GRASS VALLEY GROUP

MODEL 1200

DIGITAL VIDEO SWITCHER

Serial Editor Protocol Manual
TP0758-00 C1

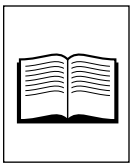
Publication Date: October, 1996

Contacting Grass Valley Group

		Voice	Fax	Address	Web Site
Customer Support	North America	(800) 547-8949 (530) 478-4148	(530) 478-3181	Grass Valley Group P.O. Box 1114 Grass Valley, CA 95945, USA	www.grassvalleygroup.com
	Elsewhere	Distributor or sales office from which equipment was purchased.			
Product, Service, Sales Information	North America	(800) 547-8949	(503) 627-7275	P.O. Box 500 M.S. 58-965 Beaverton, OR 97077-0001, USA	www.grassvalleygroup.com
	Europe	44 (10) 1628 40 3300	44 (0)1628 40 3301		
	Asia	(852) 2585-6688	(852) 2802-2996		
	Japan	81 (3) 5992 0621	81 (3) 5992 9377		
	Latin America	(305) 477-5488	(305) 477-5385		

Copyright © Grass Valley Group, Inc. All rights reserved.

This document may not be copied, in whole or in part, or otherwise reproduced, except as specifically permitted under U.S. copyright law, without the prior written consent of Grass Valley Group, Inc., P.O. Box 1114, Grass Valley, CA 95945, USA. GRASS VALLEY GROUP is a registered trademark and Grass Valley is a trademark of Grass Valley Group, Inc. All registered trademarks and trademarks are property of their respective holders. Grass Valley Group products are covered by U.S. and foreign patents, issued and pending. Product options and specifications subject to change without notice. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Grass Valley Group. Grass Valley Group assumes no responsibility or liability for any errors or inaccuracies that may appear in this publication.



Contents

Introduction 1-1

Manual Contents 1-1
Conventions Used in This Manual 1-1
Serial Interface Protocol 1-2
Functional Overview 1-3
Serial Data Word Description 1-4

Protocol Description 2-1

Serial Interface Protocol—What It Provides 2-1
Functional Description 2-1
- Crosspoint Matrix Control 2-2
- Pushbutton/Lamp Control 2-2
- Analog Control 2-2
- Auto Transition Rate Display 2-2
- E-MEM Control 2-3
  - Break Character 2-4
- Address Byte 2-5

Command/Message Block Structure 2-6
- Byte Count Byte 2-6
- Effects Address Byte 2-6
- Command Code Byte 2-7
- Read Command/Response 2-8
- Write Command/Response 2-9
- Message Bytes 2-10
- ESbus Protocol 2-10

Dialect Command Tables 3-1

Introduction to Commands 3-1
Keyer Commands 3-2
- Read Key1 Accumulation 3-2
- Write Key1 Accumulation 3-2
- Read Key2/DSK Accumulation 3-3
- Write Key2/DSK Accumulation 3-4

Write Effect Send 3-5
Crosspoint Commands 3-6
Read Xpt Prog/Aux 1 3-6
Write Xpt Program/Aux 1 3-7
Read Xpt Preset/Aux 2 3-8
Write Xpt Preset/Aux 2 3-9
Read Xpt Key1 Fill 3-10
Write Xpt Key1 Fill 3-10
Analog Control Commands 3-12
Read Analog 3-12
Write Analog 3-12
Wipe Pattern Commands 3-13
Read Wipe Pattern 3-13
Transition Commands 3-15
Read Trans Mode 3-15
Write Trans Mode 3-15
Read Auto Trans Rate 3-16
Write Auto Trans Rate 3-16
Read DSK Trans Rate 3-16
Write DSK Trans Rate 3-17
Read Fade to Black Rate 3-17
Write Fade to Black Rate 3-18
Read Fade to Black Rate 3-18
Write Fade to Black Rate 3-19
Write All Stop 3-19
E-MEM Commands 3-20
Write Learn E-MEM 3-20
Write Recall E-MEM 3-20
Read 200 Transfer E-MEM 3-21
Write 200 Transfer E-MEM 3-21
Protocol, 1200 Specific Dialect 3-22
Analog Read Commands 3-22
Analog Write Commands 3-34
Read Digital Value Function 3-49
Write Digital Value Function 3-50
Read Function 3-51
Write Function 3-56

1 *Introduction*

This document defines and discusses the communications protocol supported by the GVG Model 1200 switcher. The following pages detail the data formats and system timing requirements for commands exchanged between an Edit controller and the Model 1200. Note that the Model 1200 switcher is intended to support point-to-point communications between the switcher and a single controller (i.e., Editor).

Manual Contents

- Section 1 provides a *General Description* of the Serial Interface Protocol.
- Section 2, *Protocol Description*, describes the protocol functions, access scheme, and the command/message structure.
- Section 3 contains *Protocol Tables* that list each command in numerical order.

Conventions Used in This Manual

The following items are printed in all capital letters:

- Exact names of controls, lamps, and section titles on the control panel
- Major Headings

The following items are printed in initial capital letters:

- Names of figures, tables, and drawings
- Proper names of assemblies and modules
- Subordinate headings

Serial Interface Protocol

The Serial Interface Protocol enables an external device to have real-time remote control of Model 1200 switcher functions, except control of the ***[EDITOR ENABLE]*** pushbutton. The protocol operates via an editing device such as a videotape editing computer (from here on, this is simply referred to as an *editor*). Using the serial interface protocol, the editor may operate the Model 1200 switcher without the switcher control panel.

The Model 1200 implements a subset of the Model 1000 serial interface protocol which is based on the communications protocol of the SMPTE (Society of Motion Picture and Television Engineers) Recommended Practice 113. This protocol includes features similar to protocols for other GVG Switchers.

The Model 1200 takes action on a command from the editor two vertical intervals after the command has been completely received. (Note that a command must be received in its entirety at least 4 milliseconds before a vertical interval to ensure this occurs.) See Figure 1-1 for an example of the required timing.

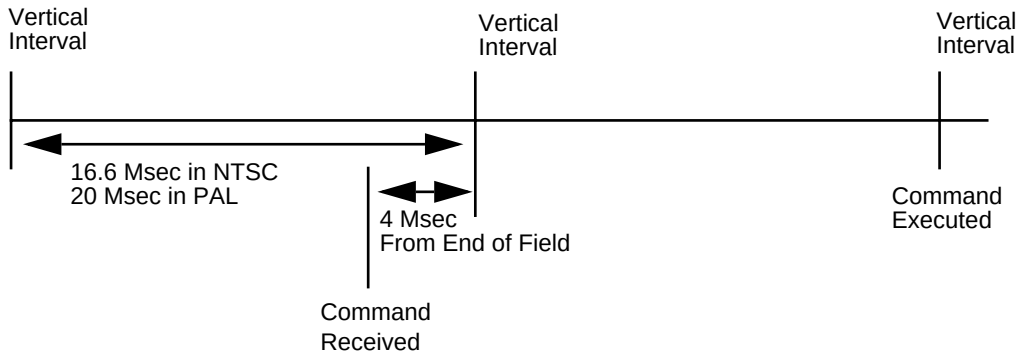


Figure 1-1. Command Execution Timing

The protocol for the Model 1200 embodies similar support for crosspoint control and E-MEM recalls and transitions as the Model 200-1, using a similar command system.

Functional Overview

An external editor controls the switcher via a 9-pin EDITOR serial connector (J32), located on the back of the electronics frame. Electrical and mechanical specifications for the port are shown in Table 1-1. Communication specifications are user-selectable from the status terminal (Refer to the installation section of your Model 1200 Service Manual, for details on using the status terminal.)

Table 1-1. Default Electrical Specifications (110 and 200-1)

Item	Description
Baud Rate	38,400
Word Size	8 bits, 1 start bit, 1 stop bit
Parity	Odd
Communications	RS-422 (SMPTE) or RS-232 (Jumper Selected)
Connector	9-pin D-subminiature
Device Address	30 (ES Bus 8080)

The pin outs for the ***EDITOR*** port are shown in Tables 1-2 and 1-3.

Table 1-2. RS-232 Pin-out

*Pin	Signal	Pin	Signal
1	NC	6	NC
2	RX	7	TX+
3	TX+	8	NC
4	NC	9	NC
5	Ground		

Table 1-3. RS-422 Pin-out (SMPTE)

*Pin	Signal	Pin	Signal
1	Ground	6	Ground
2	TX-	7	TX+
3	RX+	8	RX-
4	Ground	9	Ground
5	Ground		

Serial Data Word Description

The serial data word contains eleven bits as shown in Figure 1-2.

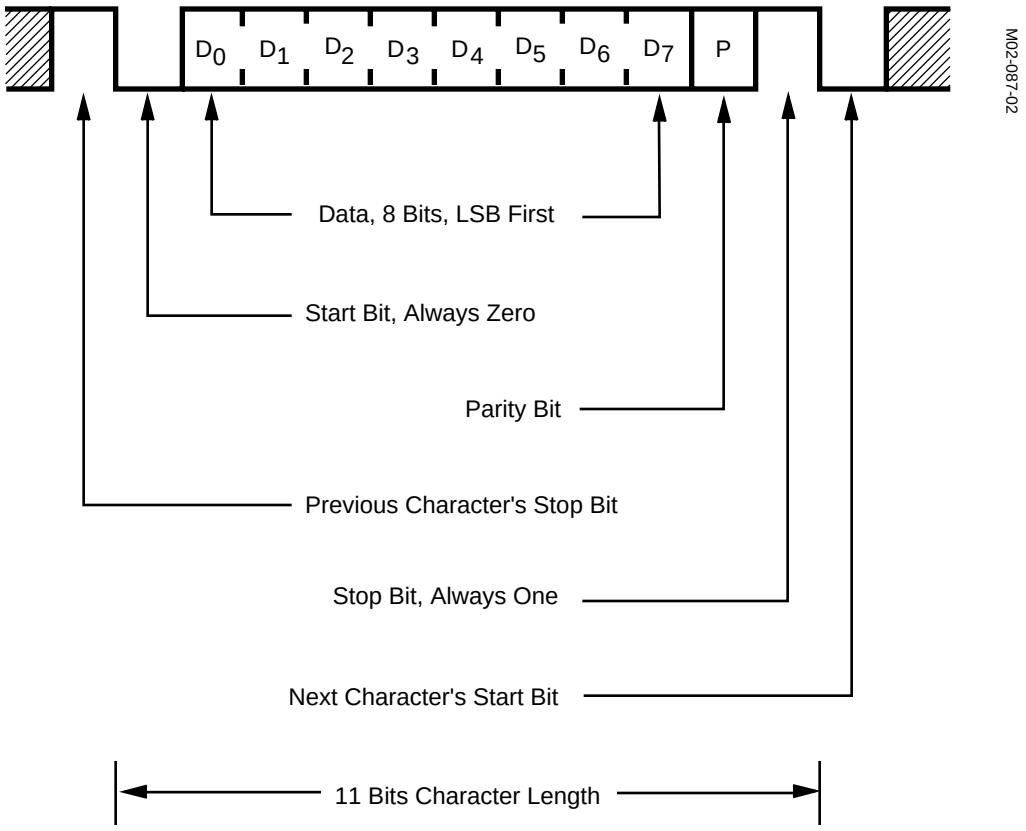


Figure 1-2. Serial Data Word

2 *Protocol Description*

Serial Interface Protocol—What It Provides

Model 1200 Serial Interface Protocol provides full duplex communications between the Model 1200 switcher and an external editor. In addition to providing remote switcher control by the editor, the Serial Interface also enables transfer of the entire switcher status, stored in an "effects memory" register (called the E-MEM register), to the editor. E-MEM contents can be read by the editor, stored off-line, and later written back into E-MEM and used to set the switcher to a previous configuration.

This section discusses the functional operation and implementation of the protocol.

Functional Description

The 1200 Serial Interface Protocol enables real-time remote operation of the Model 1200 switcher by an external editor. When the editor port is enabled, it can be used to operate the switcher, bypassing the switcher control panel; in fact, the control panel can be disconnected from the system entirely. If it is connected, an operator can use it to operate the switcher whether or not the editor is enabled. The following paragraphs highlight the primary functions of the switcher that can be accessed via the protocol interface. Refer to Section 3 for specific details (in table form) of each protocol command.

Crosspoint Matrix Control

The crosspoint matrix consists of the 3 Key Buses, Program Background Bus, Preset Background Bus, and 2 Aux Buses. The protocol can be used to read from and write to the matrix, or switcher frame. Write commands select the desired sources. One read command supplies the switcher software version number and a second read command is used to transfer E-MEM registers from the switcher to the editor.

Pushbutton/Lamp Control

All the switcher pushbuttons and their associated lamps, except the ***EDITOR ENABLE*** pushbutton, can be individually turned on and off.

Analog Control

Twelve knobs on the control panel adjust potentiometer settings. Some of these potentiometers adjust multiple parameters, according to other control panel selections. Two additional controls are a joystick (that contains two potentiometers, one for horizontal and one for vertical positioning) and a lever arm.

All of these controls, except the lever arm, produce digital values that range between a defined low value and a high value. In the case of the lever arm, the protocol enables the editor to write to the analog control.

Auto Transition Rate Display

The Transition Rate command sets the number of frames (0 to 999) for an automatic transition of the mix effects, a Downstream Keyer (DSK) mix, or a Fade to Black (FTB). The protocol can select the type of transition as well as the rate it is performed.

E-MEM Control

There are 20 valid E-MEM registers (Effects MEMory registers), numbered 0 through 19, available for recall or saving (called Learning). Figure 2-1 shows how data is transferred between the editor, E-MEM, and the switcher.

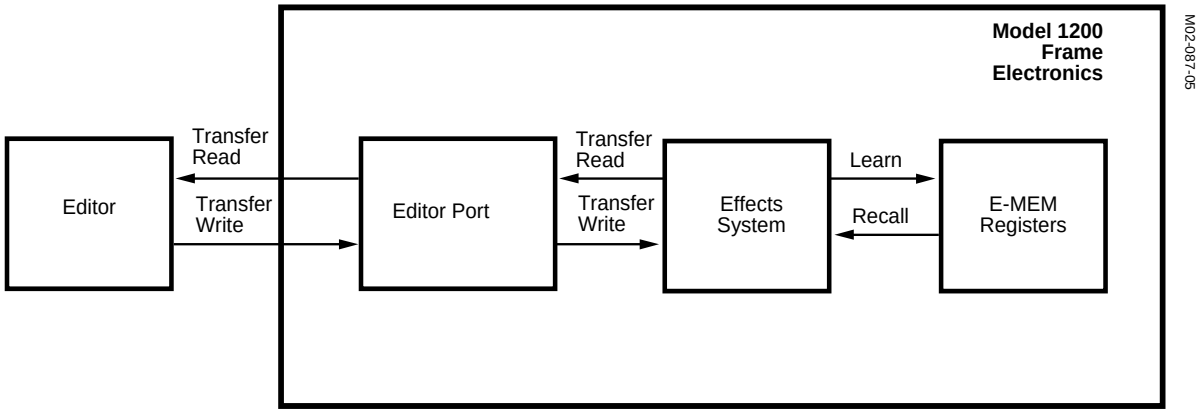


Figure 2-1. E-MEM Transfer

The protocol is able to learn up to 20 switcher configurations in E-MEM registers. The editor can then recall an E-MEM register in the switcher to configure it to the stored settings. The read command is used by the editor to read an E-MEM register and store its contents off-line in the editor. The write command is used to retrieve the data stored off-line in the editor and write it back into switcher E-MEM registers for later configuration of the switcher.

The 1200 Serial Interface Editor Protocol is based on the GVG 1000 protocol, as selected through menu options. The major components of the protocol are shown in Figure 2-2 and discussed in the following paragraphs.

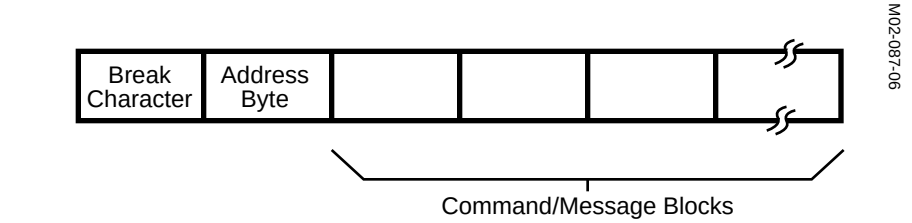


Figure 2-2. Components of the Model 1200 Editor Protocol Access

Break Character

The main component of the Editor Interface protocol is a 1-1/2 character-length "break." A break consists of a spacing or logical low (0) condition on the serial bus (see Figure 2-3) followed by a minimum of two bit-times of marking (logic 1) condition. The Bus Controller (editor) transmits a Break to the Model 1200 Switcher, forewarning of an impending message (Address Byte).

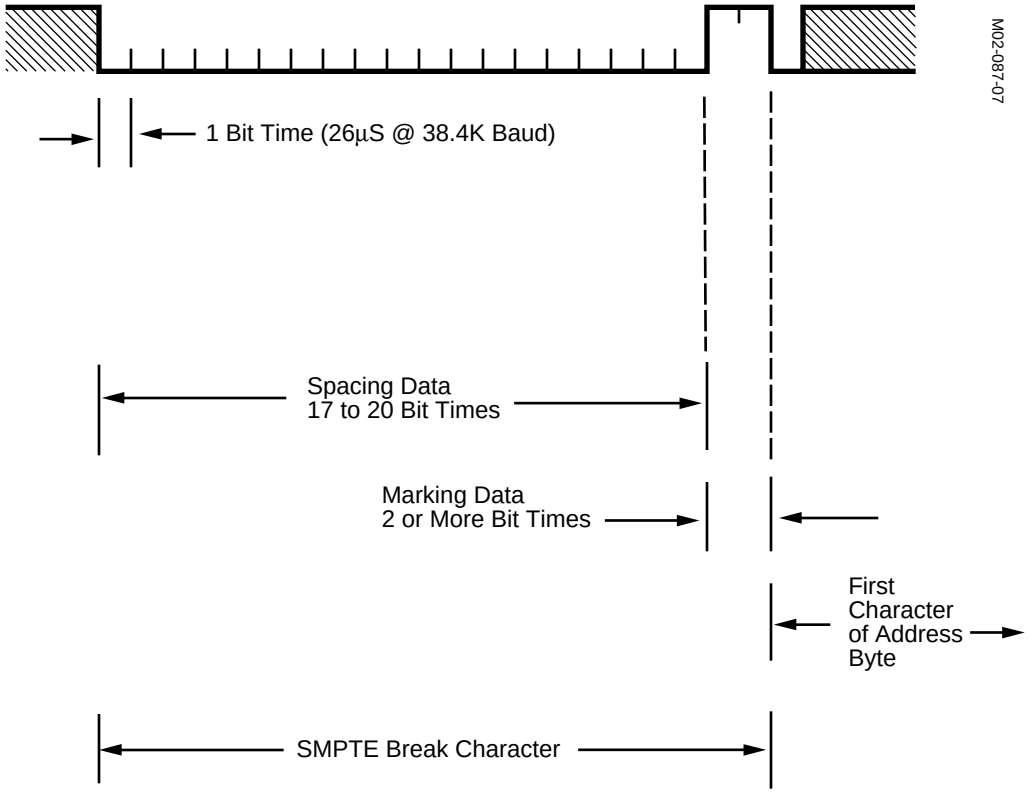


Figure 2-3. Break Character

The Break character forces all *tributaries*, or listeners (e.g.,the Model 1200 editor port), from the Idle state into the Active state. A Break can only be sent by the Bus Controller, or in this case the editor (e.g., a GVG VPE running Super Edit). When the Model 1200 switcher receives the Break character, it ceases all bus communications, resets any communication error condition, and prepares to receive an Address Byte.

Address Byte

Once a Break character has been received, the Model 1200 switcher advances from the Idle State to the Active State and begins expecting a one-byte address. The Model 1200 will respond to a default switcher address 30H (hexadecimal), when transmitted by the editor.

After receiving the address and responding with a single byte acknowledgement of 84H, the Editor Interface accepts command/message block transmissions containing commands and other data directed toward the switcher. Command/message blocks should be sent repeatedly, one after another, without Break characters.

In response to the address, the Model 1200 transmits an acknowledgment byte of 84H to the Editor, indicating the change in status to the Selected State. If an address of 31H is received (addr-poll), the Model 1200 transmits an acknowledgment byte of 84H to the controller, but stays in the Active State (i.e., does not go into the Selected State). If the address does not match, the tributary drops off the bus (i.e., returns to the Idle State) and waits for another Break.

Command/Message Block Structure

The serial interface can operate continuously, receiving a contiguous stream of command/message blocks. This enables multiple switcher operations to be performed without constant cycling through the break-address sequence. Command/message blocks received (all bytes) with more than four milliseconds remaining in the current television field are executed in the second vertical interval that follows. Blocks received after this time may be delayed one additional field.

Byte Count Byte

The command/message blocks can range in size from 2 to 128 bytes. The first byte of the block contains the byte count. The byte count is the number of the subsequent bytes in the block. Thus the valid byte count values are 02H to FFH (2 to 255). After receiving a valid byte count and the correct number of subsequent bytes, the command is executed.

Effects Address Byte

The effects address byte is typically referred to as EX and can have one of several values. To access any DSK analog controls, EX must be set to 00H. For the analog controls of the effects systems (EFFECTS KEYER, EFFECTS TRANSITION, and PATTERN CONTROL), EX must be set to 01H. Refer to the specific command descriptions for use of effects addresses on all other controls: EX should be 01H.

Command Code Byte

Command codes fall into two broad categories; read commands and write commands (see Figure 2-4). When the editor issues a read command, the switcher responds by sending the current status of an operational parameter or parameters back to the editor. The status information is returned in the format of a write command. Write commands are used by the editor to change operational parameters of the switcher.

NOTE: *There is not a one-to-one correspondence between read and write commands. Only commands specifically described in Section 3 are implemented.*

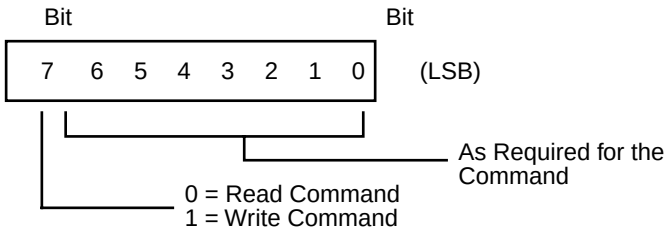


Figure 2-4. Command Byte Structure

Both read and write commands may be sent at any time. When a write command is issued with the ***EDITOR ENABLE*** pushbutton ON (lamp lit), a 2-byte status message is returned to the editor and the command is executed.

With the ***EDITOR ENABLE*** pushbutton OFF, the write command cannot be executed but the status message is still returned. Read commands are always performed, regardless of the state of the ***EDITOR ENABLE*** pushbutton.

Read Command/Response

Read commands interrogate the status of the specified operational parameter of the switcher, such as the crosspoint currently selected on the program bus. Read commands may or may not require message bytes following the command code.

The switcher responds to a read command by transmitting to the editor a command/message block in the format of a write command. By echoing the write command format, responses to a read command can later be sent back to the switcher by the editor to cause execution of that command.

Write Command/Response

Write commands are sent to the switcher to modify an operational parameter (such as to change a crosspoint or to set an analog control) or to initiate execution of a function (such as an auto transition). All write commands require at least one byte of message data, which typically specifies the control parameter or setting of the control parameter.

The response to a write command is either 80H (Command Accepted), 40H (Command Rejected), or an Error has occur. The switcher response to the command is shown in Figure 2-5. The bits that are set to one (1) indicate the status.

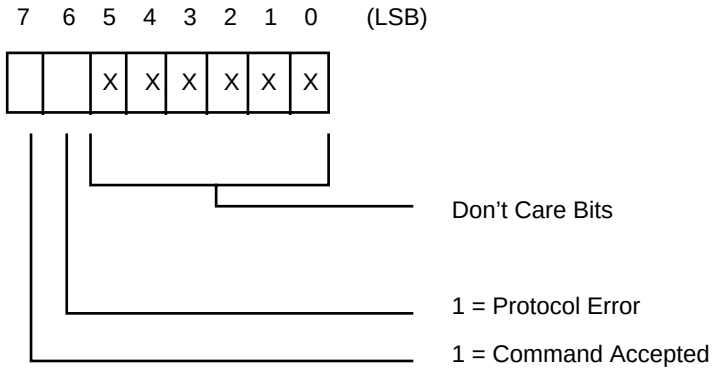


Figure 2-5. Second Byte of Status Message

In the Figure 2-5, bit 7 set to one (1) indicates the command was accepted and execution of it shall occur (if the ***EDITOR ENABLE*** is on). Bit 6 set to one indicates an illegal function may have been specified.

Message Bytes

The number of required message bytes depends upon the particular command. Write commands require at least two message bytes that typically specify effects address plus the code number of the switcher parameter to be changed. Additional message bytes are required for many commands.

Read commands typically do not require a message byte. However, the Read Analog Control, Read Pushbutton/Lamp Control, and Read Lamp Status Map Commands do require message bytes that indicate the control or lamp to be read.

ESbus Protocol

The Model 1200 supports ESbus protocol. ESbus is a communication system developed specifically for the remote control of television equipment. It is the joint work of the Society of Motion Picture and Television Engineers (SMPTE) and the European Broadcast Union (EBU). The ESbus provides a means for anywhere from two to a large number of devices to communicate with each other in a very flexible manner over a common bus. The bus consists of two twisted pairs that carry data at 38.4K baud, even parity, RS-422. Its uses a 9-pin D connector.

The ESbus uses a Break character as an attention signal to establish communication between a tributary and the controller. This protocol provides the greatest protection (though no correction) from communication errors.

3 *Dialect Command Tables*

Introduction to Commands

This section lists the Model 1200 dialect commands used to provide serial control of the Model 1200 switcher. In general, write commands will set the switcher to the desired condition. Read commands return the corresponding write command which would set the current condition.

NOTE: *All values within the tables are hexadecimal, unless otherwise indicated by the notation of "decimal". Unless indicated as 00H or 01H, EX may be any value.*

Keyer Commands

In general, read commands result in the return of the corresponding write command which would set the current condition.

Read Key1 Accumulation

Provides a view of key source and fill selections to the editor. The response is in the form of a Write Key1 Acc command which reflects the current status.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code

BC EF CC
02 01 3F          Read Key 1 Accumulation
```

Write Key1 Accumulation

Provides the editor with the means of setting the source and fill selections to either a self key or a split key. The accumulation command does not allow for accumulation of key sources since the accumulation cannot be performed by the switcher. This form of command was chosen to conform to the forms used by the Model 200 switcher.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      pf = Primary Fill
      ef = External Fill = 00
      ks = 00 = not split
           01 = split
      ps = Primary Source
      es = External Source = 00

BC EF CC MG
         pf ef ks ps es
07 01 BF 01 00 00 01 00 Write Key 1 Fill Xpt 1 no-split
07 01 BF 02 00 00 02 00 Write Key 1 Fill Xpt 2 no-split
07 01 BF 03 00 00 03 00 Write Key 1 Fill Xpt 3 no-split
07 01 BF 04 00 00 04 00 Write Key 1 Fill Xpt 4 no-split
```

```
07 01 BF 05 00 00 05 00 Write Key 1 Fill Xpt 5 no-split
07 01 BF 06 00 00 06 00 Write Key 1 Fill Xpt 6 no-split
07 01 BF 07 00 00 07 00 Write Key 1 Fill Xpt 7 no-split
07 01 BF 08 00 00 08 00 Write Key 1 Fill Xpt 8 no-split
07 01 BF 09 00 00 09 00 Write Key 1 Fill Xpt 9 no-split
07 01 BF 0A 00 00 0A 00 Write Key 1 Fill Xpt 10 no-split
07 01 BF 0B 00 00 0B 00 Write Key 1 Fill Xpt 11 no-split
07 01 BF 0C 00 00 0C 00 Write Key 1 Fill Xpt 12 no-split
07 01 BF 0D 00 00 0D 00 Write Key 1 Fill Xpt 13 no-split
07 01 BF 0E 00 00 0E 00 Write Key 1 Fill Xpt 14 no-split

07 01 BF 01 00 01 ps 00 Write Key 1 Fill Xpt 1 split src ps
07 01 BF 02 00 01 ps 00 Write Key 1 Fill Xpt 2 split src ps
07 01 BF 03 00 01 ps 00 Write Key 1 Fill Xpt 3 split src ps
07 01 BF 04 00 01 ps 00 Write Key 1 Fill Xpt 4 split src ps
07 01 BF 05 00 01 ps 00 Write Key 1 Fill Xpt 5 split src ps
07 01 BF 06 00 01 ps 00 Write Key 1 Fill Xpt 6 split src ps
07 01 BF 07 00 01 ps 00 Write Key 1 Fill Xpt 7 split src ps
07 01 BF 08 00 01 ps 00 Write Key 1 Fill Xpt 8 split src ps
07 01 BF 09 00 01 ps 00 Write Key 1 Fill Xpt 9 split src ps
07 01 BF 0A 00 01 ps 00 Write Key 1 Fill Xpt 10 split src ps
07 01 BF 0B 00 01 ps 00 Write Key 1 Fill Xpt 11 split src ps
07 01 BF 0C 00 01 ps 00 Write Key 1 Fill Xpt 12 split src ps
07 01 BF 0D 00 01 ps 00 Write Key 1 Fill Xpt 13 split src ps
07 01 BF 0E 00 01 ps 00 Write Key 1 Fill Xpt 14 split src ps
```

Read Key2/DSK Accumulation

Provides a view of key source and fill selections to the editor. The response is in the form of a Write Key2 Acc command which reflects the current status. The effects code of the command has been misused to differentiate between Key 2 and the DSK (0=DSK, 1=Key 2).

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message

BC EF CC MG
02 01 40                        Read Key 2 Accumulation

02 00 40                        Read DSK Accumulation
```

Write Key2/DSK Accumulation

Provides the editor with the means of setting the source and fill selections to either a self key or a split key. The accumulation command does not allow for accumulation of key sources since the accumulation cannot be performed by the switcher. This form of command was chosen to conform to the forms used by the Model 200 switcher. The effects code of the command has been misused to differentiate between Key 2 and the DSK. It is important to not select the keyer "Video Key" or "Auto Select Key" buttons after a split key source has been selected because either of these buttons forces the key to a self key, dropping the breakaway source.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message

BC EF CC MG
         pf ef ks ps es
07 01 C0 01 00 00 01 00 Write Key 2 Fill Xpt 1 no-split
07 01 C0 02 00 00 02 00 Write Key 2 Fill Xpt 2 no-split
07 01 C0 03 00 00 03 00 Write Key 2 Fill Xpt 3 no-split
07 01 C0 04 00 00 04 00 Write Key 2 Fill Xpt 4 no-split
07 01 C0 05 00 00 05 00 Write Key 2 Fill Xpt 5 no-split
07 01 C0 06 00 00 06 00 Write Key 2 Fill Xpt 6 no-split
07 01 C0 07 00 00 07 00 Write Key 2 Fill Xpt 7 no-split
07 01 C0 08 00 00 08 00 Write Key 2 Fill Xpt 8 no-split
07 01 C0 09 00 00 09 00 Write Key 2 Fill Xpt 9 no-split
07 01 C0 0A 00 00 0A 00 Write Key 2 Fill Xpt 10 no-split
07 01 C0 0B 00 00 0B 00 Write Key 2 Fill Xpt 11 no-split
07 01 C0 0C 00 00 0C 00 Write Key 2 Fill Xpt 12 no-split
07 01 C0 0D 00 00 0D 00 Write Key 2 Fill Xpt 13 no-split
07 01 C0 0E 00 00 0E 00 Write Key 2 Fill Xpt 14 no-split

07 01 C0 01 00 01 ps 00 Write Key 2 Fill Xpt 1 split src ps
07 01 C0 02 00 01 ps 00 Write Key 2 Fill Xpt 2 split src ps
07 01 C0 03 00 01 ps 00 Write Key 2 Fill Xpt 3 split src ps
07 01 C0 04 00 01 ps 00 Write Key 2 Fill Xpt 4 split src ps
07 01 C0 05 00 01 ps 00 Write Key 2 Fill Xpt 5 split src ps
07 01 C0 06 00 01 ps 00 Write Key 2 Fill Xpt 6 split src ps
07 01 C0 07 00 01 ps 00 Write Key 2 Fill Xpt 7 split src ps
07 01 C0 08 00 01 ps 00 Write Key 2 Fill Xpt 8 split src ps
07 01 C0 09 00 01 ps 00 Write Key 2 Fill Xpt 9 split src ps
07 01 C0 0A 00 01 ps 00 Write Key 2 Fill Xpt 10 split src ps
07 01 C0 0B 00 01 ps 00 Write Key 2 Fill Xpt 11 split src ps
07 01 C0 0C 00 01 ps 00 Write Key 2 Fill Xpt 12 split src ps
07 01 C0 0D 00 01 ps 00 Write Key 2 Fill Xpt 13 split src ps
07 01 C0 0E 00 01 ps 00 Write Key 2 Fill Xpt 14 split src ps

07 00 C0 01 00 00 01 00 Write DSK Fill Xpt 1 no-split
07 00 C0 02 00 00 02 00 Write DSK Fill Xpt 2 no-split
07 00 C0 03 00 00 03 00 Write DSK Fill Xpt 3 no-split
```

```
07 00 C0 04 00 00 04 00 Write DSK Fill Xpt 4 no-split
07 00 C0 05 00 00 05 00 Write DSK Fill Xpt 5 no-split
07 00 C0 06 00 00 06 00 Write DSK Fill Xpt 6 no-split
07 00 C0 07 00 00 07 00 Write DSK Fill Xpt 7 no-split
07 00 C0 08 00 00 08 00 Write DSK Fill Xpt 8 no-split
07 00 C0 09 00 00 09 00 Write DSK Fill Xpt 9 no-split
07 00 C0 0A 00 00 0A 00 Write DSK Fill Xpt 10 no-split
07 00 C0 0B 00 00 0B 00 Write DSK Fill Xpt 11 no-split
07 00 C0 0C 00 00 0C 00 Write DSK Fill Xpt 12 no-split
07 00 C0 0D 00 00 0D 00 Write DSK Fill Xpt 13 no-split
07 00 C0 0E 00 00 0E 00 Write DSK Fill Xpt 14 no-split

07 00 C0 01 00 01 ps 00 Write DSK Fill Xpt 1 split src ps
07 00 C0 02 00 01 ps 00 Write DSK Fill Xpt 2 split src ps
07 00 C0 03 00 01 ps 00 Write DSK Fill Xpt 3 split src ps
07 00 C0 04 00 01 ps 00 Write DSK Fill Xpt 4 split src ps
07 00 C0 05 00 01 ps 00 Write DSK Fill Xpt 5 split src ps
07 00 C0 06 00 01 ps 00 Write DSK Fill Xpt 6 split src ps
07 00 C0 07 00 01 ps 00 Write DSK Fill Xpt 7 split src ps
07 00 C0 08 00 01 ps 00 Write DSK Fill Xpt 8 split src ps
07 00 C0 09 00 01 ps 00 Write DSK Fill Xpt 9 split src ps
07 00 C0 0A 00 01 ps 00 Write DSK Fill Xpt 10 split src ps
07 00 C0 0B 00 01 ps 00 Write DSK Fill Xpt 11 split src ps
07 00 C0 0C 00 01 ps 00 Write DSK Fill Xpt 12 split src ps
07 00 C0 0D 00 01 ps 00 Write DSK Fill Xpt 13 split src ps
07 00 C0 0E 00 01 ps 00 Write DSK Fill Xpt 14 split src ps
```

Write Effect Send

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      sw = sw - 00 = OFF
                01 = ON

BC EF CC MM
         sw

03 01 CE sw
```

Crosspoint Commands

Read Xpt Prog/Aux1

Provides a means for the editor to determine which crosspoint is selected on either the program row or aux 1. The effects code of the command is used to differentiate between the program row and Aux 1.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message

BC EF CC MG

03 01 41 01       Read Program Crosspoint 1
03 01 41 02       Read Program Crosspoint 2
03 01 41 03       Read Program Crosspoint 3
03 01 41 04       Read Program Crosspoint 4
03 01 41 05       Read Program Crosspoint 5
03 01 41 06       Read Program Crosspoint 6
03 01 41 07       Read Program Crosspoint 7
03 01 41 08       Read Program Crosspoint 8
03 01 41 09       Read Program Crosspoint 9
03 01 41 0A       Read Program Crosspoint 10
03 01 41 0B       Read Program Crosspoint 11
03 01 41 0C       Read Program Crosspoint 12
03 01 41 0D       Read Program Crosspoint 13
03 01 41 0E       Read Program Crosspoint 14

03 07 41 01       Read Aux 1 Crosspoint 1
03 07 41 02       Read Aux 1 Crosspoint 2
03 07 41 03       Read Aux 1 Crosspoint 3
03 07 41 04       Read Aux 1 Crosspoint 4
03 07 41 05       Read Aux 1 Crosspoint 5
03 07 41 06       Read Aux 1 Crosspoint 6
03 07 41 07       Read Aux 1 Crosspoint 7
03 07 41 08       Read Aux 1 Crosspoint 8
03 07 41 09       Read Aux 1 Crosspoint 9
03 07 41 0A       Read Aux 1 Crosspoint 10
03 07 41 0B       Read Aux 1 Crosspoint 11
03 07 41 0C       Read Aux 1 Crosspoint 12
03 07 41 0D       Read Aux 1 Crosspoint 13
03 07 41 0E       Read Aux 1 Crosspoint 14
```

Write Xpt Program/Aux1

Provides a means for the editor to select the crosspoint on either the Program row or Aux 1. The effects code of the command is used to differentiate between the Program row and Aux 1.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message

BC EF CC MG

03 01 C1 01       Write Program Crosspoint 1
03 01 C1 02       Write Program Crosspoint 2
03 01 C1 03       Write Program Crosspoint 3
03 01 C1 04       Write Program Crosspoint 4
03 01 C1 05       Write Program Crosspoint 5
03 01 C1 06       Write Program Crosspoint 6
03 01 C1 07       Write Program Crosspoint 7
03 01 C1 08       Write Program Crosspoint 8
03 01 C1 09       Write Program Crosspoint 9
03 01 C1 0A       Write Program Crosspoint 10
03 01 C1 0B       Write Program Crosspoint 11
03 01 C1 0C       Write Program Crosspoint 12
03 01 C1 0D       Write Program Crosspoint 13
03 01 C1 0E       Write Program Crosspoint 14

03 07 C1 01       Write Aux 1 Crosspoint 1
03 07 C1 02       Write Aux 1 Crosspoint 2
03 07 C1 03       Write Aux 1 Crosspoint 3
03 07 C1 04       Write Aux 1 Crosspoint 4
03 07 C1 05       Write Aux 1 Crosspoint 5
03 07 C1 06       Write Aux 1 Crosspoint 6
03 07 C1 07       Write Aux 1 Crosspoint 7
03 07 C1 08       Write Aux 1 Crosspoint 8
03 07 C1 09       Write Aux 1 Crosspoint 9
03 07 C1 0A       Write Aux 1 Crosspoint 10
03 07 C1 0B       Write Aux 1 Crosspoint 11
03 07 C1 0C       Write Aux 1 Crosspoint 12
03 07 C1 0D       Write Aux 1 Crosspoint 13
03 07 C1 0E       Write Aux 1 Crosspoint 14
```

Read Xpt Preset/Aux2

Provides a means for the editor to determine which crosspoint is selected on either the preset row or aux 2. The effects code of the command is used to differentiate between the preset row and Aux 2.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message

BC EF CC MG

03 01 42 01       Read Preset Crosspoint 1
03 01 42 02       Read Preset Crosspoint 2
03 01 42 03       Read Preset Crosspoint 3
03 01 42 04       Read Preset Crosspoint 4
03 01 42 05       Read Preset Crosspoint 5
03 01 42 06       Read Preset Crosspoint 6
03 01 42 07       Read Preset Crosspoint 7
03 01 42 08       Read Preset Crosspoint 8
03 01 42 09       Read Preset Crosspoint 9
03 01 42 0A       Read Preset Crosspoint 10
03 01 42 0B       Read Preset Crosspoint 11
03 01 42 0C       Read Preset Crosspoint 12
03 01 42 0D       Read Preset Crosspoint 13
03 01 42 0E       Read Preset Crosspoint 14

03 07 42 01       Read Aux 2 Crosspoint 1
03 07 42 02       Read Aux 2 Crosspoint 2
03 07 42 03       Read Aux 2 Crosspoint 3
03 07 42 04       Read Aux 2 Crosspoint 4
03 07 42 05       Read Aux 2 Crosspoint 5
03 07 42 06       Read Aux 2 Crosspoint 6
03 07 42 07       Read Aux 2 Crosspoint 7
03 07 42 08       Read Aux 2 Crosspoint 8
03 07 42 09       Read Aux 2 Crosspoint 9
03 07 42 0A       Read Aux 2 Crosspoint 10
03 07 42 0B       Read Aux 2 Crosspoint 11
03 07 42 0C       Read Aux 2 Crosspoint 12
03 07 42 0D       Read Aux 2 Crosspoint 13
03 07 42 0E       Read Aux 2 Crosspoint 14
```

Write Xpt Preset/Aux2

Provides a means for the editor to select the crosspoint on either the preset row or aux 2. The effects code of the command is used to differentiate between the preset row and Aux 2.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message

BC EF CC MG

03 01 C2 01       Write Preset Crosspoint 1
03 01 C2 02       Write Preset Crosspoint 2
03 01 C2 03       Write Preset Crosspoint 3
03 01 C2 04       Write Preset Crosspoint 4
03 01 C2 05       Write Preset Crosspoint 5
03 01 C2 06       Write Preset Crosspoint 6
03 01 C2 07       Write Preset Crosspoint 7
03 01 C2 08       Write Preset Crosspoint 8
03 01 C2 09       Write Preset Crosspoint 9
03 01 C2 0A       Write Preset Crosspoint 10
03 01 C2 0B       Write Preset Crosspoint 11
03 01 C2 0C       Write Preset Crosspoint 12
03 01 C2 0D       Write Preset Crosspoint 13
03 01 C2 0E       Write Preset Crosspoint 14

03 07 C2 01       Write Aux 2 Crosspoint 1
03 07 C2 02       Write Aux 2 Crosspoint 2
03 07 C2 03       Write Aux 2 Crosspoint 3
03 07 C2 04       Write Aux 2 Crosspoint 4
03 07 C2 05       Write Aux 2 Crosspoint 5
03 07 C2 06       Write Aux 2 Crosspoint 6
03 07 C2 07       Write Aux 2 Crosspoint 7
03 07 C2 08       Write Aux 2 Crosspoint 8
03 07 C2 09       Write Aux 2 Crosspoint 9
03 07 C2 0A       Write Aux 2 Crosspoint 10
03 07 C2 0B       Write Aux 2 Crosspoint 11
03 07 C2 0C       Write Aux 2 Crosspoint 12
03 07 C2 0D       Write Aux 2 Crosspoint 13
03 07 C2 0E       Write Aux 2 Crosspoint 14
```

Read Xpt Key1 Fill

Provides a means for the editor to determine the key 1 fill crosspoint. It does not indicate the source crosspoint.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message

BC EF CC

03 01 43    Read Key 1 Fill Crosspoint
```

Write Xpt Key1 Fill

Provides a means for the editor to set the key 1 fill crosspoint. This command results in a self key, in which the source and fill are the same selection.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message

BC EF CC MG

03 01 C3 01       Write Key 1 Fill Crosspoint 1
03 01 C3 02       Write Key 1 Fill Crosspoint 2
03 01 C3 03       Write Key 1 Fill Crosspoint 3
03 01 C3 04       Write Key 1 Fill Crosspoint 4
03 01 C3 05       Write Key 1 Fill Crosspoint 5
03 01 C3 06       Write Key 1 Fill Crosspoint 6
03 01 C3 07       Write Key 1 Fill Crosspoint 7
03 01 C3 08       Write Key 1 Fill Crosspoint 8
03 01 C3 09       Write Key 1 Fill Crosspoint 9
03 01 C3 0A       Write Key 1 Fill Crosspoint 10
03 01 C3 0B       Write Key 1 Fill Crosspoint 11
03 01 C3 0C       Write Key 1 Fill Crosspoint 12
03 01 C3 0D       Write Key 1 Fill Crosspoint 13
03 01 C3 0E       Write Key 1 Fill Crosspoint 14
```

Read Xpt Key2 Fill

Provides a means for the editor to determine the key 2 fill crosspoint. It does not indicate the source crosspoint.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message

BC EF CC

03 01 44    Read Key 2 Fill Crosspoint
```

Write Xpt Key2 Fill

Provides a means for the editor to set the key 2 fill crosspoint. This command results in a self key, in which the source and fill are the same selection.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message

BC EF CC MG

03 01 C4 01       Write Key 2 Fill Crosspoint 1
03 01 C4 02       Write Key 2 Fill Crosspoint 2
03 01 C4 03       Write Key 2 Fill Crosspoint 3
03 01 C4 04       Write Key 2 Fill Crosspoint 4
03 01 C4 05       Write Key 2 Fill Crosspoint 5
03 01 C4 06       Write Key 2 Fill Crosspoint 6
03 01 C4 07       Write Key 2 Fill Crosspoint 7
03 01 C4 08       Write Key 2 Fill Crosspoint 8
03 01 C4 09       Write Key 2 Fill Crosspoint 9
03 01 C4 0A       Write Key 2 Fill Crosspoint 10
03 01 C4 0B       Write Key 2 Fill Crosspoint 11
03 01 C4 0C       Write Key 2 Fill Crosspoint 12
03 01 C4 0D       Write Key 2 Fill Crosspoint 13
03 01 C4 0E       Write Key 2 Fill Crosspoint 14
```

Analog Control Commands

Read Analog

Provides a means for the editor to determine the analog control values for the switcher lever arm.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      an = Analog number

BC EF CC MG
         an

03 01 45 00            00 Read lever arm
```

Write Analog

Provides a means for the editor to set the analog control values for the switcher lever arm.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      an = Analog value
      dd = Value - a 12 bit binary value with the 8 high order bits
           in the first byte and the 4 low order bits in the high
           order bit positions of the second byte. The 4 low order
           bits of the second byte should be 0's and are considered
           "don't care" bits.

BC EF CC MG
         an dd dd

05 01 CF 00 DD DD00 Write lever arm value
```

Wipe Pattern Commands

Read Wipe Pattern

Reads the currently selected wipe pattern.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code

BC EF CC

02 01 48    Read Wipe Pattern
```

Write Wipe Pattern

Selects a wipe pattern.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message

ww = (hex) Pattern Number -   0A = Curtain Vertical-Moving
                              01 = Corner Top Left
                              02 = Vee Top
                              03 = Circle
                              0E = Diagonal Top Right

                              00 = Curtain Horizontal-Moving
                              0B = Corner Top Right
                              0C = Vee Left
                              28 = Rectangle
                              04 = Diagonal Top Left

                              1E = Curtain Split Vertical-Moving
                              15 = Corner Bottom Right
                              16 = Vee Right
                              17 = Box
                              18 = Cross

                              14 = Curtain Split Horizontal-Moving
                              1F = Corner Bottom Left
                              20 = Vee Bottom
                              21 = Diamond
                              22 = Bow Tie


BC EF CC MG
         ww

03 01 C8 ww                     Write Wipe Pattern
```

Transition Commands

Read Trans Mode

Reads the currently selected mode on the transition panel

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code

BC EF CC

02 01 4A
```

Write Trans Mode

Sets the transition mode on the transition panel.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      tm = Transmission Mode - XXXXXXXX (one byte)
                          |||||||| - 1 = Key 2 Selected
                          |||||||-- 1 = Key 1 Selected
                          ||||||--- 1 = Bkgd Selected
                          |||||---- Don't Care
                          ||||----- Don't Care
                          |||------ Don't Care
                          ||------- Don't Care
                          |-------- Don't Care

BC EF CC MG
         tm

03 01 CA tm
```

Read Auto Trans Rate

Reads current auto transition rate for the selected transition mode.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code

BC EF CC

02 01 4C
```

Write Auto Trans Rate

The Auto Trans rate is expressed in hex characters ranging from 000 to 999. These characters can be considered binary character decimal characters (BCD).

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      r1 = Rate, Most Significant Character
      r2 = Rate, Second Most Significant Character
      r3 = Rate, Least significant Character

BC EF CC MM
         r1 r2 r3

05 01 CC r1 r2 r3
```

Read DSK Trans Rate

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code

BC EF CC

02 01 4D
```

Write DSK Trans Rate

The Auto Trans rate is expressed in hex characters ranging from 000 to 999. These characters can be considered binary character decimal characters (BCD).

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      r1 = Rate, Most Significant Character
      r2 = Rate, Second Most Significant Character
      r3 = Rate, Least significant Character

BC EF CC MM
         r1 r2 r3

05 01 CD r1 r2 r3
```

Read Fade to Black Rate

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code

BC EF CC

02 01 7D
```

Write Fade to Black Rate

The Auto Trans rate is expressed in hex characters ranging from 000 to 999. These characters can be considered binary coded decimal characters (BCD).

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      r1 = Rate, Most Significant Character
      r2 = Rate, Second Most Significant Character
      r3 = Rate, Least significant Character

BC EF CC MM
         r1 r2 r3

05 01 FD r1 r2 r3
```

Read Fade to Black Rate

Returns to the editor the Fade to Black rate in the form of a command to set the current rate.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code

BC EF CC

03 01 7D
```

Write Fade to Black Rate

Causes the switcher to set the Fade to Black rate (FTB rate).

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message Bytes
      r1 = Rate, Most Significant Character
      r2 = Rate, Second Most Significant Character
      r3 = Rate, Least significant Character

BC EF CC MG
         r1 r2 r3

05 00 FD r1 r2 r3
```

Write All Stop

Causes the switcher to halt any ongoing auto transition. The message byte is included to maintain similarity to other protocols but is ignored by the switcher.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message Byte
      mb = message byte - ignored by the 1200

BC EF CC MM
         mb

03 01 F2 00
```

E-MEM Commands

Write Learn E-MEM

Causes the 1200 switcher to perform a learn status to the register specified.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      rg = Register Number - 0 to 13 hex

BC EF CC MM
         rg

03 01 DA rg
```

Write Recall E-MEM

Causes the 1200 switcher to perform a register recall from the register specified.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      rg = Register Number - 0 to 13 hex

BC EF CC MM
         rg

03 01 DB rg
```

Read 200 Transfer E-MEM

Causes the 1200 switcher to write a register to the editor interface. Since the registers are large they are broken into multiple segments, designated with the effects numbers 00, 01, 02,... The editor must request all the segments to get a single entire register. Typically, the editor will request successive segments until it receives a null segment which is indicated by a byte count of 12 (0C hex). The set of segments excluding the null segment constitute the register. When returned to the switcher in the same format that they were sent to the editor, they will result in a replacement of the indicated register.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      rg = Register Number - 0 to 13 hex

BC EF CC MM
         rg

03 00 5E rg Read the first segment of register "rg"
03 01 5E rg Read the second segment of register "rg"
03 02 5E rg Read the third (and last) segment of
            register "rg"
```

Write 200 Transfer E-MEM

Causes the 1200 switcher to accept or read a register. Since the registers are large they are broken into three segments, designated with the effects numbers 00, 01, and 02. The editor must write all three segments to transfer a single entire register.

Protocol Command:

```
BC = Byte Count
EF = Effects
CC = Command Code
MG = Message
      rg = Register Number - 0 to 13 hex

BC EF CC MM
         rg data

BC 00 FE rg dataWrite the first segment of register "rg"
```

Note: the preset wipe border must have the border button on to get a border or to adjust its width, the softness button must be on to get the softness to work, and the symmetry button must be on to get the symmetry value to work. The same is true for preset pattern—the preset pattern/wipe delegation is strictly in the panel.

Protocol, 1200 Specific Dialect

Analog Read Commands

```
Analog reads:

03 01 57 nn
|  |  |  |_____________ designates analog value to read
|  |  |________________ command: 57h = read an analog value
|  |___________________ effects (use 01h for all)
|______________________ byte count for the command
```

All the analog read comands return a four byte integer value. The range of values that have meaning are found in the description of the corresponding analog write command.

The following commands read matte values for the currently selected crosspoint for the bus indicated. The matte values are in effect if the crosspoint is defined as a matte.

```
1   Read crosspoint prog hue

    03 01 57 01
          Returns: 07 01 D7 01 v1 v2 v3 v4
                where v1v2v3v4 is the 32 bit integer value

2   Read crosspoint prog chroma

    03 01 57 02
          Returns: 07 01 D7 02 v1 v2 v3 v4
                where v1v2v3v4 is the 32 bit integer value

3   Read crosspoint prog luma

    03 01 57 03
          Returns: 07 01 D7 03 v1 v2 v3 v4
                where v1v2v3v4 is the 32 bit integer value

4   Read crosspoint preset hue

    03 01 57 04
          Returns: 07 01 D7 04 v1 v2 v3 v4
                where v1v2v3v4 is the 32 bit integer value

5   Read crosspoint preset chroma

    03 01 57 05
          Returns: 07 01 D7 05 v1 v2 v3 v4
                where v1v2v3v4 is the 32 bit integer value
```

```
 6   Read crosspoint preset luma

     03 01 57 06
           Returns: 07 01 D7 06 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

 7   Read crosspoint key1 hue

     03 01 57 07
           Returns: 07 01 D7 07 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

 8   Read crosspoint key1 chroma

     03 01 57 08
           Returns: 07 01 D7 08 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

 9   Read crosspoint key1 luma

     03 01 57 09
           Returns: 07 01 D7 09 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

10   Read crosspoint key2 hue

     03 01 57 0A
           Returns: 07 01 D7 0A v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

11   Read crosspoint key2 chroma

     03 01 57 0B
           Returns: 07 01 D7 0B v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

12   Read crosspoint key2 luma

     03 01 57 0C
           Returns: 07 01 D7 0C v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

13   Read crosspoint dsk hue

     03 01 57 0D
           Returns: 07 01 D7 0D v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

14   Read crosspoint dsk chroma

     03 01 57 0E
           Returns: 07 01 D7 0E v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value
```

```
15   Read crosspoint dsk luma

     03 01 57 0F
           Returns: 07 01 D7 0F v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

16   Read crosspoint aux1 hue

     03 01 57 10
           Returns: 07 01 D7 10 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

17   Read crosspoint aux1 chroma

     03 01 57 11
           Returns: 07 01 D7 11 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

18   Read crosspoint aux1 luma

     03 01 57 12
           Returns: 07 01 D7 12 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

19   Read crosspoint aux2 hue

     03 01 57 13
           Returns: 07 01 D7 13 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

20   Read crosspoint aux2 chroma

     03 01 57 14
           Returns: 07 01 D7 14 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

21   Read crosspoint aux2 luma

     03 01 57 15
           Returns: 07 01 D7 15 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

22   Read emem position wipe x

     03 01 57 16
           Returns: 07 01 D7 16 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

23   Read emem position wipe y

     03 01 57 17
           Returns: 07 01 D7 17 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value
```

```
24   Read ememm multiply wipe x

     03 01 57 18
           Returns: 07 01 D7 18 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

25   Read ememm multiply wipe y

     03 01 57 19
           Returns: 07 01 D7 19 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

26   Read ememm wipe mask preset size

     03 01 57 1A
           Returns: 07 01 D7 1A v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

27   Read ememm wipe aspect

     03 01 57 1B
           Returns: 07 01 D7 1B v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

28   Read ememm preset pattern edge softness

     03 01 57 1C
           Returns: 07 01 D7 1C v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

29   Read ememm preset pattern edge width

     03 01 57 1D
           Returns: 07 01 D7 1D v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

30   Read ememm preset pattern edge hue

     03 01 57 1E
           Returns: 07 01 D7 1E v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

31   Read ememm preset pattern edge chroma

     03 01 57 1F
           Returns: 07 01 D7 1F v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

32   Read ememm preset pattern edge luma

     03 01 57 20
           Returns: 07 01 D7 20 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value
```

```
34   Read emem wipe edge softness

     03 01 57 22
           Returns: 07 01 D7 22 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

35   Read emem wipe edge width

     03 01 57 23
           Returns: 07 01 D7 23 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

36   Read emem wipe edge hue

     03 01 57 24
           Returns: 07 01 D7 24 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

37   Read emem wipe edge chroma

     03 01 57 25
           Returns: 07 01 D7 25 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

38   Read emem wipe edge luma

     03 01 57 26
           Returns: 07 01 D7 26 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

39   Read emem wipe edge opacity

     03 01 57 27
           Returns: 07 01 D7 27 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

40   Read emem wipe rotation

     03 01 57 28
           Returns: 07 01 D7 28 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

41   Read emem wipe h modulation frequency

     03 01 57 29
           Returns: 07 01 D7 29 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

42   Read emem wipe h modulation amplitude

     03 01 57 2A
           Returns: 07 01 D7 2A v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value
```

```
43   Read ememm wipe h modulation phase

     03 01 57 2B
           Returns: 07 01 D7 2B v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

44   Read ememm wipe v modulation frequency

     03 01 57 2C
           Returns: 07 01 D7 2C v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

45   Read ememm wipe v modulation amplitude

     03 01 57 2D
           Returns: 07 01 D7 2D v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

46   Read ememm wipe v modulation phase

     03 01 57 2E
           Returns: 07 01 D7 2E v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

47   Read ememm wipe symmetry

     03 01 57 2F
           Returns: 07 01 D7 2F v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

48   Read keyer key1 opacity

     03 01 57 30
           Returns: 07 01 D7 30 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

49   Read keyer key1 clip

     03 01 57 31
           Returns: 07 01 D7 31 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

50   Read keyer key1 gain

     03 01 57 32
           Returns: 07 01 D7 32 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

51   Read keyer key1 depth

     03 01 57 33
           Returns: 07 01 D7 33 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value
```

```
52   Read keyer key1 matte hue

     03 01 57 34
           Returns: 07 01 D7 34 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

53   Read keyer key1 matte chroma

     03 01 57 35
           Returns: 07 01 D7 35 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

54   Read keyer key1 matte luma

     03 01 57 36
           Returns: 07 01 D7 36 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

55   Read keyer key1 borderline opacity

     03 01 57 37
           Returns: 07 01 D7 37 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

56   Read keyer key1 borderline hue

     03 01 57 38
           Returns: 07 01 D7 38 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

57   Read keyer key1 borderline chroma

     03 01 57 39
           Returns: 07 01 D7 39 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

58   Read keyer key1 borderline luma

     03 01 57 3A
           Returns: 07 01 D7 3A v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

59   Read keyer key2 opacity

     03 01 57 3B
           Returns: 07 01 D7 3B v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

60   Read keyer key2 clip

     03 01 57 3C
           Returns: 07 01 D7 3C v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value
```

```
61   Read keyer key2 gain

     03 01 57 3D
           Returns: 07 01 D7 3D v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

62   Read keyer key2 depth

     03 01 57 3E
           Returns: 07 01 D7 3E v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

63   Read keyer key2 matte hue

     03 01 57 3F
           Returns: 07 01 D7 3F v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

64   Read keyer key2 matte chroma

     03 01 57 40
           Returns: 07 01 D7 40 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

65   Read keyer key2 matte luma

     03 01 57 41
           Returns: 07 01 D7 41 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

66   Read keyer key2 borderline opacity

     03 01 57 42
           Returns: 07 01 D7 42 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

67   Read keyer key2 borderline hue

     03 01 57 43
           Returns: 07 01 D7 43 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

68   Read keyer key2 borderline chroma

     03 01 57 44
           Returns: 07 01 D7 44 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

69   Read keyer key2 borderline luma

     03 01 57 45
           Returns: 07 01 D7 45 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value
```

```
70   Read keyer dsk opacity

     03 01 57 46
           Returns: 07 01 D7 46 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

71   Read keyer dsk clip

     03 01 57 47
           Returns: 07 01 D7 47 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

72   Read keyer dsk gain

     03 01 57 48
           Returns: 07 01 D7 48 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

74   Read keyer dsk matte hue

     03 01 57 4A
           Returns: 07 01 D7 4A v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

75   Read keyer dsk matte chroma

     03 01 57 4B
           Returns: 07 01 D7 4B v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

76   Read keyer dsk matte luma

     03 01 57 4C
           Returns: 07 01 D7 4C v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

77   Read keyer dsk borderline opacity

     03 01 57 4D
           Returns: 07 01 D7 4D v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

78   Read keyer dsk borderline hue

     03 01 57 4E
           Returns: 07 01 D7 4E v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

79   Read keyer dsk borderline chroma

     03 01 57 4F
           Returns: 07 01 D7 4F v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value
```

```
80   Read keyer dsk borderline luma

     03 01 57 50
           Returns: 07 01 D7 50 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

81   Read keyer chroma key auto setup x cursor

     03 01 57 51
           Returns: 07 01 D7 51 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

82   Read keyer chroma key auto setup y cursor

     03 01 57 52
           Returns: 07 01 D7 52 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

83   Read keyer chroma key primary hue

     03 01 57 53
           Returns: 07 01 D7 53 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

84   Read keyer chroma key bkgd selectivity

     03 01 57 54
           Returns: 07 01 D7 54 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

85   Read keyer chroma key background suppression

     03 01 57 55
           Returns: 07 01 D7 55 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

86   Read keyer chroma_key luma bkgd suppr

     03 01 57 56
           Returns: 07 01 D7 56 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

87   Read keyer chroma key fgd hue select angle

     03 01 57 57
           Returns: 07 01 D7 57 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

88   Read keyer chroma key fgd selectivity

     03 01 57 58
           Returns: 07 01 D7 58 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value
```

```
89   Read keyer chroma key fgd hue suppress angle

     03 01 57 59
           Returns: 07 01 D7 59 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

90   Read keyer chroma key secondary suppression

     03 01 57 5A
           Returns: 07 01 D7 5A v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

91   Read keyer chroma key fgd luma suppression

     03 01 57 5B
           Returns: 07 01 D7 5B v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

92   Read keyer chroma key fringe

     03 01 57 5C
           Returns: 07 01 D7 5C v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

93   Read keyer chroma key h size

     03 01 57 5D
           Returns: 07 01 D7 5D v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

94   Read keyer chroma key v size

     03 01 57 5E
           Returns: 07 01 D7 5E v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

95   Read keyer chroma key h softness

     03 01 57 5F
           Returns: 07 01 D7 5F v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

96   Read keyer chroma key v softness

     03 01 57 60
           Returns: 07 01 D7 60 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value

97   Read keyer chroma key h position

     03 01 57 61
           Returns: 07 01 D7 61 v1 v2 v3 v4
                 where v1v2v3v4 is the 32 bit integer value
```

```
 98   Read keyer chroma key v position

      03 01 57 62
            Returns: 07 01 D7 62 v1 v2 v3 v4
                  where v1v2v3v4 is the 32 bit integer value

 99   Read keyer chroma key shadow clip

      03 01 57 63
            Returns: 07 01 D7 63 v1 v2 v3 v4
                  where v1v2v3v4 is the 32 bit integer value

100   Read keyer chroma key shadow gain

      03 01 57 64
            Returns: 07 01 D7 64 v1 v2 v3 v4
                  where v1v2v3v4 is the 32 bit integer value

101   Read keyer chroma key shadow opacity

      03 01 57 65
            Returns: 07 01 D7 65 v1 v2 v3 v4
                  where v1v2v3v4 is the 32 bit integer value

102   Read keyer chroma key box mask size

      03 01 57 66
            Returns: 07 01 D7 66 v1 v2 v3 v4
                  where v1v2v3v4 is the 32 bit integer value

103   Read keyer chroma key box_mask aspect

      03 01 57 67
            Returns: 07 01 D7 67 v1 v2 v3 v4
                  where v1v2v3v4 is the 32 bit integer value

104   Read keyer chroma key box mask position x

      03 01 57 68
            Returns: 07 01 D7 68 v1 v2 v3 v4
                  where v1v2v3v4 is the 32 bit integer value

105   Read keyer chroma key box mask position y

      03 01 57 69
            Returns: 07 01 D7 69 v1 v2 v3 v4
                  where v1v2v3v4 is the 32 bit integer value

106   Read keyer chroma key box mask softness

      03 01 57 6A
            Returns: 07 01 D7 6A v1 v2 v3 v4
                  where v1v2v3v4 is the 32 bit integer value
```

```
107   Read lever arm xpt mix value

      03 01 57 6B
            Returns: 07 01 D7 6B v1 v2 v3 v4
                  where v1v2v3v4 is the 32 bit integer value
```

Analog Write Commands

```
07 01 D7 nn v1 v2 v3 v4
|  |  |  |  |  |  |  |_ byte 4 of the value
|  |  |  |  |  |  |____ byte 3 of the value
|  |  |  |  |  |_______ byte 2 of the value
|  |  |  |  |__________ byte 1 of the value
|  |  |  |_____________ designates analog value to change
|  |  |________________ command: D7h = write an analog value
|  |___________________ effects (use 01h for all)
|______________________ byte count for the command
```

The following 21 commands set matte values for the currently selected crosspoint for the bus indicated. The matte values are in effect if the crosspoint is defined as a matte.

```
  1   Write crosspoint prog hue

      07 01 D7 01 V1 V2 V3 V4
                        Def value = 0x00000000 (L)
                        Min value = 0xFFE20000 (L)
                        Max value = 0x01860000 (L)

  2   Write crosspoint prog chroma

      07 01 D7 02 V1 V2 V3 V4
                        Def value = 0x00010000 (L)
                        Min value = 0x00000000 (L)
                        Max value = 0x00010000 (L)

  3   Write crosspoint prog luma

      07 01 D7 03 V1 V2 V3 V4
                        Def value = 0x00010000 (L)
                        Min value = 0x00000000 (L)
                        Max value = 0x00010000 (L)

  4   Write crosspoint preset hue

      07 01 D7 04 V1 V2 V3 V4
                        Def value = 0x00000000 (L)
                        Min value = 0xFFE20000 (L)
                        Max value = 0x01860000 (L)
```

```
 5   Write crosspoint preset chroma

     07 01 D7 05 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

 6   Write crosspoint preset luma

     07 01 D7 06 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

 7   Write crosspoint key1 hue

     07 01 D7 07 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

 8   Write crosspoint key1 chroma

     07 01 D7 08 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

 9   Write crosspoint key1 luma

     07 01 D7 09 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

10   Write crosspoint key2 hue

     07 01 D7 0A V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

11   Write crosspoint key2 chroma

     07 01 D7 0B V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

12   Write crosspoint key2 luma

     07 01 D7 0C V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)
```

```
13   Write crosspoint dsk hue

     07 01 D7 0D V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

14   Write crosspoint dsk chroma

     07 01 D7 0E V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

15   Write crosspoint dsk luma

     07 01 D7 0F V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

16   Write crosspoint aux1 hue

     07 01 D7 10 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

17   Write crosspoint aux1 chroma

     07 01 D7 11 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

18   Write crosspoint aux1 luma

     07 01 D7 12 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

19   Write crosspoint aux2 hue

     07 01 D7 13 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

20   Write crosspoint aux2 chroma

     07 01 D7 14 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)
```

21 Write crosspoint aux2 luma

07 01 D7 15 V1 V2 V3 V4
Def value = 0x00010000 (L)
Min value = 0x00000000 (L)
Max value = 0x00010000 (L)

22 Write emem position wipe x
(This command will operate only if the positioner is enabled.)

07 01 D7 16 V1 V2 V3 V4
(4X3)
Def value = 0x00000000 (L)
Min value = 0xFFF80000 (L)
Max value = 0x00080000 (L)

(16X9)
Def value = 0x00000000 (L)
Min value = 0xFFE00000 (L)
Max value = 0x00200000 (L)

23 Write emem position wipe y
(This command will operate only if the positioner is enabled.)

07 01 D7 17 V1 V2 V3 V4
(4X3)
Def value = 0x00000000 (L)
Min value = 0xFFFA0000 (L)
Max value = 0x00060000 (L)

(16X9)
Def value = 0x00000000 (L)
Min value = 0xFFEE0000 (L)
Max value = 0x00120000 (L)

24 Write emem multiply wipe x
(This command will operate only if the wipe multiply x enable is enabled (on).)

07 01 D7 18 V1 V2 V3 V4
Def value = 0x00000000 (L)
Min value = 0x00000000 (L)
Max value = 0x000F0000 (L)

```
25   Write emem multiply wipe y
     (This command will operate only if the wipe multiply x enable
     is enabled (on).)

     07 01 D7 19 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x000F0000 (L)

26   Write emem wipe mask preset size

     07 01 D7 1A V1 V2 V3 V4
                       Def value = 0x00008000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

27   Write emem wipe aspect
     (This command works only if the wipe width enable is on.)

     07 01 D7 1B V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFFF0000 (L)
                       Max value = 0x00010000 (L)

28   Write emem preset pattern edge softness

     07 01 D7 1C V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x000A0000 (L)

29   Write emem preset pattern edge width

     07 01 D7 1D V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

30   Write emem preset pattern edge hue

     07 01 D7 1E V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

31   Write emem preset pattern edge chroma

     07 01 D7 1F V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)
```

```
32   Write ememem preset pattern edge luma

     07 01 D7 20 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

34   Write ememem wipe edge softness

     07 01 D7 22 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x000A0000 (L)

35   Write ememem wipe edge width

     07 01 D7 23 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

36   Write ememem wipe edge hue

     07 01 D7 24 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

37   Write ememem wipe edge chroma

     07 01 D7 25 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

38   Write ememem wipe edge luma

     07 01 D7 26 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

39   Write ememem wipe edge opacity

     07 01 D7 27 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

40   Write ememem wipe rotation

     07 01 D7 28 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFF4C0000 (L)
                       Max value = 0x00B40000 (L)
```

```
41   Write emem wipe h modulation frequency

     07 01 D7 29 V1 V2 V3 V4
                       Def value = 0x00020000 (L)
                       Min value = 0x00008000 (L)
                       Max value = 0x00100000 (L)
                         Note: The value 0x00000000 (L)
                             should never be sent.
                             Zero is not a valid
                             value.

42   Write emem wipe h modulation amplitude

     07 01 D7 2A V1 V2 V3 V4
                       Def value = 0x00004000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

43   Write emem wipe h modulation phase

     07 01 D7 2B V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFF4C0000 (L)
                       Max value = 0x00B40000 (L)

44   Write emem wipe v modulation frequency

     07 01 D7 2C V1 V2 V3 V4
                       Def value = 0x00020000 (L)
                       Min value = 0x00008000 (L)
                       Max value = 0x000C0000 (L)

45   Write emem wipe v modulation amplitude

     07 01 D7 2D V1 V2 V3 V4
                       Def value = 0x00004000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

46   Write emem wipe v modulation phase

     07 01 D7 2E V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00B40000 (L)
                       Max value = 0xFF4C0000 (L)

47   Write emem wipe symmetry

     07 01 D7 2F V1 V2 V3 V4
                       Def value = 0x00008000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)
```

```
48   Write keyer key1 opacity

     07 01 D7 30 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

49   Write keyer key1 clip

     07 01 D7 31 V1 V2 V3 V4
                       Def value = 0x00008000 (L)
                       Min value = 0xFFFE8000 (L)
                       Max value = 0x00028000 (L)

50   Write keyer key1 gain

     07 01 D7 32 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00005555 (L)
                       Max value = 0x04000000 (L)

51   Write keyer key1 depth

     07 01 D7 33 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0xFFFF0000 (L)
                       Max value = 0x00010000 (L)

52   Write keyer key1 matte hue

     07 01 D7 34 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

53   Write keyer key1 matte chroma

     07 01 D7 35 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

54   Write keyer key1 matte luma

     07 01 D7 36 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

55   Write keyer key1 borderline opacity

     07 01 D7 37 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)
```

```
56   Write keyer key1 borderline hue

     07 01 D7 38 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

57   Write keyer key1 borderline chroma

     07 01 D7 39 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

58   Write keyer key1 borderline luma

     07 01 D7 3A V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

59   Write keyer key2 opacity

     07 01 D7 3B V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

60   Write keyer key2 clip

     07 01 D7 3C V1 V2 V3 V4
                       Def value = 0x00008000 (L)
                       Min value = 0xFFFE8000 (L)
                       Max value = 0x00028000 (L)

61   Write keyer key2 gain

     07 01 D7 3D V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00005555 (L)
                       Max value = 0x04000000 (L)

62   Write keyer key2 depth

     07 01 D7 3E V1 V2 V3 V4
                       Def value = 0xFFFF0000 (L)
                       Min value = 0xFFFF0000 (L)
                       Max value = 0x00010000 (L)

63   Write keyer key2 matte hue

     07 01 D7 3F V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)
```

```
64   Write keyer key2 matte chroma

     07 01 D7 40 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

65   Write keyer key2 matte luma

     07 01 D7 41 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

66   Write keyer key2 borderline opacity

     07 01 D7 42 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

67   Write keyer key2 borderline hue

     07 01 D7 43 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

68   Write keyer key2 borderline chroma

     07 01 D7 44 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

69   Write keyer key2 borderline_luma

     07 01 D7 45 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

70   Write keyer dsk opacity

     07 01 D7 46 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

71   Write keyer dsk clip

     07 01 D7 47 V1 V2 V3 V4
                       Def value = 0x00008000 (L)
                       Min value = 0xFFFE8000 (L)
                       Max value = 0x00028000 (L)
```

```
72   Write keyer dsk gain

     07 01 D7 48 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00005555 (L)
                       Max value = 0x04000000 (L)

74   Write keyer dsk matte hue

     07 01 D7 4A V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

75   Write keyer dsk matte chroma

     07 01 D7 4B V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

76   Write keyer dsk matte luma

     07 01 D7 4C V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

77   Write keyer dsk borderline opacity

     07 01 D7 4D V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

78   Write keyer dsk borderline hue

     07 01 D7 4E V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

79   Write keyer dsk borderline chroma

     07 01 D7 4F V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

80   Write keyer dsk borderline luma

     07 01 D7 50 V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)
```

```
81   Write keyer chroma key auto setup x cursor

     07 01 D7 51 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFF80000 (L)
                       Max value = 0x00080000 (L)

82   Write keyer chroma key auto setup y cursor

     07 01 D7 52 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFFA0000 (L)
                       Max value = 0x00060000 (L)

83   Write keyer chroma key primary hue

     07 01 D7 53 V1 V2 V3 V4
                       Def value = 0xFEA20000 (L)
                       Min value = 0xFFE20000 (L)
                       Max value = 0x01860000 (L)

84   Write keyer chroma key bkgd selectivity

     07 01 D7 54 V1 V2 V3 V4
                       Def value = 0x00003333 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

85   Write keyer chroma key background suppression

     07 01 D7 55 V1 V2 V3 V4
                       Def value = 0x00008000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

86   Write keyer chroma key luma bkgd suppr

     07 01 D7 56 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

87   Write keyer chroma key fgd hue select angle

     07 01 D7 57 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFA60000 (L)
                       Max value = 0x005A0000 (L)

88   Write keyer chroma key fgd selectivity

     07 01 D7 58 V1 V2 V3 V4
                       Def value = 0x00002666 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)
```

```
89   Write keyer chroma key fgd hue suppress angle

     07 01 D7 59 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFD30000 (L)
                       Max value = 0x002D0000 (L)

90   Write keyer chroma key secondary suppression

     07 01 D7 5A V1 V2 V3 V4
                       Def value = 0x00010000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

91   Write keyer chroma key fgd luma suppression

     07 01 D7 5B V1 V2 V3 V4
                       Def value = 0x00004000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

92   Write keyer chroma key fringe

     07 01 D7 5C V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

93   Write keyer chroma key h size

     07 01 D7 5D V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFFF0000 (L)
                       Max value = 0x00010000 (L)

94   Write keyer chroma key v size

     07 01 D7 5E V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0xFFFF0000 (L)
                       Max value = 0x00010000 (L)

95   Write keyer chroma key h softness

     07 01 D7 5F V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)

96   Write keyer chroma key v softness

     07 01 D7 60 V1 V2 V3 V4
                       Def value = 0x00000000 (L)
                       Min value = 0x00000000 (L)
                       Max value = 0x00010000 (L)
```

```
 97   Write keyer chroma key h position

      07 01 D7 61 V1 V2 V3 V4
                        Def value = 0x00000000 (L)
                        Min value = 0xFFFF0000 (L)
                        Max value = 0x00010000 (L)

 98   Write keyer chroma key v position

      07 01 D7 62 V1 V2 V3 V4
                        Def value = 0x00000000 (L)
                        Min value = 0xFFFF0000 (L)
                        Max value = 0x00010000 (L)

 99   Write keyer chroma key shadow clip

      07 01 D7 63 V1 V2 V3 V4
                        Def value = 0x00008000 (L)
                        Min value = 0x00000000 (L)
                        Max value = 0x00010000 (L)

100   Write keyer chroma key shadow gain

      07 01 D7 64 V1 V2 V3 V4
                        Def value = 0x00004000 (L)
                        Min value = 0x00000000 (L)
                        Max value = 0x00010000 (L)

101   Write keyer chroma key shadow opacity

      07 01 D7 65 V1 V2 V3 V4
                        Def value = 0x00010000 (L)
                        Min value = 0x00000000 (L)
                        Max value = 0x00010000 (L)

102   Write keyer chroma_key box mask size

      07 01 D7 66 V1 V2 V3 V4
                        Def value = 0x00000000 (L)
                        Min value = 0x00000000 (L)
                        Max value = 0x00010000 (L)

103   Write keyer chroma key box mask aspect

      07 01 D7 67 V1 V2 V3 V4
                        Def value = 0x00000000 (L)
                        Min value = 0xFFCE0000 (L)
                        Max value = 0x00320000 (L)

104   Write keyer chroma key box mask position x

      07 01 D7 68 V1 V2 V3 V4
                        Def value = 0x00000000 (L)
                        Min value = 0xFFFC0000 (L)
                        Max value = 0x00040000 (L)
```

```
105   Write keyer chroma key box mask position y

      07 01 D7 69 V1 V2 V3 V4
                        Def value = 0x00000000 (L)
                        Min value = 0xFFFD0000 (L)
                        Max value = 0x00030000 (L)

106   Write keyer chroma key box mask softness

      07 01 D7 6A V1 V2 V3 V4
                        Def value = 0x00000000 (L)
                        Min value = 0x00000000 (L)
                        Max value = 0x00010000 (L)

107   Write lever arm xpt mix value

      07 01 D7 6B V1 V2 V3 V4
                        Def value = 0x00000000 (L)
                        Min value = 0x00000000 (L)
                        Max value = 0x00010000 (L)
```

Read Digital Value Function

```
03 01 58 nn
|  |  |  |_____________ designates digital function to read
|  |  |________________ command: 58h = read a composite function
|  |___________________ effects (use 01h for all)
|______________________ byte count for the command
```

001 Read keyer chroma key fgd noise suppress

03 01 58 01
Returns: 04 01 D8 01 nn, where nn is a hex value ranging from 00 to 03 representing none, or three different levels of field-to-field averaging of the video.

002 Read keyer chroma key box mask_control

03 01 58 02
Returns: 04 01 D8 02 nn, where nn is a hex value ranging from 00 to 07 where:
00 = No Chroma Key Box Mask
01 = Background Mask - Normal
02 = Foreground Mask - Normal
03 = Background + Forground Mask - Normal
04 = No Chroma Key Box Mask
05 = Background Mask - Inverted
06 = Foreground Mask - Inverted
07 = Background + Foreground Mask - Inverted

Write Digital Value Function

```
04 01 D8 nn dv
|  |  |  |  |___________ Actual Digital Function value
|  |  |  |______________ designates digital function value to write
|  |  |_________________ command: D8h = write a Composite function
|  |                    value
|  |____________________ effects (use 01h for all)
|_______________________ byte count for the command
```

001 Write keyer chroma key fgd noise suppress

04 01 D8 01
where: nn is a hex value ranging from 00 to 03 representing none, or three different levels of field-to-field averaging of the video.

002 Write keyer chroma key box mask control

04 01 D8 02
where: nn is a hex value ranging from 00 to 07:
00 = No Chroma Key Box Mask
01 = Background Mask - Normal
02 = Foreground Mask - Normal
03 = Background + Forground Mask - Normal
04 = No Chroma Key Box Mask
05 = Background Mask - Inverted
06 = Foreground Mask - Inverted
07 = Background + Foreground Mask - Inverted

Read Function

```
03 01 54 nn
|  |  |  |_____________ designates function to read
|  |  |________________ command: 57h = read function
|  |___________________ effects (use 01h for all)
|______________________ byte count for the command
```

001 Read ememe preset mask invert

```
03 01 54 01
      Returns: 03 01 D4 01 (if on), 03 01 D4 02 (if off)
```

002 Read pattern border on off

```
03 01 54 02
      Returns: 03 01 D4 03 (if on), 03 01 D4 04 (if off)
```

003 Read pattern softness on off

```
03 01 54 03
      Returns: 03 01 D4 05 (if on), 03 01 D4 06 (if off)
```

004 Read pattern symmetry on off

```
03 01 54 04
      Returns: 03 01 D4 07 (if on), 03 01 D4 08 (if off)
```

005 Read wipe border on off

```
03 01 54 05
      Returns: 03 01 D4 09 (if on), 03 01 D4 0A (if off)
```

006 Read wipe softness on off

```
03 01 54 06
      Returns: 03 01 D4 0B (if on), 03 01 D4 0C (if off)
```

007 Read wipe symmetry on off

```
03 01 54 07
      Returns: 03 01 D4 0C (if on), 03 01 D4 0D (if off)
```

008 Read keyer key1 edge

```
03 01 54 08
      Returns: 03 01 D4 11 (if NORMAL)
               03 01 D4 12 (if BORDER)
               03 01 D4 13 (if DROP SHADOW)
               03 01 D4 14 (if EXTRUDE)
               03 01 D4 15 (if OUTLINE)
```

009 Read keyer key1 mask

03 01 54 09
Returns: 03 01 D4 16 (if on), 03 01 D4 17 (if off)

010 Read keyer key1 invert

03 01 54 0A
Returns: 03 01 D4 18 (if INVERT), 03 01 D4 19 (if NORMAL)

011 Read keyer key1 fill

03 01 54 0B
Returns: 03 01 D4 1A (if VIDEO), 03 01 D4 1B (if MATTE)

012 Read keyer key1 video source

03 01 54 0C
Returns: 03 01 D4 1C (if SELF), 03 01 D4 1D (if AUTO SELECT)

013 Read keyer key1 key type

03 01 54 0D
Returns: 03 01 D4 1E (if LINEAR)
03 01 D4 1f (if CHROMA)
03 01 D4 20 (if LUMA)
03 01 D4 21 (if PRESET PATTERN)

014 Read keyer key2 edge

03 01 54 0E
Returns: 03 01 D4 24 (if NORMAL)
03 01 D4 25 (if BORDER)
03 01 D4 26 (if DROP SHADOW)
03 01 D4 27 (if EXTRUDE)
03 01 D4 28 (if OUTLINE)

015 Read keyer key2 mask

03 01 54 0F
Returns: 03 01 D4 29 (if on), 03 01 D4 2A (if off)

016 Read keyer key2 invert

03 01 54 10
Returns: 03 01 D4 2B (if INVERT), 03 01 D4 2C (if NORMAL)

017 Read keyer key2 fill

03 01 54 11
Returns: 03 01 D4 2D (if VIDEO),
03 01 D4 2E (if MATTE)

```
018   Read keyer key2 video source

      03 01 54 12
            Returns: 03 01 D4 2F (if SELF),
                     03 01 D4 30 (if AUTO SELECT)

019   Read keyer key2 key type

      03 01 54 13
            Returns: 03 01 D4 31 (if LINEAR)
                     03 01 D4 32 (if CHROMA)
                     03 01 D4 33 (if LUMA)
                     03 01 D4 34 (if PRESET PATTERN)

020   Read keyer_dsk_edge

      03 01 54 14
            Returns: 03 01 D4 37 (if NORMAL)
                     03 01 D4 38 (if BORDER)
                     03 01 D4 39 (if DROP SHADOW)
                     03 01 D4 3A (if EXTRUDE)
                     03 01 D4 3B (if OUTLINE)

021   Read keyer dsk mask

      03 01 54 15
            Returns: 03 01 D4 3C (if on),
                     03 01 D4 3D (if off)

022   Read keyer dsk invert

      03 01 54 16
            Returns: 03 01 D4 2E (if INVERT),
                     03 01 D4 2F (if NORMAL)

023   Read keyer dsk fill

      03 01 54 17
            Returns: 03 01 D4 40 (if VIDEO),
                     03 01 D4 41 (if MATTE)

024   Read keyer dsk video source

      03 01 54 18
            Returns: 03 01 D4 42 (if SELF),
                     03 01 D4 43 (if AUTO SELECT)

025   Read keyer dsk key type

      03 01 54 19
            Returns: 03 01 D4 44 (if LINEAR)
                     03 01 D4 45 (if CHROMA)
                     03 01 D4 46 (if LUMA)
                     03 01 D4 47 (if PRESET PATTERN)
```

```
027   Read wipe direction

      03 01 54 1B
            Returns: 03 01 D4 4A (if normal),
                     03 01 D4 4B (if reverse)

028   Read wipe flip flop

      03 01 54 1C
            Returns: 03 01 D4 4C (if on),
                     03 01 D4 4D (if off)

029   Read effects send

      03 01 54 1D
            Returns: 03 01 D4 4E (if EFFECT SEND OFF)
                     03 01 D4 4F (if SEND KEY 1)
                     03 01 D4 50 (if SEND KEY 2)
                     03 01 D4 51 (if SEND DSK)

030   Read positioner enable

      03 01 54 1E
            Returns: 03 01 D4 52 (if on),
                     03 01 D4 53 (if off)

031   Read keyer chroma key auto setup enable

      03 01 54 1F
            Returns: 03 01 D4 55 (if ENABLED),
                     03 01 D4 56 (if NOT ENABLED)

032   Read keyer chroma key show key

      03 01 54 20
            Returns: 03 01 D4 58 (if SHOW KEY),
                     03 01 D4 59 (if NOT SHOW KEY)

033   Read keyer chroma key fgd suppress enable

      03 01 54 21
            Returns: 03 01 D4 5A (if SUPPRESS),
                     03 01 D4 5B (if NOT SUPPRESS)

034   Read keyer chroma key 422 control

      03 01 54 22
            Returns: 03 01 D4 5C (if 422 CTRL),
                     03 01 D4 5D (if NOT 422 CTRL)

035   Read keyer chroma key frame of delay

      03 01 54 23
            Returns: 03 01 D4 5E (if DELAY),
                     03 01 D4 5F (if NOT DELAY)
```

```
036   Read keyer chroma key temporal filter enable

      03 01 54 24
            Returns: 03 01 D4 60 (if ENABLE),
                     03 01 D4 61 (if NOT ENABLE)

037   Read keyer chroma key key edge control

      03 01 54 25
            Returns: 03 01 D4 62 (if CONTROL),
                     03 01 D4 63 (if NOT CONTROL)

038   Read keyer chroma key key shadow control

      03 01 54 26
            Returns: 03 01 D4 64 (if CONTROL),
                     03 01 D4 65 (if NOT CONTROL)

039   Read trans mix wipe

      03 01 54 27
            returns: 03 01 D4 6C (if MIX),
                     03 01 D4 6D (if WIPE)

040   Read keyer key1 on off

      03 01 54 28
            returns: 03 01 D4 0F (if on),
                     03 01 D4 10 (if off)

041   Read keyer key2 on off

      03 01 54 29
            returns: 03 01 D4 22 (if on),
                     03 01 D4 23 (if off)

042   Read keyer dsk on off

      03 01 54 2A
            returns: 03 01 D4 35 (if on),
                     03 01 D4 36 (if off)
```

Write Function

```
03 01 D4 nn
|  |  |  |_____________ designates function
|  |  |________________ command: D8h = write a function
|  |___________________ effects (use 01h for all)
|______________________ byte count for the command
```

001 Write emem preset mask invert on

03 01 D4 01

002 Write emem preset mask invert off

03 01 D4 02

003 Write pattern border on

03 01 D4 03

004 Write pattern border off

03 01 D4 04

005 Write pattern softness on

03 01 D4 05

006 Write pattern softness off

03 01 D4 06

007 Write pattern symmetry on

03 01 D4 07

008 ite pattern symmetry off

03 01 D4 08

009 Write wipe border on

03 01 D4 09

010 Write wipe border off

03 01 D4 0A

011 Write wipe softness on

03 01 D4 0B

012 Write wipe softness off

03 01 D4 0C

013 Write wipe symmetry on

03 01 D4 0D

014 Write wipe symmetry off

03 01 D4 0E

015 Write keyer key1 on

03 01 D4 0F

016 Write keyer key1 off

03 01 D4 10

017 Write keyer key1 edge normal

03 01 D4 11

018 Write keyer key1 edge border

03 01 D4 12

019 Write keyer key1 edge shadow

03 01 D4 13

020 Write keyer key1 edge extrude

03 01 D4 14

021 Write keyer key1 edge outline

03 01 D4 15

022 Write keyer key1 mask on

03 01 D4 16

023 Write keyer key1 mask off

03 01 D4 17

024 Write keyer key1 invert on

03 01 D4 18

025 Write keyer key1 invert off

03 01 D4 19

```
026   Write keyer key1 video fill
      03 01 D4 1A
027   Write keyer key1 matte fill
      03 01 D4 1B
028   Write keyer key1 video key
      03 01 D4 1C
029   Write keyer key1 auto select
      03 01 D4 1D
030   Write keyer key1 linear key
      03 01 D4 1E
031   Write keyer key1 chroma key
      03 01 D4 1F
032   Write keyer key1 luma key
      03 01 D4 20
033   Write keyer key1 preset pattern
      03 01 D4 21
034   Write keyer key2 on
      03 01 D4 22
035   Write keyer key2 off
      03 01 D4 23
036   Write keyer key2 edge normal
      03 01 D4 24
037   Write keyer key2 edge border
      03 01 D4 25
038   Write keyer key2 edge shadow
      03 01 D4 26
039   Write keyer key2 edge extrude
      03 01 D4 27
```

040 Write keyer key2 edge outline

03 01 D4 28

041 Write keyer key2 mask on

03 01 D4 29

042 Write keyhr key2 mask off

03 01 D4 2A

043 Write keyer key2 invert on

03 01 D4 2B

044 Write keyer key2 invert off

03 01 D4 2C

045 Write keyer key2 video fill

03 01 D4 2D

046 Write keyer key2 matte fill

03 01 D4 2E

047 Write keyer key2 video key

03 01 D4 2F

048 Write keyer key2 auto select

03 01 D4 30

049 Write keyer key2 linear key

03 01 D4 31

050 Write keyer key2 chroma key

03 01 D4 32

051 Write keyer key2 luma key

03 01 D4 33

052 Write keyer key2 preset pattern

03 01 D4 34

053 Write keyer dsk on

03 01 D4 35

```
054   Write keyer dsk off
      03 01 D4 36
055   Write keyer dsk edge normal
      03 01 D4 37
056   Write keyer dsk edge border
      03 01 D4 38
057   Write keyer dsk edge shadow
      03 01 D4 39
058   Write keyer dsk edge extrude
      03 01 D4 3A
059   Write keyer dsk edge outline
      03 01 D4 3B
060   Write keyer dsk mask on
      03 01 D4 3C
061   Write keyer dsk mask off
      03 01 D4 3D
062   Write keyer dsk invert on
      03 01 D4 3E
063   Write keyer dsk invert off
      03 01 D4 3F
064   Write keyer dsk video fill
      03 01 D4 40
065   Write keyer dsk matte fill
      03 01 D4 41
066   Write keyer dsk video key
      03 01 D4 42
067   Write keyer dsk auto select
      03 01 D4 43
```

068 Write keyer dsk linear key

03 01 D4 44

069 Write keyer dsk chroma key

03 01 D4 45

070 Write keyer dsk luma key

03 01 D4 46

071 Write keyer dsk preset pattern

03 01 D4 47

072 Write keyer key1 over

03 01 D4 48

073 Write keyer key2 over

03 01 D4 49

074 Write wipe direction normal

03 01 D4 4A

075 Write wipe direction reverse

03 01 D4 4B

076 Write wipe direction flip flop on

03 01 D4 4C

077 Write wipe direction flip flop off

03 01 D4 4D

078 Write effects send off

03 01 D4 4E

079 Write effects send key1

03 01 D4 4F

080 Write effects send key2

03 01 D4 50

081 Write effects send dsk

03 01 D4 51

082 Write positioner enable on

03 01 D4 52

083 Write positioner enable off

03 01 D4 53

084 Write positioner center
(This command works only when the positioner enable is on.)

03 01 D4 54

085 Write keyer chroma key auto setup enable

03 01 D4 55

086 Write keyer chroma key no auto setup enable

03 01 D4 56

087 Write keyer chroma key auto setup take

03 01 D4 57

088 Write keyer chroma key show key

03 01 D4 58

089 Write keyer chroma key no show key

03 01 D4 59

090 Write keyer chroma key fgd suppress enable

03 01 D4 5A

091 Write keyer chroma key no fgd suppress enable

03 01 D4 5B

092 Write keyer chroma key 422 control

03 01 D4 5C

093 Write keyer chroma key no 422 control

03 01 D4 5D

094 Write keyer chroma key frame of delay

03 01 D4 5E

095 Write keyer chroma key no frame of delay

03 01 D4 5F

```
096   Croma key temporal filter enable

      03 01 D4 60

097   Write keyer chroma key no temporal filter enable

      03 01 D4 61

098   Write keyer chroma key key edge control

      03 01 D4 62

099   Write keyer chroma key no key edge control

      03 01 D4 63

100   Write keyer chroma key key shadow control

      03 01 D4 64

101   Write keyer chroma key no key shadow control

      03 01 D4 65

102   Write trans effect cut

      03 01 D4 66

103   Write trans effect at

      03 01 D4 67

104   Write trans dsk cut

      03 01 D4 68

104   Write trans dsk at

      03 01 D4 69

106   Write trans black cut

      03 01 D4 6A

107   Write trans black at

      03 01 D4 6B

108   Write trans mix

      03 01 D4 6C

109   Write trans wipe

      03 01 D4 6D
```

```
110   Write wipe multiply x enable on

      03 01 D4 6E

111   Write wipe multiply x enable off

      03 01 D4 6F

112   Write wipe multiply y enable on

      03 01 D4 70

113   Write wipe multiply y enable off

      03 01 D4 71

114   Write wipe aspect enable on

      03 01 D4 72

115   Write wipe aspect enable off

      03 01 D4 73

116   Write wipe width enable on

      03 01 D4 74

117   Write wipe width enable off

      03 01 D4 75

118   Write wipe softness enable on

      03 01 D4 76

119   Write wipe softness enable off

      03 01 D4 77

120   Write wipe symmetry enable on

      03 01 D4 78

121   Write wipe symmetry enable off

      03 01 D4 79

122   Write preset pattern width enable on

      03 01 D4 7A

123   Write preset pattern width enable off

      03 01 D4 7B
```

124 Write preset pattern softness enable on

03 01 D4 7C

125 Write preset pattern softness enable off

03 01 D4 7D

126 Write preset pattern symmetry enable on

03 01 D4 7E

127 Write preset pattern symmetry enable off

03 01 D4 7F

128 Write rotation disabled

03 01 D4 80

129 Write rotation manual enabled

03 01 D4 81

130 Write rotation velocity enabled

03 01 D4 82

131 Write rotation lever arm enabled

03 01 D4 83