159.	0x9f	9?	1101 1111	223.	0xdf	=?
0010 0000	32.	0x20	20	0110 0000	96.	0x60	60	1010 0000	160.	0xa0	:0	1110 0000	224.	0xe0	>0
0010 0001	33.	0x21	21	0110 0001	97.	0x61	61	1010 0001	161.	0xa1	:1	1110 0001	225.	0xe1	>1
0010 0010	34.	0x22	22	0110 0010	98.	0x62	62	1010 0010	162.	0xa2	:2	1110 0010	226.	0xe2	>2
0010 0011	35.	0x23	23	0110 0011	99.	0x63	63	1010 0011	163.	0xa3	:3	1110 0011	227.	0xe3	>3
0010 0100	36.	0x24	24	0110 0100	100.	0x64	64	1010 0100	164.	0xa4	:4	1110 0100	228.	0xe4	>4
0010 0101	37.	0x25	25	0110 0101	101.	0x65	65	1010 0101	165.	0xa5	:5	1110 0101	229.	0xe5	>5
0010 0110	38.	0x26	26	0110 0110	102.	0x66	66	1010 0110	166.	0xa6	:6	1110 0110	230.	0xe6	>6
0010 0111	39.	0x27	27	0110 0111	103.	0x67	67	1010 0111	167.	0xa7	:7	1110 0111	231.	0xe7	>7
0010 1000	40.	0x28	28	0110 1000	104.	0x68	68	1010 1000	168.	0xa8	:8	1110 1000	232.	0xe8	>8
0010 1001	41.	0x29	29	0110 1001	105.	0x69	69	1010 1001	169.	0xa9	:9	1110 1001	233.	0xe9	>9
0010 1010	42.	0x2a	2:	0110 1010	106.	0x6a	6:	1010 1010	170.	0xaa	::	1110 1010	234.	0xea	>:
0010 1011	43.	0x2b	2;	0110 1011	107.	0x6b	6;	1010 1011	171.	0xab	;;	1110 1011	235.	0xeb	>;
0010 1100	44.	0x2c	2<	0110 1100	108.	0x6c	6<	1010 1100	172.	0xac	:<	1110 1100	236.	0xec	><
0010 1101	45.	0x2d	2=	0110 1101	109.	0x6d	6=	1010 1101	173.	0xad	:=	1110 1101	237.	0xed	>=
0010 1110	46.	0x2e	2>	0110 1110	110.	0x6e	6>	1010 1110	174.	0xae	:>	1110 1110	238.	0xee	>>
0010 1111	47.	0x2f	2?	0110 1111	111.	0x6f	6?	1010 1111	175.	0xaf	:?	1110 1111	239.	0xef	>?
0011 0000	48.	0x30	30	0111 0000	112.	0x70	70	1011 0000	176.	0xb0	:0	1111 0000	240.	0xf0	?0
0011 0001	49.	0x31	31	0111 0001	113.	0x71	71	1011 0001	177.	0xb1	:1	1111 0001	241.	0xf1	?1
0011 0010	50.	0x32	32	0111 0010	114.	0x72	72	1011 0010	178.	0xb2	:2	1111 0010	242.	0xf2	?2
0011 0011	51.	0x33	33	0111 0011	115.	0x73	73	1011 0011	179.	0xb3	:3	1111 0011	243.	0xf3	?3
0011 0100	52.	0x34	34	0111 0100	116.	0x74	74	1011 0100	180.	0xb4	:4	1111 0100	244.	0xf4	?4
0011 0101	53.	0x35	35	0111 0101	117.	0x75	75	1011 0101	181.	0xb5	:5	1111 0101	245.	0xf5	?5
0011 0110	54.	0x36	36	0111 0110	118.	0x76	76	1011 0110	182.	0xb6	:6	1111 0110	246.	0xf6	?6
0011 0111	55.	0x37	37	0111 0111	119.	0x77	77	1011 0111	183.	0xb7	:7	1111 0111	247.	0xf7	?7
0011 1000	56.	0x38	38	0111 1000	120.	0x78	78	1011 1000	184.	0xb8	:8	1111 1000	248.	0xf8	?8
0011 1001	57.	0x39	39	0111 1001	121.	0x79	79	1011 1001	185.	0xb9	:9	1111 1001	249.	0xf9	?9
0011 1010	58.	0x3a	3:	0111 1010	122.	0x7a	7:	1011 1010	186.	0xba	;;	1111 1010	250.	0xfa	?:
0011 1011	59.	0x3b	3;	0111 1011	123.	0x7b	7;	1011 1011	187.	0xbb	;;	1111 1011	251.	0xfb	?;
0011 1100	60.	0x3c	3<	0111 1100	124.	0x7c	7<	1011 1100	188.	0xbc	:<	1111 1100	252.	0xfc	?<
0011 1101	61.	0x3d	3=	0111 1101	125.	0x7d	7=	1011 1101	189.	0xbd	:=	1111 1101	253.	0xfd	?=
0011 1110	62.	0x3e	3>	0111 1110	126.	0x7e	7>	1011 1110	190.	0xbe	:>	1111 1110	254.	0xfe	?>
0011 1111	63.	0x3f	3?	0111 1111	127.	0x7f	7?	1011 1111	191.	0xbf	:?	1111 1111	255.	0xff	??