Symetrix

460 Presentation Audio Mixer

CONTROL PROTOCOL

Revision 2.1 - For 460 version 1.08

460 Presentation Audio Mixer
Command Protocol

Printed in the United States of America

Symetrix, Inc.
6408 216th St. SW
Mountlake Terrace WA 98043
USA

Tel: (425) 778-7728
Fax: (425) 778-7727
Web: http://www.symetrixaudio.com
E-mail: tech@symetrixaudio.com

Contents

Introduction

About this document

The purpose of this document is to provide a technical understanding of the Symetrix 460 Presentation Audio Mixer Command Protocol. It will define and illustrate the data string structure used to communicate with the 460 via a serial RS-232 or RS-485.

Conventions used in this document

A dollar sign "$" preceding a set of two alphanumeric characters denotes a hex value. All other number values should be considered decimal values. Ex., "**$A0**" represents the decimal value of "160".

Getting Started

Data string format

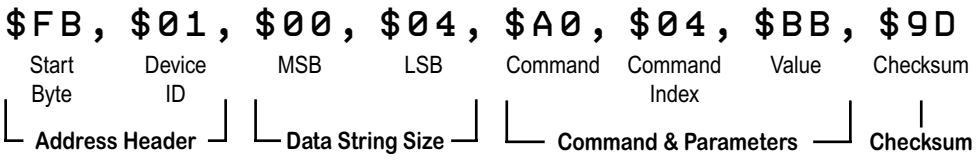
We can say, for purposes of illustration, that the data string is made up of three parts; the address header, the MSB and LSB byte count. The address header consists of the address escape byte, <$FB>, and the number of the addressed unit, <$ADDR>. The <$FB> signals the beginning of a data string as well as an escape for the end of one. Anywhere a <$FB> byte is present in the outgoing data stream, it must be escaped with another <$FB> byte to indicate that the byte is to be treated as a data value and not the address mark. This additional escape byte is not factored into the checksum. The <$ADDR> is the unit ID number (**$01–$FA** or 1-250; an address of **$00** or 0 is used for global or "broadcast" type commands). The MSB and LSB byte count indicate the number of bytes to follow (not including any <$FB> escape bytes). The MSB and LSB together are treated as a 16 bit unsigned quantity, the MSB being the upper byte and the LSB the lower. The MSB will always be zero unless the command stream is more than 255 bytes long.

Here is another way to look at it:

PART	LENGTH	DESCRIPTION
Address Header	2 bytes	byte 1: Escape byte <$FB> byte 2: Device Address <$ADDR> (**$01–$FA** or 1-250; 0 = global)
Data String Size	2 bytes	byte 1: MSB = normally zero (see above paragraph) byte 2: LSB = Command (1 byte) + Parameters (nn bytes) + Checksum (1 byte)
Command & Parameters	1 byte nn bytes	For example, **$A0** (**Send Parameter Data**) Format and size varies by command type (See **Parameter Indexes**, pgs. 13-14)
Checksum	1 byte	See **Checksum** on page 5

Data string construction

An example command string: Set IN 1 -> BUS 1 Gain to maximum output level using **$A0 Send Parameter Data**.



An example return status string: No error.

Return codes

Returned status codes (PA-422 ANNEX A defined):
- **$00**: no error
- **$01**: invalid data
- **$02**: invalid command code
- **$03**: device locked
- **$04**: device not locked
- **$05**: channel(s) muted
- **$06**: channel(s) not muted

460 specific status codes:
- **$07**: checksum error
- **$10**: flash write error
- **$11**: invalid S Record
- **$12**: invalid password
- **$13**: command failed

Device Type Code:
- **$46**: 460 Multimedia Mixer

Manufacturers' Code:
- **$38**: Symetrix

Checksum

The checksum is the 2's complement of the LSB byte of the (32 bit internal) checksum. To compute the checksum, ignore the initial <**$FB**> and <**$ADDR**> bytes of the string so you are left with the MSB, LSB, command, and parameter data. Add the remaining bytes. Here is a simple formula:

sum = sum AND **$FF** :make sure the sum is less than **$100** (256 in decimal)
checksum = **$100** - sum :take the two's complement of sum

Example:
Data String with out checksum: **$FB, $01, $00, $04, $A0, $04, $BB** (251, 1, 0, 4, 160, 4, 187 in decimal)
Remove FB and address bytes: **$00, $4, $A0, $4, $BB** (0, 4, 160, 4, 187 in decimal)
Add remaining bytes: **$163** (355 in decimal)
Ignore all but the bottom byte: **$63** (99 in decimal)
Two's compliment: **$9D** (157 in decimal)
Data String with checksum: **$FB, $1, $0, $4, $A0, $4, $BB, $9D** (251, 1, 0, 4, 160, 4, 187, 157 in decimal)

Commands

$82 Load Program - Loads a program into the edit buffer
Example: Load program 1
<$FB, $01, $00, $03, $82, $01, $7A>

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$03		(LSB) including command and checksum
$82		command
$nn		memory number (1-8)
$nn		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$02	(LSB) including status and checksum
	$nn	returned status
	$nn	checksum (of all returned bytes)

$83 Set Program Pointer - deferred load program
Note: Receiving a global load program command (address mark + unit address of 0) will load the program number set in this command. See Command: **Global Load Program** on page 13.

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$03		(LSB) including command and checksum
$83		command
$nn		program pointer (0 = off, 1-8 = program memory)
$nn		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow,
	$02	(LSB) including status and checksum
	$nn	returned status
	$nn	checksum (of all returned bytes)

$85 Lock Device
Lock states are controlled by setting bits in the lock word that is 16 bits long (for future expandibility). Locks for the front panel and the remote control are handled separately. A set bit enables the appropriate lock.

For the remote interface and rear panel inputs:

BIT	Function
0	Disables program stores
1	Changes to the edit buffer are disabled except for output level control
2	Changes to the edit buffer output level parameters are disabled
3	Program loads from RS-232/RS-485 are disabled

For the front panel:

BIT	Function
0	Disables program stores
1	Changes to the edit buffer are disabled except for output level control
2	All parameter changes are locked
3	Program loads are disabled
4	Changes are disabled from the external A/D inputs
5	Program loads from the external program pins are disabled

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$16		(LSB) including command and checksum
$85		command
$nn		password (16 bytes, 0 filled) If no password was stored in the
...		device then this field is ignored.
$nn		(MSB) rear/remote lock level
$nn		(LSB) rear/remote lock level
$nn		(MSB) front lock level
$nn		(LSB) front lock level
$nn		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$02	(LSB) including status and checksum
	$nn	returned status
	$nn	checksum (of all returned bytes)

$86 Unlock Device

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$22		(LSB) including command and checksum
$86		command
$nn		password (16 bytes, 0 filled) If no password was stored in the
...		device then this field is ignored.
$78		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$02	(LSB) including status and checksum
	$nn	returned status
	$nn	checksum (of all returned bytes)

$87 Mute Output(s)

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$03		(LSB) including command and checksum
$87		command
$nn		output channel (0 = all, 1 = stereo output 1, 2 = stereo output 2)
$nn		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$02	(LSB) including status and checksum
	$nn	returned status
	$nn	checksum (of all returned bytes)

$88 Unmute Output(s)

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$03		(LSB) including command and checksum
$88		command
$nn		output channel (0 = all, 1 = stereo output 1, 2 = stereo output 2)
$nn		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$02	(LSB) including status and checksum
	$nn	returned status
	$nn	checksum (of all returned bytes)

$89 Mute All Outputs

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$02		(LSB) including command and checksum
$89		command
$75		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$02	(LSB) including status and checksum
	$nn	returned status
	$nn	checksum (of all returned bytes)

$8A Unmute All Outputs

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$02		(LSB) including command and checksum
$8A		command
$74		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$02	(LSB) including status and checksum
	$nn	returned status
	$nn	checksum (of all returned bytes)

$93 Save Program

Note: Two consecutive saves of program 255 will initialize all programs and global parameters.

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$03		(LSB) including command and checksum
$93		command
$nn		save current edit buffer to user program (1-8)
$nn		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$02	(LSB) including status and checksum
	$nn	returned status
	$nn	checksum (of all returned bytes)

$94 Set System Data - Set up system parameters
Note: Any field filled with zeros (nulls) uses the current value.

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) bytes to follow
$50		(LSB) including command and checksum
$94		command
$nn		16 character old password
...		
$nn		16 character new password
...		If 16 null's, then password not altered
$nn		16 character device name
...		If 16 null's, then name not altered
$nn		inputs 1 & 2 operating mode (0 = separate mono, 1 = combined stereo)
...		in revisions 1.08 and above, the upper 4 bits have additional meaning: bit 4 clear = input 3 is separate mono bit 4 set = input 3 is combined stereo bit 5 clear = input 4 is separate mono bit 5 set = input 4 is combined stereo bit 6 clear = input 5 is separate mono bit 6 set = input 5 is combined stereo bit 7 clear = input 6 is separate mono bit 7 set = input 6 is combined stereo
$nn		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$02	(LSB) including status and checksum
	$nn	returned status
	$nn	checksum (of all returned bytes)
	$nn	checksum (of all returned bytes)

$A0 Send Parameter Data
Example 1: Turn IN 1 -> BUS 1 gain to maximum level.
<$FB, $01, $00, $04, $A0, $04, $BB, $9D>
Note: The above example changes the gain for IN 1 only if the INPUTS 1&2 MODE is set to "SEPARATE MONO". If INPUTS 1&2 MODE is set to "COMBINED STEREO," the gain is controlled by the parameter definitions for "IN 2."

Example 2: Turns IN 3 -> BUS 1 gain to maximum level.
<$FB, $01, $00, $04, $A0, $0C, $BB, $95>
Note: Although it is possible to read from the edit buffer and any stored program using command **$20**, writing into the program storage EEPROM requires that the edit buffer be updated, then a save to program, command **$93**, be executed.

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$nn		(MSB) number of bytes to follow
$nn		(LSB) including command, data and checksum
$A0		command
$nn		starting parameter index number
$nn		parameter data starting with given index
$nn		last parameter byte
$nn		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$02	(LSB) including status and checksum
	$nn	returned status

$A1 Send Program Name

Note: Although it is possible to read from the edit buffer and any stored program using command **$20**, writing into the program storage EEPROM requires that the edit buffer be updated, then a save to program, command **$93**, be executed.

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$nn		(MSB) number of bytes to follow
$nn		(LSB) including command, data and checksum
$A1		command
$nn		first program name character
...		
$nn		last program name character
$nn		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$02	(LSB) including status and checksum
	$nn	returned status
	$nn	checksum (of all returned bytes)

$00 Get Operational Status

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$02		(LSB) including command and checksum
$00		command
$FE		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$05	(LSB) including status and checksum
	$nn	current program pointer (0 = not active)
	$nn	1 = edit buffer modified
	$nn	last error status (0 if none)
	$nn	returned status
	$nn	checksum (of all returned bytes)

$02 Get Device Type and ID

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$02		(LSB) including command and checksum
$02		command
$FC		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$04	(LSB) including status and checksum
	$DT	device type
	$ID	manufacturer's code
	$nn	returned status
	$nn	checksum (of all returned bytes)

$12 Get software statistics

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$02		(LSB) including command and checksum
$12		command
$EC		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$44	(LSB) including status and checksum
	$nn	16 byte password
	...	
	$nn	16 byte device name
	...	
	$nn	revision number *10
	$nn	day
	$nn	month
	$nn	year (20<nn>)
	$nn	reserved
	$nn	(MSB) rear/remote lock level
	$nn	(LSB) rear/remote lock level
	$nn	(MSB) front lock level
	$nn	(LSB) front lock level
	$nn	inputs 1 & 2 operating mode (0 = separate mono, 1 = combined stereo)
	...	in revisions 1.08 and above, the upper 4 bits have additional meaning: bit 4 clear = input 3 is separate mono bit 4 set = input 3 is combined stereo bit 5 clear = input 4 is separate mono bit 5 set = input 4 is combined stereo bit 6 clear = input 5 is separate mono bit 6 set = input 5 is combined stereo bit 7 clear = input 6 is separate mono bit 7 set = input 6 is combined stereo
	$nn	return status
	$nn	checksum (of all returned bytes)

$20 Receive Parameter Data

Note: Executing this command resets the 'EBCHANGED_LOCAL' (bit 1) flag in the real-time status command so that the front panel will no longer display the program as "dirty" or changed.

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$05		(LSB) including command and checksum
$20		command
$nn		buffer to read from, 0 = edit, 1-8 = programs
$nn		starting parameter index number
$nn		number of parameters to read if = $FF then read all parameters up to the last available
$nn		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$nn	(MSB) number of bytes to follow
	$nn	(LSB) including status and checksum
	$nn	data: These are ordered according to the data ordering in the Parameter Definition section. (See pgs. 14-15)
	$nn	returned status

$21 Read Program Name

Note: Although it isn't possible to write into any arbitrary program store in the EEPROM, it is possible to read data from any arbitrary program or edit buffer location.

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$03		(LSB) including command and checksum
$21		command
$nn		program to read from (0 = edit buffer, 1-8 = user programs)
$nn		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$nn	(MSB) number of bytes to follow
	$nn	(LSB) including status and checksum
	$nn	name string
	...	Note: name string is not necessarily '\0' terminated
	$nn	returned status
	$nn	checksum (of all returned bytes)

$22 Get Real-time Status

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$ADDR		unit address (1-250)
$00		(MSB) number of bytes to follow
$02		(LSB) including command and checksum
$22		command: get real-time data
$DC		checksum (of all sent bytes after addressing)
	$ADDR	unit address (1-250)
	$DT	device type
	$ID	manufacturer's code
	$00	(MSB) number of bytes to follow
	$27	(LSB) including status and checksum

Note: Level values are 0.5dB/step below 0dBFS with a value of zero indicating 0dBFS. Likewise, gate and compression attenuation is also 0.5dB/step below 0dBFS.

$nn	IN 1 input level
$nn	IN 1 gate expansion
$nn	IN 2 input level
$nn	IN 2 gate expansion
$nn	IN 3L input level
$nn	IN 3R input level
$nn	IN 4L input level
$nn	IN 4R input level
$nn	IN 5L input level
$nn	IN 5R input level
$nn	IN 6L input level
$nn	IN 6R input level
$nn	OUT 1L output level
$nn	OUT 1R output level
$nn	OUT 1 output compression
$nn	OUT 2L output level
$nn	OUT 2R output level
$nn	OUT 2 output compression
$nn	Reserved
$nn	Reserved

Map of overload status bits. The bit is set if in overload, cleared after 3 seconds of inactivity.

	BIT	MODULE
$nn	0	internal stereo bus 1 (either L or R)
	1	internal stereo bus 2 (either L or R)
	2	output 1 parametric (either L or R)
	3	output 2 parametric (either L or R)
	4	output 1 hold state (0 = following signal, 1 = holding compression)
	5	output 2 hold state (0 = following signal, 1 = holding compression)
$nn		current program

Note: The top bit will be set if edit buffer has been changed by the front panel. The bit will be reset upon the next reading of the changed parameter(s) using **$20 Receive Parameter Data**.

$nn	edit buffer changed flag Bit 0 set: edit buffer different from stored program Bit 1 set: changed since last status read
$nn	System settings changed flag Bit 0 set: changed since the last **$12 Get Software Statistics** command.
$nn	Mute status: Bits 0-2 set indicate output channels 1L/1R, 2L/2R respectively are muted
$nn	return status
$nn	checksum (of all returned bytes)

Global Load Program - This is a special “broadcast” type command that will load the program that has been set up with command **$83 Set Program Pointer** into all units that have received a non-zero preset load value. Any unit with a zero value for the preset load will ignore this command.

SEND	RECEIVE	DESCRIPTION
$FB		address mark
$00		load program pointed to by program pointer

Parameter Definition

Parameter Indexes

Note: To simplify the controlling GUI it is possible to bypass some individual modules by setting bit 7 in one of their parameters. This alleviates the need to remember their values when adjusting the bypass state. The variable used for the bypass function varies with the module (see below):

1. The Parametric module uses **Boost/Cut Gain**
2. The Gate/Compression module uses **Ratio**

In addition, the bus gains and output gains can be muted by setting bit 7 in their respective gain parameters.

PARAMETER INDEX	FUNCTION	MAPPING TABLE
Channel 1 Input		
$00	High Pass In/Out	0: Out, 1: In
$01	Low Pass In/Out	0: Out, 1: In
$02	Gate Threshold	Thresh1
$03	Gate Depth	Thresh1
$04	Bus 1 Gain	Gain2
$05	Bus 2 Gain	Gain2

PARAMETER INDEX	FUNCTION	MAPPING TABLE
Channel 2 Input		
$06	High Pass In/Out	0: Out, 1: In
$07	Low Pass In/Out	0: Out, 1: In
$08	Gate Threshold	Thresh1
$09	Gate Depth	Thresh1
$0A	Bus 1 Gain	Gain2
$0B	Bus 2 Gain	Gain2
Channel 3 Input (3L/3R)		
$0C	Bus 1 Gain	Gain2
$0D	Bus 2 Gain	Gain2
Channel 4 Input (4L/4R)		
$0E	Bus 1 Gain	Gain2
$0F	Bus 2 Gain	Gain2
Channel 5 Input (5L/5R)		
$10	Bus 1 Gain	Gain2
$11	Bus 2 Gain	Gain2
Channel 6 Input (6L/6R)		
$12	Bus 1 Gain	Gain2
$13	Bus 2 Gain	Gain2
Channel 1 Output (1L/1R)		
$14	Low Shelving Boost/Cut	Gain1
$15	Mid Eq Boost/Cut	Gain1
$16	Mid Eq Frequency	Freq1
$17	Mid Eq Band Width	Bw1
$18	Hi Shelving Boost/Cut	Gain1
$19	Compressor Mode	0: Bypassed, 1: Limit, 2: Compressor, 3: AGC
$1A	Compressor Threshold	Thresh1
$1B	AGC Autorelease Threshold	Thresh1
$1C	Compressor Ratio	Ratio1
$1D	Compressor Makeup Gain	Gain1 + 12 dB, set to 0 for limiter mode
$1E	Output Delay	Delay1
$1F	Stereo Output Gain	Gain2
$20	Mono/Stereo	0: mono, 1: stereo
Channel 2 Output (2L/2R)		
$21	Low Shelving Boost/Cut	Gain1
$22	Mid Eq Boost/Cut	Gain1
$23	Mid Eq Frequency	Freq1
$24	Mid Eq Band Width	Bw1
$25	Hi Shelving Boost/Cut	Gain1
$26	Compressor Mode	0: Bypassed, 1: Limit, 2: Compressor, 3: AGC
$27	Compressor Threshold	Thresh1
$28	AGC Autorelease Threshold	Thresh1
$29	Compressor Ratio	Ratio1
$2A	Compressor Makeup Gain	Gain1 + 12, set to 0 for limiter mode
$2B	Output Delay	Delay1
$2C	Stereo Output Gain	Gain2
$2D	Mono/Stereo	0: mono, 1: stereo
Test Oscillator Output		
$2E	Test oscillator type	0: sine, 1: pink, 2: white
$2F	Test oscillator	freq (sine only)
$30	Test oscillator bus 1L	attenuation only, 0-100dB
$31	Test oscillator bus 1R	attenuation only, 0-100dB
$32	Test oscillator bus 2L	attenuation only, 0-100dB
$33	Test oscillator bus 2R	attenuation only, 0-100dB
Program Name		
$34...$43	Program name	16 characters, null filled

PARAMETER INDEX	FUNCTION	MAPPING TABLE
ADC Controllers		
$44	ADC1 control destination	0: off, 1: Out 1 Gain, 2: Out 2 Gain, 3: Out 1/2 Gain
$45	ADC2 control destination	0: off, 1: Out 1 Gain, 2: Out 2 Gain, 3: Out 1/2 Gain
For when Channels 3-6 are in SEPARATE MONO mode		
$46	3R Bus 1 Gain	Gain2
$47	3R Bus 2 Gain	Gain2
$48	4R Bus 1 Gain	Gain2
$49	4R Bus 2 Gain	Gain2
$4A	5R Bus 1 Gain	Gain2
$4B	5R Bus 2 Gain	Gain2
$4C	6R Bus 1 Gain	Gain2
$4D	6R Bus 2 Gain	Gain2

PARAMETER INDEX	FUNCTION	MAPPING TABLE
ADC Controllers		
$44	ADC1 control destination	0: off, 1: Out 1 Gain, 2: Out 2 Gain, 3: Out 1/2 Gain
$45	ADC2 control destination	0: off, 1: Out 1 Gain, 2: Out 2 Gain, 3: Out 1/2 Gain
For when Channels 3-6 are in SEPARATE MONO mode		
$46	3R Bus 1 Gain	Gain2
$47	3R Bus 2 Gain	Gain2
$48	4R Bus 1 Gain	Gain2
$49	4R Bus 2 Gain	Gain2
$4A	5R Bus 1 Gain	Gain2
$4B	5R Bus 2 Gain	Gain2
$4C	6R Bus 1 Gain	Gain2
$4D	6R Bus 2 Gain	Gain2

Parameter Encoding Tables

Ratio1: 1.0 to 6.0 in 0.2 steps, 6.0 to 20.0 in 1.0 steps. Encoded from 0 to 39, where 0 = 1.0 and 39 = 20.0

$00	1.0	$0A	3.0	$14	5.0	$1E	11.0
$01	1.2	$0B	3.2	$15	5.2	$1F	12.0
$02	1.4	$0C	3.4	$16	5.4	$20	13.0
$03	1.6	$0D	3.6	$17	5.6	$21	14.0
$04	1.8	$0E	3.8	$18	5.8	$22	15.0
$05	2.0	$0F	4.0	$19	6.0	$23	16.0
$06	2.2	$10	4.2	$1A	7.0	$24	17.0
$07	2.4	$11	4.4	$1B	8.0	$25	18.0
$08	2.6	$12	4.6	$1C	9.0	$26	19.0
$09	2.8	$13	4.8	$1D	10.0	$27	20.0

Gain1: +/- 12dB in 0.5dB steps, encoded from 0 to 48, where 0 = -12.0 dB, 24 = 0 dB and 48 = +12.0 dB. For the case of makeup gain, the range is shifted by internally adding 24dB of gain to the given value.

$00	-12.0dB	$0D	- 5.5dB	$19	+ 0.5dB	$25	+ 6.5dB
$01	-11.5dB	$0E	- 5.0dB	$1A	+ 1.0dB	$26	+ 7.0dB
$02	-11.0dB	$0F	- 4.5dB	$1B	+ 1.5dB	$27	+ 7.5dB
$03	-10.5dB	$10	- 4.0dB	$1C	+ 2.0dB	$28	+ 8.0dB
$04	-10.0dB	$11	- 3.5dB	$1D	+ 2.5dB	$29	+ 8.5dB
$05	- 9.5dB	$12	- 3.0dB	$1E	+ 3.0dB	$2A	+ 9.0dB
$06	- 9.0dB	$13	- 2.5dB	$1F	+ 3.5dB	$2B	+ 9.5dB
$07	- 8.5dB	$14	- 2.0dB	$20	+ 4.0dB	$2C	+10.0dB
$08	- 8.0dB	$15	- 1.5dB	$21	+ 4.5dB	$2D	+10.5dB
$09	- 7.5dB	$16	- 1.0dB	$22	+ 5.0dB	$2E	+11.0dB
$0A	- 7.0dB	$17	- 0.5dB	$23	+ 5.5dB	$2F	+11.5dB
$0B	- 6.5dB	$18	0.0dB	$24	+ 6.0dB	$30	+12.0dB
$0C	- 6.0dB						

Gain2: OFF to -90dB to -60dB in 1dB steps. -60dB to +18dB in 0.5dB steps. Encoded from 0 to 187, where 0 = OFF.

$00	OFF	$2F	−52.0dB	$5E	−28.5dB	$8D	− 5.0dB
$01	−90.0dB	$30	−51.5dB	$5F	−28.0dB	$8E	− 4.5dB
$02	−89.0dB	$31	−51.0dB	$60	−27.5dB	$8F	− 4.0dB
$03	−88.0dB	$32	−50.5dB	$61	−27.0dB	$90	− 3.5dB
$04	−87.0dB	$33	−50.0dB	$62	−26.5dB	$91	− 3.0dB
$05	−86.0dB	$34	−49.5dB	$63	−26.0dB	$92	− 2.5dB
$06	−85.0dB	$35	−49.0dB	$64	−25.5dB	$93	− 2.0dB
$07	−84.0dB	$36	−48.5dB	$65	−25.0dB	$94	− 1.5dB
$08	−83.0dB	$37	−48.0dB	$66	−24.5dB	$95	− 1.0dB
$09	−82.0dB	$38	−47.5dB	$67	−24.0dB	$96	− 0.5dB
$0A	−81.0dB	$39	−47.0dB	$68	−23.5dB	$97	0.0dB
$0B	−80.0dB	$3A	−46.5dB	$69	−23.0dB	$98	+ 0.5dB
$0C	−79.0dB	$3B	−46.0dB	$6A	−22.5dB	$99	+ 1.0dB
$0D	−78.0dB	$3C	−45.5dB	$6B	−22.0dB	$9A	+ 1.5dB
$0E	−77.0dB	$3D	−45.0dB	$6C	−21.5dB	$9B	+ 2.0dB
$0F	−76.0dB	$3E	−44.5dB	$6D	−21.0dB	$9C	+ 2.5dB
$10	−75.0dB	$3F	−44.0dB	$6E	−20.5dB	$9D	+ 3.0dB
$11	−74.0dB	$40	−43.5dB	$6F	−20.0dB	$9E	+ 3.5dB
$12	−73.0dB	$41	−43.0dB	$70	−19.5dB	$9F	+ 4.0dB
$13	−72.0dB	$42	−42.5dB	$71	−19.0dB	$A0	+ 4.5dB
$14	−71.0dB	$43	−42.0dB	$72	−18.5dB	$A1	+ 5.0dB
$15	−70.0dB	$44	−41.5dB	$73	−18.0dB	$A2	+ 5.5dB
$16	−69.0dB	$45	−41.0dB	$74	−17.5dB	$A3	+ 6.0dB
$17	−68.0dB	$46	−40.5dB	$75	−17.0dB	$A4	+ 6.5dB
$18	−67.0dB	$47	−40.0dB	$76	−16.5dB	$A5	+ 7.0dB
$19	−66.0dB	$48	−39.5dB	$77	−16.0dB	$A6	+ 7.5dB
$1A	−65.0dB	$49	−39.0dB	$78	−15.5dB	$A7	+ 8.0dB
$1B	−64.0dB	$4A	−38.5dB	$79	−15.0dB	$A8	+ 8.5dB
$1C	−63.0dB	$4B	−38.0dB	$7A	−14.5dB	$A9	+ 9.0dB
$1D	−62.0dB	$4C	−37.5dB	$7B	−14.0dB	$AA	+ 9.5dB
$1E	−61.0dB	$4D	−37.0dB	$7C	−13.5dB	$AB	+10.0dB
$1F	−60.0dB	$4E	−36.5dB	$7D	−13.0dB	$AC	+10.5dB
$20	−59.5dB	$4F	−36.0dB	$7E	−12.5dB	$AD	+11.0dB
$21	−59.0dB	$50	−35.5dB	$7F	−12.0dB	$AE	+11.5dB
$22	−58.5dB	$51	−35.0dB	$80	−11.5dB	$AF	+12.0dB
$23	−58.0dB	$52	−34.5dB	$81	−11.0dB	$B0	+12.5dB
$24	−57.5dB	$53	−34.0dB	$82	−10.5dB	$B1	+13.0dB
$25	−57.0dB	$54	−33.5dB	$83	−10.0dB	$B2	+13.5dB
$26	−56.5dB	$55	−33.0dB	$84	− 9.5dB	$B3	+14.0dB
$27	−56.0dB	$56	−32.5dB	$85	− 9.0dB	$B4	+14.5dB
$28	−55.5dB	$57	−32.0dB	$86	− 8.5dB	$B5	+15.0dB
$29	−55.0dB	$58	−31.5dB	$87	− 8.0dB	$B6	+15.5dB
$2A	−54.5dB	$59	−31.0dB	$88	− 7.5dB	$B7	+16.0dB
$2B	−54.0dB	$5A	−30.5dB	$89	− 7.0dB	$B8	+16.5dB
$2C	−53.5dB	$5B	−30.0dB	$8A	− 6.5dB	$B9	+17.0dB
$2D	−53.0dB	$5C	−29.5dB	$8B	− 6.0dB	$BA	+17.5dB
$2E	−52.5dB	$5D	−29.0dB	$8C	− 5.5dB	$BB	+18.0dB

Thresh1: -100dB to 0dBFS in 0.5dB steps. Encoded from 0 to 200, where 0 = -100dB.

$00	OFF	$32	–75.5dB	$64	–50.5dB	$96	–25.5dB
$01	–100.0dB	$33	–75.0dB	$65	–50.0dB	$97	–25.0dB
$02	–99.5dB	$34	–74.5dB	$66	–49.5dB	$98	–24.5dB
$03	–99.0dB	$35	–74.0dB	$67	–49.0dB	$99	–24.0dB
$04	–98.5dB	$36	–73.5dB	$68	–48.5dB	$9A	–23.5dB
$05	–98.0dB	$37	–73.0dB	$69	–48.0dB	$9B	–23.0dB
$06	–97.5dB	$38	–72.5dB	$6A	–47.5dB	$9C	–22.5dB
$07	–97.0dB	$39	–72.0dB	$6B	–47.0dB	$9D	–22.0dB
$08	–96.5dB	$3A	–71.5dB	$6C	–46.5dB	$9E	–20.5dB
$09	–96.0dB	$3B	–71.0dB	$6D	–46.0dB	$9F	–20.0dB
$0A	–95.5dB	$3C	–70.5dB	$6E	–45.5dB	$A0	–19.5dB
$0B	–95.0dB	$3D	–70.0dB	$6F	–45.0dB	$A1	–19.0dB
$0C	–94.5dB	$3E	–69.5dB	$70	–44.5dB	$A2	–18.5dB
$0D	–94.0dB	$3F	–69.0dB	$71	–44.0dB	$A3	–18.0dB
$0E	–93.5dB	$40	–68.5dB	$72	–43.5dB	$A4	–17.5dB
$0F	–93.0dB	$41	–68.0dB	$73	–43.0dB	$A5	–17.0dB
$10	–92.5dB	$42	–67.5dB	$74	–42.5dB	$A6	–16.5dB
$11	–92.0dB	$43	–67.0dB	$75	–42.0dB	$A7	–16.0dB
$12	–91.5dB	$44	–66.5dB	$76	–41.5dB	$A8	–15.5dB
$13	–91.0dB	$45	–66.0dB	$77	–41.0dB	$A9	–15.0dB
$14	–90.5dB	$46	–65.5dB	$78	–40.5dB	$AA	–14.5dB
$15	–90.0dB	$47	–65.0dB	$79	–40.0dB	$AB	–14.0dB
$16	–89.5dB	$48	–64.5dB	$7A	–39.5dB	$AC	–13.5dB
$17	–89.0dB	$49	–64.0dB	$7B	–39.0dB	$AD	–13.0dB
$18	–88.5dB	$4A	–63.5dB	$7C	–38.5dB	$AE	–12.5dB
$19	–88.0dB	$4B	–63.0dB	$7D	–38.0dB	$AF	–12.0dB
$1A	–87.5dB	$4C	–62.5dB	$7E	–37.5dB	$B0	–11.5dB
$1B	–87.0dB	$4D	–62.0dB	$7F	–37.0dB	$B1	–11.0dB
$1C	–86.5dB	$4E	–61.5dB	$80	–36.5dB	$B2	–10.5dB
$1D	–86.0dB	$4F	–61.0dB	$81	–36.0dB	$B3	–10.0dB
$1E	–85.5dB	$50	–60.5dB	$82	–35.5dB	$B4	– 9.5dB
$1F	–85.0dB	$51	–60.0dB	$83	–35.0dB	$B5	– 9.0dB
$20	–84.5dB	$52	–59.5dB	$84	–34.5dB	$B6	– 8.5dB
$21	–84.0dB	$53	–59.0dB	$85	–34.0dB	$B7	– 8.0dB
$22	–83.5dB	$54	–58.5dB	$86	–33.5dB	$B8	– 7.5dB
$23	–83.0dB	$55	–58.0dB	$87	–33.0dB	$B9	– 7.0dB
$24	–82.5dB	$56	–57.5dB	$88	–32.5dB	$BA	– 6.5dB
$25	–82.0dB	$57	–57.0dB	$89	–32.0dB	$BB	– 6.0dB
$26	–81.5dB	$58	–56.5dB	$8A	–31.5dB	$BC	– 5.5dB
$27	–81.0dB	$59	–56.0dB	$8B	–31.0dB	$BD	– 5.0dB
$28	–80.5dB	$5A	–55.5dB	$8C	–30.5dB	$BE	– 4.5dB
$29	–80.0dB	$5B	–55.0dB	$8D	–30.0dB	$BF	– 4.0dB
$2A	–79.5dB	$5C	–54.5dB	$8E	–29.5dB	$C0	– 3.5dB
$2B	–79.0dB	$5D	–54.0dB	$8F	–29.0dB	$C1	– 3.0dB
$2C	–78.5dB	$5E	–53.5dB	$90	–28.5dB	$C2	– 2.5dB
$2D	–78.0dB	$5F	–53.0dB	$91	–28.0dB	$C3	– 2.0dB
$2E	–77.5dB	$60	–52.5dB	$92	–26.5dB	$C4	– 1.5dB
$2F	–77.0dB	$61	–52.0dB	$93	–27.0dB	$C5	– 1.0dB
$30	–76.5dB	$62	–51.5dB	$94	–27.5dB	$C6	– 0.5dB
$31	–76.0dB	$63	–51.0dB	$95	–26.0dB	$C7	0.0dB

Freq1: 16Hz to 19.6kHz in 1/20 octave steps. Encoded from 0 to 205, where 0 = 16Hz and 205 = 19.6kHz.

$00	16.176Hz	$34	98.073Hz	$68	594.604Hz	$9C	3.605kHz
$01	16.746Hz	$35	101.532Hz	$69	615.572Hz	$9D	3.732kHz
$02	17.337Hz	$36	105.112Hz	$6A	637.280Hz	$9E	3.863kHz
$03	17.948Hz	$37	108.819Hz	$6B	659.754Hz	$9F	4.000kHz
$04	18.581Hz	$38	112.656Hz	$6C	683.020Hz	$A0	4.141kHz
$05	19.237Hz	$39	116.629Hz	$6D	707.107Hz	$A1	4.287kHz
$06	19.915Hz	$3A	120.742Hz	$6E	732.043Hz	$A2	4.438kHz
$07	20.617Hz	$3B	125.000Hz	$6F	757.858Hz	$A3	4.594kHz
$08	21.344Hz	$3C	129.408Hz	$70	784.584Hz	$A4	4.756kHz
$09	22.097Hz	$3D	133.972Hz	$71	812.252Hz	$A5	4.924kHz
$0A	22.876Hz	$3E	138.696Hz	$72	840.896Hz	$A6	5.098kHz
$0B	23.683Hz	$3F	143.587Hz	$73	870.551Hz	$A7	5.278kHz
$0C	24.518Hz	$40	148.651Hz	$74	901.250Hz	$A8	5.464kHz
$0D	25.383Hz	$41	153.893Hz	$75	933.033Hz	$A9	5.656kHz
$0E	26.278Hz	$42	159.320Hz	$76	965.936Hz	$AA	5.856kHz
$0F	27.205Hz	$43	164.938Hz	$77	1.000kHz	$AB	6.062kHz
$10	28.164Hz	$44	170.755Hz	$78	1.035kHz	$AC	6.276kHz
$11	29.157Hz	$45	176.777Hz	$79	1.071kHz	$AD	6.498kHz
$12	30.186Hz	$46	183.001Hz	$7A	1.109kHz	$AE	6.727kHz
$13	31.250Hz	$47	189.465Hz	$7B	1.148kHz	$AF	6.964kHz
$14	32.352Hz	$48	196.146Hz	$7C	1.189kHz	$B0	7.210kHz
$15	33.493Hz	$49	203.063Hz	$7D	1.231kHz	$B1	7.464kHz
$16	34.674Hz	$4A	210.224Hz	$7E	1.274kHz	$B2	7.727kHz
$17	35.897Hz	$4B	217.638Hz	$7F	1.319kHz	$B3	8.000kHz
$18	37.163Hz	$4C	225.313Hz	$80	1.366kHz	$B4	8.282kHz
$19	38.473Hz	$4D	233.258Hz	$81	1.414kHz	$B5	8.574kHz
$1A	39.830Hz	$4E	241.484Hz	$82	1.464kHz	$B6	8.876kHz
$1B	41.235Hz	$4F	250.000Hz	$83	1.515kHz	$B7	9.189kHz
$1C	42.689Hz	$50	258.816Hz	$84	1.569kHz	$B8	9.513kHz
$1D	44.194Hz	$51	267.943Hz	$85	1.624kHz	$B9	9.849kHz
$1E	45.753Hz	$52	277.392Hz	$86	1.681kHz	$BA	10.196kHz
$1F	47.366Hz	$53	287.175Hz	$87	1.741kHz	$BB	10.556kHz
$20	49.037Hz	$54	297.302Hz	$88	1.802kHz	$BC	10.928kHz
$21	50.766Hz	$55	307.786Hz	$89	1.866kHz	$BD	11.313kHz
$22	52.566Hz	$56	318.640Hz	$8A	1.931kHz	$BE	11.712kHz
$23	54.409Hz	$57	329.877Hz	$8B	2.000kHz	$BF	12.125kHz
$24	56.328Hz	$58	341.510Hz	$8C	2.070kHz	$C0	12.553kHz
$25	58.315Hz	$59	353.553Hz	$8D	2.143kHz	$C1	12.996kHz
$26	60.371Hz	$5A	366.021Hz	$8E	2.219kHz	$C2	13.454kHz
$27	62.500Hz	$5B	378.929Hz	$8F	2.297kHz	$C3	13.928kHz
$28	64.704Hz	$5C	392.292Hz	$90	2.378kHz	$C4	14.420kHz
$29	66.986Hz	$5D	406.126Hz	$91	2.462kHz	$C5	14.928kHz
$2A	69.348Hz	$5E	420.448Hz	$92	2.549kHz	$C6	15.454kHz
$2B	71.794Hz	$5F	435.275Hz	$93	2.639kHz	$C7	16.000kHz
$2C	74.325Hz	$60	450.625Hz	$94	2.732kHz	$C8	16.564kHz
$2D	76.947Hz	$61	466.517Hz	$95	2.828kHz	$C9	17.148kHz
$2E	79.660Hz	$62	482.968Hz	$96	2.928kHz	$CA	17.753kHz
$2F	82.469Hz	$63	500.000Hz	$97	3.031kHz	$CB	18.379kHz
$30	85.378Hz	$64	517.632Hz	$98	3.138kHz	$CC	19.027kHz
$31	88.388Hz	$65	535.887Hz	$99	3.249kHz	$CD	19.698kHz
$32	91.505Hz	$66	554.785Hz	$9A	3.363kHz		
$33	94.732Hz	$67	574.349Hz	$9B	3.482kHz		

Delay1: 0 to 20 mS in 1 mS steps							
$00	0 mS	$06	6 mS	$0B	11	$10	16 mS
$01	1 mS	$07	7 mS	$0C	12	$11	17 mS
$02	2 mS	$08	8 mS	$0D	13	$12	18 mS
$03	3 mS	$09	9 mS	$0E	14	$13	19 mS
$04	4 mS	$0A	10 mS	$0F	15	$14	20 mS
$05	5 mS						

Bw1: 0.050 to 0.095 in 0.005 octave steps and 0.10 to 3.0 in 0.1 octave steps.							
$00	0.050	$0A	0.10	$14	1.1	$1E	2.1
$01	0.055	$0B	0.20	$15	1.2	$1F	2.2
$02	0.060	$0C	0.30	$16	1.3	$20	2.3
$03	0.065	$0D	0.40	$17	1.4	$21	2.4
$04	0.070	$0E	0.50	$18	1.5	$22	2.5
$05	0.075	$0F	0.60	$19	1.6	$23	2.6
$06	0.080	$10	0.70	$1A	1.7	$24	2.7
$07	0.085	$11	0.80	$1B	1.8	$25	2.8
$08	0.090	$12	0.90	$1C	1.9	$26	2.9
$09	0.095	$13	1.00	$1D	2.0	$27	3.0

END OF DOCUMENT